

{ } Appliphyto Doc

10 juin 2018

appliphyto.com

```
+ www > controller > Admin.php
```

```
class Admin extends ControllerPrincipal {  
  
    /**  
     * Affiche la vue gestion admin  
     */  
    public function gestion_admin () {}  
  
    public function gestion_admin_pdf () {}  
  
    public function gestion_admin_entreprise_creation () {}  
  
    public function redirect_admin () {}  
  
    public function gestion_admin_accueil () {}  
  
    public function gestion_admin_accueil_update_simple () {}  
  
    public function gestion_admin_accueil_update_delete () {}  
  
    public function gestion_admin_accueil_update_insert () {}  
  
    public function gestion_admin_accueil_update () {}  
  
    public function entreprise_supprDef () {}  
  
    public function client_supprDef () {}  
  
    public function exploit_supprDef () {}  
  
    public function ilot_supprDef () {}  
  
    public function culture_supprDef () {}  
  
    public function materiel_supprDef () {}  
  
    public function entretien_supprDef () {}  
  
    public function chauffeur_supprDef () {}  
  
    public function decideur_supprDef () {}  
  
    public function periode_supprDef () {}  
  
    public function fiche_supprDef () {}  
  
}
```

```
+ www > controller > Connect.php
```

```
class Connect extends ControllerPrincipal {  
  
    public function index () {}  
  
    public function client () {}  
  
    public function client_export_csv () {}  
  
    public function client_pdf () {}  
  
    public function client_info () {}  
  
    public function client_info_pdf () {}  
  
    public function client_creation () {}  
  
    public function client_info_exploitation_info () {}  
  
    public function exploit_export_csv () {}  
  
    public function client_info_exploitation_info_pdf () {}  
  
    public function client_info_exploitation_ajout () {}  
  
    public function client_info_ilot_info () {}  
  
    public function ilot_export_csv () {}  
  
    public function client_info_ilot_ajout () {}  
  
    public function client_info_exploitation_info_culture_info () {}  
  
    public function culture_export_csv () {}  
  
    public function client_info_exploitation_info_culture_edit () {}  
  
    public function local_phyto () {}  
  
    public function local_phyto_visuel () {}  
  
    public function materiel_prestataire () {}  
  
    public function materiel_prestataire_pdf () {}  
  
    public function materiel_export_csv () {}  
  
    public function materiel_fiche () {}  
  
    public function entretien_export_csv () {}  
  
    public function materiel_fiche_pdf () {}  
  
}
```

```
public function materiel_prestataire_ajout () {}

public function materiel_entretien_ajout () {}

public function chauffeur_prestataire () {}

public function chauffeur_export_csv () {}

public function periode_export_csv () {}

public function chauffeur_prestataire_pdf () {}

public function chauffeur_info () {}

public function chauffeur_info_pdf () {}

public function chauffeur_ajout () {}

public function periode_ajout () {}

public function decideur_prestataire () {}

public function decideur_export_csv () {}

public function decideur_prestataire_pdf () {}

public function decideur_info () {}

public function decideur_info_pdf () {}

public function decideur_ajout () {}

public function gestion_prestataire () {}

public function gestion_prestataire_pdf () {}

public function gestion_prestataire_entreprise_modif () {}

public function gestion_prestataire_entreprise_modif_logo () {}

public function gestion_prestataire_entreprise_modif_password () {}

public function gestion_prestataire_ce_modif () {}

public function entreprise_export_csv () {}

public function liste_fiche_chantier () {}

public function liste_fiche_chantier_pdf () {}

public function fiche_info () {}

public function fiche_ajout_client () {}
```

```
public function fiche_ajout_chantier () {}

public function fiche_ajout_option () {}

public function fiche_ajout_consigne () {}

public function fiche_dose_recommande () {}

public function fiche_fin () {}

public function fiche_rendu () {}

public function contact () {}

public function contactSend () {}

public function dossier () {}

public function organigramme_ajouter_personnel_administratif () {}

public function organigramme_modifier_personnel_administratif () {}

public function organigramme_supprimer_personnel_administratif () {}

public function organigramme_responsable_referentiel_application () {}

public function organigramme () {}

}
```

```
+ www > controller > ControllerPrincipal.php
```

```
class ControllerPrincipal {  
  
    protected $vars;  
  
    protected $layout;  
  
    /**  
     *Envoi de données à la vue  
     *@param array Données à envoyer à la vue  
     */  
    protected function set ($d) {}  
  
    /**  
     *Envoi de la vue (view)  
     *@param string Nom de la vue  
     */  
    protected function render ($filename) {}  
  
}
```

```
+ www > controller > Home.php
```

```
class Home extends ControllerPrincipal {  
  
    public function doc () {}  
  
    /**  
     * Affiche la vue index  
     */  
    public function index ($params = false) {}  
  
    /**  
     * Affiche la vue login  
     */  
    public function login () {}  
  
    public function logout () {}  
  
    public function register () {}  
  
    public function send_form () {}  
  
    public function send_form_register () {}  
  
    public function forget () {}  
  
    public function reset () {}  
  
    public function information () {}  
  
}
```

```
+ www > model > App.php
```

```
class App {  
  
    public static $db;  
  
    public static function getDatabase () {}  
  
    public static function getAuth () {}  
  
    public static function redirect ($page) {}  
  
}
```

```
+ www > model > Auth.php
```

```
class Auth {

    private $options;

    private $session;

    public function __construct (Session $session,
                                $options = array) {}

    /**
     * Cryptage du mot de passe
     * @param varchar $_POST['password']
     * @return le mot de passe crypté
     */
    public function hashPassword ($password) {}

    /**
     * Restriction des utilisateurs lors de la connexion
     */
    public function restrict ($acceptRoles = array) {}

    public function user () {}

    public function connect ($user) {}

    /**
     * Reconnexion
     * @param object Instance de la BDD
     */
    public function connectFromCookie ($db) {}

    /**
     * Connexion
     * @param object Instance de la BDD
     * @param varchar $_POST['email']
     * @param varchar $_POST['password']
     * @param bool Connexion automatique par le cookie
     * @return Informations de connexion de l'utilisateur
     */
    public function login ($db,
                          $email,
                          $password,
                          $remember = false) {}

    /**
     * Création d'un cookie
     * @param object Instance de la BDD
     * @param varchar ID
     * @return Requête vers BDD de Création d'un cookie
     */
    public function remember ($db,
```

```
$user_id) {}
```

```
/**
```

```
 * Déconnexion
```

```
 */
```

```
public function logout () {}
```

```
/**
```

```
 * Réinitialiser le mot de passe
```

```
 * @param object Instance de la BDD
```

```
 * @param varchar Email de l'utilisateur
```

```
 * @return $user
```

```
 */
```

```
public function resetPassword ($db,  
                               $email) {}
```

```
/**
```

```
 * Confirmation de la réinitialisation du mot de passe
```

```
 * @param object Instance de la BDD
```

```
 * @param varchar Id de l'utilisateur
```

```
 * @param varchar Jeton de sécurité de l'utilisateur
```

```
 */
```

```
public function checkResetToken ($db,  
                                 $user_id,  
                                 $token) {}
```

```
}
```

```
+ www > model > CSV.php
```

```
class CSV {  
  
    private $out;  
  
    public function __construct ($filename) {}  
  
    public function __destruct () {}  
  
    public function arrayToCSV ($array) {}  
  
    public function lineToCSV ($line) {}  
  
}
```

```
+ www > model > Database.php
```

```
class Database {

    public $pdo;

    public function __construct ($login,
                                $password,
                                $database_name,
                                $host = 'appliphyidroot.mysql.db') {}

    /**
     * @param $query
     * @param array $params
     * @return PDOStatement
     */
    public function query ($query,
                           $params = false) {}

    /**
     * Quote un champ
     * @param array Champ à quoter
     * @return Champ quoté
     */
    public function quote ($param) {}

    /**
     * Quote un tableau
     * @param array Tableau à quoter
     * @return Tableau quoté
     */
    public function allQuote ($params) {}

    /**
     * Récupere le dernier ID inséré en BDD
     * @return Retourne le dernier ID inséré en BDD
     */
    public function lastInsertId () {}

    /**
     * Récupère la visibilité d'un champ
     * @return Retourne le champ si visible
     */
    public function visible ($term,
                            $table,
                            $id,
                            $prefix = 'ID_') {}

    /**
     * Titre
     * @param
     * @param
     * @return
     */
}
```

```
*/  
public function addyears ($date,  
                        $annees) {}  
  
}
```

```
+ www > model > Edit.php
```

```
class Edit {  
  
    public function insert ($table,  
                            $fields,  
                            $except = false) {}  
  
}
```

```
+ www > model > FPDF_Merge.php
```

```
/**
 * FPDF_Merge
 * Tool to merge PDFs created by fpdf 1.6
 *
 * @version 1.0
 * @date 2011-02-18
 * @author DEMONTE Jean-Baptiste <jbdemonte@gmail.com>
 * @copyright ? Digitick <www.digitick.net> 2011
 * @license GNU Lesser General Public License v3.0
 *
 * Why this tool ?
 *
 * All the library tested (fpdi, ...) produce too heavy pdf
 * because, these are not optimized.
 * This library parses the pages, get the objects included (font, images)
 * generate a hash to store only once these objects or reuse previous stored ones.
 *
 *
 * Notes
 *
 * - links are not supported in this version
 * - all pages are included (because this was what we needed) but update it
 * to create an add(array pages) should be easy
 *
 *
 * I tried to optimize a lot this tool using X-Debug, let me know if you do best :)
 * If you get trouble or want to comment, feel free to send me an email.
 *
 * Use
 *
 * $merge = new FPDF_Merge();
 * $merge->add('/tmp/pdf-1.pdf');
 * $merge->add('/tmp/pdf-2.pdf');
 * $merge->output('/tmp/pdf-merge.pdf'); // or $merge->output(); to open it directly in
the browser
 *
 */
class FPDF_Merge {

    const TYPE_NULL = 0;

    const TYPE_TOKEN = 1;

    const TYPE_REFERENCE = 2;

    const TYPE_REFERENCE_F = 3;

    const TYPE_NUMERIC = 4;

    const TYPE_HEX = 5;
```

```

const TYPE_BOOL = 6;

const TYPE_STRING = 7;

const TYPE_ARRAY = 8;

const TYPE_DICTIONARY = 9;

const TYPE_STREAM = 10;

private $buffer;

private $compress;

private $fonts;

private $objects;

private $pages;

private $ref;

private $n;

private $xref;

public function __construct () {}

private function error ($msg) {}

private function parse ($buffer,
                        &$len,
                        &$off) {}

private function getObject ($f,
                            $xref,
                            $index,
                            $includeSubObject = false) {}

private function _resolveValues ($f,
                                $xref,
                                $item) {}

private function getResources ($f,
                              $xref,
                              $page) {}

private function getContent ($f,
                             $xref,
                             $page) {}

private function _getContent ($f,
                              $xref,

```

```

        $content) {}

private function getCompression ($f,
                                $xref,
                                $item) {}

private function getStream ($f,
                            $xref,
                            $item) {}

private function storeObject ($item,
                              $type = false) {}

private function _out ($raw) {}

private function _strval ($value) {}

private function _toStream ($item) {}

private function _newobj ($n = null) {}

private function _addObj ($dico = null,
                          $buf = null) {}

private function _getObjectType ($object) {}

private function _putObject ($object) {}

private function _putObjects () {}

private function _putResources () {}

public function add ($filename) {}

public function output ($filename = null) {}

}

```

```
+ www > model > Fpdf.php
```

```
class FPDF {  
  
    protected $page;  
  
    protected $n;  
  
    protected $offsets;  
  
    protected $buffer;  
  
    protected $pages;  
  
    protected $state;  
  
    protected $compress;  
  
    protected $k;  
  
    protected $DefOrientation;  
  
    protected $CurOrientation;  
  
    protected $StdPageSizes;  
  
    protected $DefPageSize;  
  
    protected $CurPageSize;  
  
    protected $CurRotation;  
  
    protected $PageInfo;  
  
    protected $wPt;  
  
    protected $hPt;  
  
    protected $w;  
  
    protected $h;  
  
    protected $lMargin;  
  
    protected $tMargin;  
  
    protected $rMargin;  
  
    protected $bMargin;  
  
    protected $cMargin;  
  
    protected $x;
```

protected \$y;
protected \$lath;
protected \$LineWidth;
protected \$fontpath;
protected \$CoreFonts;
protected \$fonts;
protected \$FontFiles;
protected \$encodings;
protected \$cmaps;
protected \$FontFamily;
protected \$FontStyle;
protected \$underline;
protected \$CurrentFont;
protected \$FontSizePt;
protected \$FontSize;
protected \$DrawColor;
protected \$FillColor;
protected \$TextColor;
protected \$ColorFlag;
protected \$WithAlpha;
protected \$ws;
protected \$images;
protected \$PageLinks;
protected \$links;
protected \$AutoPageBreak;
protected \$PageBreakTrigger;
protected \$InHeader;

```

protected $InFooter;

protected $AliasNbPages;

protected $ZoomMode;

protected $LayoutMode;

protected $metadata;

protected $PDFVersion;

public function __construct ($orientation = 'P',
                             $unit = 'mm',
                             $size = 'A4') {}

public function SetMargins ($left,
                            $top,
                            $right = null) {}

public function SetLeftMargin ($margin) {}

public function SetTopMargin ($margin) {}

public function SetRightMargin ($margin) {}

public function SetAutoPageBreak ($auto,
                                   $margin = 0) {}

public function SetDisplayMode ($zoom,
                                $layout = 'default') {}

public function SetCompression ($compress) {}

public function SetTitle ($title,
                          $isUTF8 = false) {}

public function SetAuthor ($author,
                           $isUTF8 = false) {}

public function SetSubject ($subject,
                            $isUTF8 = false) {}

public function SetKeywords ($keywords,
                             $isUTF8 = false) {}

public function SetCreator ($creator,
                           $isUTF8 = false) {}

public function AliasNbPages ($alias = '257' ) {}

public function Error ($msg) {}

```

```

public function Close () {}

public function AddPage ($orientation = '',
                        $size = '',
                        $rotation = 0) {}

public function Header () {}

public function Footer () {}

public function PageNo () {}

public function SetDrawColor ($r,
                              $g = null,
                              $b = null) {}

public function SetFillColor ($r,
                              $g = null,
                              $b = null) {}

public function SetTextColor ($r,
                              $g = null,
                              $b = null) {}

public function GetStringWidth ($s) {}

public function SetLineWidth ($width) {}

public function Line ($x1,
                    $y1,
                    $x2,
                    $y2) {}

public function Rect ($x,
                    $y,
                    $w,
                    $h,
                    $style = '') {}

public function AddFont ($family,
                        $style = '',
                        $file = '') {}

public function SetFont ($family,
                        $style = '',
                        $size = 0) {}

public function SetFontSize ($size) {}

public function AddLink () {}

public function SetLink ($link,
                        $y = 0,

```

```

        $page = -1) {}

public function Link ($x,
                    $y,
                    $w,
                    $h,
                    $link) {}

public function Text ($x,
                    $y,
                    $txt) {}

public function AcceptPageBreak () {}

public function Cell ($w,
                    $h = 0,
                    $txt = '',
                    $border = 0,
                    $ln = 0,
                    $align = '',
                    $fill = false,
                    $link = '') {}

public function MultiCell ($w,
                        $h,
                        $txt,
                        $border = 0,
                        $align = 'J',
                        $fill = false) {}

public function Write ($h,
                    $txt,
                    $link = '') {}

public function Ln ($h = null) {}

public function Image ($file,
                    $x = null,
                    $y = null,
                    $w = 0,
                    $h = 0,
                    $type = '',
                    $link = '') {}

public function GetPageWidth () {}

public function GetPageHeight () {}

public function GetX () {}

public function SetX ($x) {}

public function GetY () {}

```

```

public function SetY ($y,
                    $resetX = true) {}

public function SetXY ($x,
                     $y) {}

public function Output ($dest = '',
                      $name = '',
                      $isUTF8 = false) {}

protected function _dochecks () {}

protected function _checkoutput () {}

protected function _getpagesize ($size) {}

protected function _beginpage ($orientation,
                              $size,
                              $rotation) {}

protected function _endpage () {}

protected function _loadfont ($font) {}

protected function _isascii ($s) {}

protected function _httpencode ($param,
                                $value,
                                $isUTF8) {}

protected function _UTF8toUTF16 ($s) {}

protected function _escape ($s) {}

protected function _textstring ($s) {}

protected function _dounderline ($x,
                                $y,
                                $txt) {}

protected function _parsejpg ($file) {}

protected function _parsepng ($file) {}

protected function _parsepngstream ($f,
                                    $file) {}

protected function _readstream ($f,
                               $n) {}

protected function _readint ($f) {}

protected function _parsegif ($file) {}

```

```
protected function _out ($s) {}

protected function _put ($s) {}

protected function _getoffset () {}

protected function _newobj ($n = null) {}

protected function _putstream ($data) {}

protected function _putstreamobject ($data) {}

protected function _putpage ($n) {}

protected function _putpages () {}

protected function _putfonts () {}

protected function _tounicodecmap ($uv) {}

protected function _putimages () {}

protected function _putimage (&$info) {}

protected function _putxobjectdict () {}

protected function _putresourcedict () {}

protected function _putresources () {}

protected function _putinfo () {}

protected function _putcatalog () {}

protected function _putheader () {}

protected function _puttrailer () {}

protected function _enddoc () {}

}
```

```
+ www > model > Mc_table.php
```

```
abstract class PDF_MC_Table extends FPDF {  
  
    private $entrepInfos;  
  
    private $globalHex;  
  
    public $fill;  
  
    public $autothead;  
  
    public $colors;  
  
    public $bandeau;  
  
    public $pagination;  
  
    public $description;  
  
    public $dateCreation;  
  
    public $dateRevision;  
  
    public $numDoc;  
  
    public $exigence;  
  
    protected $B;  
  
    protected $I;  
  
    protected $U;  
  
    protected $HREF;  
  
    protected $fontList;  
  
    protected $issetfont;  
  
    protected $issetcolor;  
  
    private $widths;  
  
    private $aligns;  
  
    public function __construct ($orientation = 'P',  
                                $unit = 'mm',  
                                $format = 'A4') {}  
  
    public function WriteHTML ($html) {}  
  
    public function PutLink ($URL,
```

```

        $txt) {}

public function SetStyle ($tag,
                        $enable) {}

public function CloseTag ($tag) {}

public function OpenTag ($tag,
                        $attr) {}

public function SetEntrep ($id) {}

public function Bandeau ($fontLarge,
                        $fontMed,
                        $fontSmall,
                        $height,
                        $x) {}

public function Header () {}

abstract public function thead () {}

public function Footer () {}

public function getGlobalHex () {}

public function setGlobalHex ($globalHex) {}

public function useHexAsDraw () {}

public function useHexAsFill () {}

public static function create_preview_images ($file_name) {}

public static function convertExternal ($filename,
                                       $outname) {}

public function SetWidths ($w) {}

public function SetAligns ($a) {}

public function Row ($data) {}

public function NbLines ($w,
                        $txt) {}

public function CheckPageBreak ($h) {}

public function elegantwrap ($text,
                            $width) {}

public function splitonchar ($text,
                            $char) {}

```

}

```
+ www > model > PDF.php
```

```
class InfoOp extends PDF_MC_Table {  
  
    public function thead () {}  
  
    public function Infos ($infos) {}  
  
    public function ListePeriodes ($data) {}  
  
}
```

```
+ www > model > PDF.php
```

```
class ListeOp extends PDF_MC_Table {  
  
    public function thead () {}  
  
    public function Tab ($data) {}  
  
}
```

```
+ www > model > PDF.php
```

```
class InfoExploit extends PDF_MC_Table {  
  
    public function thead () {}  
  
    public function CInfos ($infos) {}  
  
    public function ExpInfos ($infos) {}  
  
    public function IlotInfos ($infos) {}  
  
    public function ListeCultures ($data) {}  
  
}
```

```
+ www > model > PDF.php
```

```
class InfoClient extends PDF_MC_Table {  
  
    public function thead () {}  
  
    public function Infos ($infos) {}  
  
    public function ListeExploits ($data) {}  
  
}
```

```
+ www > model > PDF.php
```

```
class ListeClient extends PDF_MC_Table {  
  
    public function thead () {}  
  
    public function Tab ($data) {}  
  
}
```

```
+ www > model > PDF.php
```

```
class InfoDec extends PDF_MC_Table {  
  
    public function thead () {}  
  
    public function Infos ($infos) {}  
  
}
```

```
+ www > model > PDF.php
```

```
class InfoEnt extends PDF_MC_Table {  
  
    public function InfosChef ($infos) {}  
  
    public function thead () {}  
  
    public function InfosEnt ($data) {}  
  
}
```

```
+ www > model > PDF.php
```

```
class ListeEnt extends PDF_MC_Table {  
  
    public function thead () {}  
  
    public function Tab ($data) {}  
  
}
```

```
+ www > model > PDF.php
```

```
class ListeFDS extends PDF_MC_Table {  
  
    public function thead () {}  
  
    public function Liste ($data) {}  
  
}
```

```
+ www > model > PDF.php
```

```
class InfoMat extends PDF_MC_Table {  
  
    public function thead () {}  
  
    public function Infos ($infos) {}  
  
    public function ListeEpreuves ($data) {}  
  
}
```

```
+ www > model > PDF.php
```

```
class Agreement extends PDF_MC_Table {  
  
    public function thead () {}  
  
    public function Titre ($n) {}  
  
    public function GrandIntercalaire ($texte) {}  
  
    public function Intercalaire ($titre,  
                                $exigenceTxt,  
                                $detailTxt,  
                                $ptsControleTxt,  
                                $docsRefTxt,  
                                $modalitesTxt,  
                                $lieuTxt) {}  
  
    public function Liste ($arr) {}  
  
}
```

```
+ www > model > Session.php
```

```
class Session {

    public static $instance;

    public static function getInstance () {}

    public function __construct () {}

    /**
     * Affiche un message flash
     * @param Texte à afficher
     * @param Type de message (defaut = success)
     */
    public function setFlash ($message,
                             $type = 'success') {}

    /**
     * Détecte si il y a des messages flash en session
     * @return Oui ou Non
     */
    public function hasFlashes () {}

    /**
     * Récupère les messages flash
     * @return Retourne les messages flash
     */
    public function getFlashes () {}

    /**
     * Écrit dans la Session
     * @param Clé de la valeur à rentrer
     * @param Valeur
     */
    public function write ($key,
                          $value) {}

    /**
     * Lit les clés dans la Session
     * @param Clé de la valeur à lire
     */
    public function read ($key) {}

    /**
     * Supprime une clé dans la Session
     * @param Clé à supprimer
     */
    public function delete ($key) {}

    /**
     * Affiche le message flash dans le html
     * @return Div avec le message flash
     */
}
```

```
*/  
public function flash () {}  
  
}
```

```
+ www > model > Str.php
```

```
class Str {  
  
    /**  
     * Créer une chaine aléatoire  
     * @param Longueur de la chaine à créer  
     * @return Chaine de caractère  
     */  
    public static function random ($length) {}  
  
}
```

```
+ www > model > Validator.php
```

```
class Validator {  
  
    private $data;  
  
    private $errors;  
  
    public function __construct ($data) {}  
  
    private function getField ($field) {}  
  
    public function isAlpha ($field,  
                             $errorMsg) {}  
  
    public function isUniq ($field,  
                           $db,  
                           $table,  
                           $errorMsg) {}  
  
    public function isEmail ($field,  
                             $errorMsg) {}  
  
    public function isConfirmed ($field,  
                                 $errorMsg = '') {}  
  
    public function isValid () {}  
  
    public function getErrors () {}  
  
}
```

```
+ www > model > classes > ADR.php
```

```
class ADR {  
  
    private $numero_ONU;  
  
    private $designation;  
  
    private $etiquette;  
  
    private $emballage;  
  
    private $code_transport;  
  
    private $code_tunnel;  
  
    public function setNumeroOnu ($numero_ONU) {}  
  
    public function getNumeroOnu () {}  
  
    public function setDesignation ($designation) {}  
  
    public function getDesignation () {}  
  
    public function setEtiquette ($etiquette) {}  
  
    public function getEtiquette () {}  
  
    public function setEmballage ($etiquette) {}  
  
    public function getEmballage () {}  
  
    public function setCodeTransport ($code_transport) {}  
  
    public function getCodeTransport () {}  
  
    public function setCodeTunnel ($code_tunnel) {}  
  
    public function getCodeTunnel () {}  
  
}
```

```
+ www > model > classes > Bande_H.php
```

```
/**
 * @name Base_Bande_H
 * @version 19/05/2016 (dd/mm/yyyy)
 * @author Axel Jalabert
 */
abstract class Base_Bande_H {

    const TABLENAME = 'BANDE_H';

    const FIELDNAME_ID_BH = 'id_bh';

    const FIELDNAME_ID_CHANT = 'id_chant';

    const FIELDNAME_TEXT_BH = 'text_bh';

    /** @var PDO */
    protected $_pdo;

    /** @var array tableau pour le chargement fainéant */
    protected static $_lazyload;

    /** @var int */
    protected $_id_bh;

    /** @var int */
    protected $_id_chant;

    /** @var string */
    protected $_text_bh;

    /**
     * Construire un(e) bande_h
     * @param $pdo PDO
     * @param $id_bh int
     * @param $id_chant int
     * @param $text_bh string
     * @param $lazyload bool Activer le chargement fainéant ?
     */
    protected function __construct (PDO $pdo,
                                    $id_bh,
                                    $id_chant,
                                    $text_bh,
                                    $lazyload = true) {}

    /**
     * Créer un(e) bande_h
     * @param $pdo PDO
     * @param $id_bh int
     * @param $id_chant int
     * @param $text_bh string
     * @param $lazyload bool Activer le chargement fainéant ?
     */
}
```

```

* @return Bande_H
*/
public static function create (PDO $pdo,
                              $id_bh,
                              $id_chant,
                              $text_bh,
                              $lazyload = true) {}

/**
 * Compter les bande_hs
 * @param $pdo PDO
 * @return int Nombre de bande_hs
 */
public static function count (PDO $pdo) {}

/**
 * Requête de sélection
 * @param $pdo PDO
 * @param $where string|array
 * @param $orderby string|array
 * @param $limit string|array
 * @param $from string|array
 * @return PDOStatement
 */
protected static function _select (PDO $pdo,
                                    $where = null,
                                    $orderby = null,
                                    $limit = null,
                                    $from = null) {}

/**
 * Charger un(e) bande_h
 * @param $pdo PDO
 * @param $id_bh int
 * @param $lazyload bool Activer le chargement fainéant ?
 * @return Bande_H
 */
public static function load (PDO $pdo,
                            $id_bh,
                            $lazyload = true) {}

/**
 * Recharger les données depuis la base de données
 */
public function reload () {}

/**
 * Charger tous/toutes les bande_hs
 * @param $pdo PDO
 * @param $lazyload bool Activer le chargement fainéant ?
 * @return Bande_H[] Tableau de bande_hs
 */
public static function loadAll (PDO $pdo,

```

```

        $lazyload = true) {}

/**
 * Sélectionner tous/toutes les bande_hs
 * @param $pdo PDO
 * @return PDOStatement
 */
public static function selectAll (PDO $pdo) {}

/**
 * Récupérer le/la bande_h suivant(e) d'un jeu de résultats
 * @param $pdo PDO
 * @param $pdoStatement PDOStatement
 * @param $lazyload bool Activer le chargement fainéant ?
 * @return Bande_H
 */
public static function fetch (PDO $pdo,
                             PDOStatement $pdoStatement,
                             $lazyload = true) {}

/**
 * Récupérer tous/toutes les bande_hs d'un jeu de résultats
 * @param $pdo PDO
 * @param $pdoStatement PDOStatement
 * @param $lazyload bool Activer le chargement fainéant ?
 * @return Bande_H[] Tableau de bande_hs
 */
public static function fetchAll (PDO $pdo,
                                 PDOStatement $pdoStatement,
                                 $lazyload = true) {}

/**
 * Test d'égalité
 * @param $bande_h Bande_H
 * @return bool Les objets sont-ils égaux ?
 */
public function equals ($bande_h) {}

/**
 * Vérifier que le/la bande_h existe en base de données
 * @return bool Le/La bande_h existe en base de données ?
 */
public function exists () {}

/**
 * Supprimer le/la bande_h
 * @return bool Opération réussie ?
 */
public function delete () {}

/**
 * Mettre à jour un champ dans la base de données
 * @param $fields array

```

```

* @param $values array
* @return bool Opération réussie ?
*/
protected function _set ($fields,
                        $values) {}

/**
* Mettre à jour tous les champs dans la base de données
* @return bool Opération réussie ?
*/
public function update () {}

/**
* Récupérer le/la id_bh
* @return int
*/
public function getId_bh () {}

/**
* Récupérer le/la id_chant
* @return int
*/
public function getId_chant () {}

/**
* Définir le/la id_chant
* @param $id_chant int
* @param $execute bool Exécuter la requête update ?
* @return bool Opération réussie ?
*/
public function setId_chant ($id_chant,
                            $execute = true) {}

/**
* Récupérer le/la text_bh
* @return string
*/
public function getText_bh () {}

/**
* Définir le/la text_bh
* @param $text_bh string
* @param $execute bool Exécuter la requête update ?
* @return bool Opération réussie ?
*/
public function setText_bh ($text_bh,
                           $execute = true) {}

/**
* ToString
* @return string Représentation de bande_h sous la forme d'un string
*/
public function __toString () {}

```

```
/**
 * S rialiser
 * @param $serialize bool Activer la s rialisation ?
 * @return string S rialisation du/de la bande_h
 */
public function serialize ($serialize = true) {}

/**
 * D s rialiser
 * @param $pdo PDO
 * @param $string string S rialisation du/de la bande_h
 * @param $lazyload bool Activer le chargement fain ant ?
 * @return Bande_H
 */
public static function unserialize (PDO $pdo,
                                     $string,
                                     $lazyload = true) {}

}
```

```
+ www > model > classes > Bande_H.php
```

```
/**
 * @name Bande_H
 * @version 19/05/2016 (dd/mm/yyyy)
 */
class Bande_H extends Base_Bande_H {

    const TABLENAME = 'BANDE_H';

    const FIELDNAME_ID_BH = 'id_bh';

    const FIELDNAME_ID_CHANT = 'id_chant';

    const FIELDNAME_TEXT_BH = 'text_bh';

    public static function infoBandeH (PDO $pdo,
                                        $fiche) {}

    public static function manual_create (PDO $pdo,
                                        $id_fiche,
                                        $id_chantier,
                                        $bande_h) {}

}
```

```
+ www > model > classes > Cadastre.php
```

```
class Cadastre {  
  
    private $numero_cadastre;  
  
    public function setNumeroCadastre ($numero_cadastre) {}  
  
    public function getNumeroCadastre () {}  
  
}
```

```
+ www > model > classes > Chantier.php
```

```
/**
 * @name Base_Chantier
 * @version 19/07/2016 (dd/mm/yyyy)
 * @author Axel Jalabert
 */
abstract class Base_Chantier {

    const TABLENAME = 'CHANTIER';

    const FIELDNAME_ID_CHANT = 'id_chant';

    const FIELDNAME_NUM_CHANT = 'num_chant';

    const FIELDNAME_DATE_CHANT = 'date_chant';

    const FIELDNAME_MODIF_CHANT = 'modif_chant';

    const FIELDNAME_PAC_CHANT = 'pac_chant';

    const FIELDNAME_IMP_CHANT = 'imp_chant';

    const FIELDNAME_CODE_PROD1 = 'code_prod1';

    const FIELDNAME_CODE_PROD2 = 'code_prod2';

    const FIELDNAME_CODE_PROD3 = 'code_prod3';

    const FIELDNAME_DOSEREC_PROD1 = 'doserec_prod1';

    const FIELDNAME_DOSEREC_PROD2 = 'doserec_prod2';

    const FIELDNAME_DOSEREC_PROD3 = 'doserec_prod3';

    const FIELDNAME_ID_CLI = 'id_cli';

    const FIELDNAME_ID_ENTREP = 'id_entrep';

    const FIELDNAME_ID_MAT = 'id_mat';

    const FIELDNAME_ID_EMP_CHAUF = 'id_emp_chauf';

    const FIELDNAME_ID_EMP_DEC = 'id_emp_dec';

    const FIELDNAME_JET_CHANT = 'jet_chant';

    /** @var PDO */
    protected $_pdo;

    /** @var array tableau pour le chargement fainéant */
    protected static $_lazyload;
```

```
/** @var int */
protected $_id_chant;

/** @var int */
protected $_num_chant;

/** @var string */
protected $_date_chant;

/** @var int */
protected $_modif_chant;

/** @var int */
protected $_pac_chant;

/** @var int */
protected $_imp_chant;

/** @var string */
protected $_code_prod1;

/** @var string */
protected $_code_prod2;

/** @var string */
protected $_code_prod3;

/** @var string */
protected $_doserec_prod1;

/** @var string */
protected $_doserec_prod2;

/** @var string */
protected $_doserec_prod3;

/** @var int */
protected $_id_cli;

/** @var string */
protected $_id_entrep;

/** @var int */
protected $_id_mat;

/** @var int */
protected $_id_emp_chauf;

/** @var int */
protected $_id_emp_dec;

protected $_jet_chant;
```

```

/**
 * Construire un(e) chantier
 * @param $pdo PDO
 * @param $idChantier int
 * @param $id_chant int
 * @param $num_chant int
 * @param $date_chant string
 * @param $modif_chant int
 * @param $pac_chant int
 * @param $imp_chant int
 * @param $code_prod1 string
 * @param $code_prod2 string
 * @param $code_prod3 string
 * @param $doserec_prod1 string
 * @param $doserec_prod2 string
 * @param $doserec_prod3 string
 * @param $id_cli int
 * @param $id_entrep string
 * @param $id_mat int
 * @param $id_emp_chauf int
 * @param $id_emp_dec int
 * @param $lazyload bool Activer le chargement fainéant ?
 */

```

```

protected function __construct (PDO $pdo,
                                $id_chant,
                                $num_chant,
                                $date_chant,
                                $modif_chant,
                                $pac_chant,
                                $imp_chant,
                                $code_prod1,
                                $code_prod2,
                                $code_prod3,
                                $doserec_prod1,
                                $doserec_prod2,
                                $doserec_prod3,
                                $id_cli,
                                $id_entrep,
                                $id_mat,
                                $id_emp_chauf,
                                $id_emp_dec,
                                $jet_chant,
                                $lazyload = true) {}

```

```

/**
 * Créer un(e) chantier
 * @param $pdo PDO
 * @param $id_chant int
 * @param $num_chant int
 * @param $date_chant string
 * @param $modif_chant int
 * @param $pac_chant int
 * @param $imp_chant int

```

```

* @param $code_prod1 string
* @param $code_prod2 string
* @param $code_prod3 string
* @param $doserec_prod1 string
* @param $doserec_prod2 string
* @param $doserec_prod3 string
* @param $id_cli int
* @param $id_entrep string
* @param $id_mat int
* @param $id_emp_chauf int
* @param $id_emp_dec int
* @param $lazyload bool Activer le chargement fainéant ?
* @return Chantier
*/
public static function create (PDO $pdo,
                               $id_chant,
                               $num_chant,
                               $date_chant,
                               $modif_chant,
                               $pac_chant,
                               $imp_chant,
                               $code_prod1,
                               $code_prod2,
                               $code_prod3,
                               $doserec_prod1,
                               $doserec_prod2,
                               $doserec_prod3,
                               $id_cli,
                               $id_entrep,
                               $id_mat,
                               $id_emp_chauf,
                               $id_emp_dec,
                               $jet_chant,
                               $lazyload = true) {}

/**
 * Compter les chantiers
 * @param $pdo PDO
 * @return int Nombre de chantiers
 */
public static function count (PDO $pdo) {}

/**
 * Requête de sélection
 * @param $pdo PDO
 * @param $where string|array
 * @param $orderby string|array
 * @param $limit string|array
 * @param $from string|array
 * @return PDOStatement
 */
protected static function _select (PDO $pdo,
                                    $where = null,

```

```

        $orderby = null,
        $limit = null,
        $from = null) {}

/**
 * Charger un(e) chantier
 * @param $pdo PDO
 * @param $idChantier int
 * @param $lazyload bool Activer le chargement fainéant ?
 * @return Chantier
 */
public static function load (PDO $pdo,
                            $idChantier,
                            $lazyload = true) {}

/**
 * Recharger les données depuis la base de données
 */
public function reload () {}

/**
 * Charger tous/toutes les chantiers
 * @param $pdo PDO
 * @param $lazyload bool Activer le chargement fainéant ?
 * @return Chantier[] Tableau de chantiers
 */
public static function loadAll (PDO $pdo,
                               $lazyload = true) {}

/**
 * Sélectionner tous/toutes les chantiers
 * @param $pdo PDO
 * @return PDOStatement
 */
public static function selectAll (PDO $pdo) {}

/**
 * Récupérer le/la chantier suivant(e) d'un jeu de résultats
 * @param $pdo PDO
 * @param $pdoStatement PDOStatement
 * @param $lazyload bool Activer le chargement fainéant ?
 * @return Chantier
 */
public static function fetch (PDO $pdo,
                             PDOStatement $pdoStatement,
                             $lazyload = true) {}

/**
 * Récupérer tous/toutes les chantiers d'un jeu de résultats
 * @param $pdo PDO
 * @param $pdoStatement PDOStatement
 * @param $lazyload bool Activer le chargement fainéant ?
 * @return Chantier[] Tableau de chantiers

```

```

*/
public static function fetchAll (PDO $pdo,
                                PDOStatement $pdoStatement,
                                $lazyload = true) {}

/**
 * Test d'égalité
 * @param $chantier Chantier
 * @return bool Les objets sont-ils égaux ?
 */
public function equals ($chantier) {}

/**
 * Vérifier que le/la chantier existe en base de données
 * @return bool Le/La chantier existe en base de données ?
 */
public function exists () {}

/**
 * Supprimer le/la chantier
 * @return bool Opération réussie ?
 */
public function delete () {}

/**
 * Mettre à jour un champ dans la base de données
 * @param $fields array
 * @param $values array
 * @return bool Opération réussie ?
 */
protected function _set ($fields,
                        $values) {}

/**
 * Mettre à jour tous les champs dans la base de données
 * @return bool Opération réussie ?
 */
public function update () {}

/**
 * Récupérer le/la idChantier
 * @return int
 */
public function getIdChantier () {}

/**
 * Récupérer le/la id_chant
 * @return int
 */
public function getId_chant () {}

/**
 * Définir le/la id_chant

```

```

* @param $id_chant int
* @param $execute bool Exécuter la requête update ?
* @return bool Opération réussie ?
*/
public function setId_chant ($id_chant,
                             $execute = true) {}

/**
* Récupérer le/la num_chant
* @return int
*/
public function getNum_chant () {}

/**
* Définir le/la num_chant
* @param $num_chant int
* @param $execute bool Exécuter la requête update ?
* @return bool Opération réussie ?
*/
public function setNum_chant ($num_chant,
                              $execute = true) {}

/**
* Récupérer le/la date_chant
* @return string
*/
public function getDate_chant () {}

/**
* Définir le/la date_chant
* @param $date_chant string
* @param $execute bool Exécuter la requête update ?
* @return bool Opération réussie ?
*/
public function setDate_chant ($date_chant,
                               $execute = true) {}

/**
* Récupérer le/la modif_chant
* @return int
*/
public function getModif_chant () {}

/**
* Définir le/la modif_chant
* @param $modif_chant int
* @param $execute bool Exécuter la requête update ?
* @return bool Opération réussie ?
*/
public function setModif_chant ($modif_chant,
                                $execute = true) {}

/**

```

```

* Récupérer le/la pac_chant
* @return int
*/
public function getPac_chant () {}

/**
* Définir le/la pac_chant
* @param $pac_chant int
* @param $execute bool Exécuter la requête update ?
* @return bool Opération réussie ?
*/
public function setPac_chant ($pac_chant,
                             $execute = true) {}

/**
* Récupérer le/la imp_chant
* @return int
*/
public function getImp_chant () {}

/**
* Définir le/la imp_chant
* @param $imp_chant int
* @param $execute bool Exécuter la requête update ?
* @return bool Opération réussie ?
*/
public function setImp_chant ($imp_chant,
                              $execute = true) {}

/**
* Récupérer le/la code_prod1
* @return string
*/
public function getCode_prod1 () {}

/**
* Définir le/la code_prod1
* @param $code_prod1 string
* @param $execute bool Exécuter la requête update ?
* @return bool Opération réussie ?
*/
public function setCode_prod1 ($code_prod1,
                               $execute = true) {}

/**
* Récupérer le/la code_prod2
* @return string
*/
public function getCode_prod2 () {}

/**
* Définir le/la code_prod2
* @param $code_prod2 string

```

```

* @param $execute bool Exécuter la requête update ?
* @return bool Opération réussie ?
*/
public function setCode_prod2 ($code_prod2,
                               $execute = true) {}

/**
* Récupérer le/la code_prod3
* @return string
*/
public function getCode_prod3 () {}

/**
* Définir le/la code_prod3
* @param $code_prod3 string
* @param $execute bool Exécuter la requête update ?
* @return bool Opération réussie ?
*/
public function setCode_prod3 ($code_prod3,
                               $execute = true) {}

/**
* Récupérer le/la doserec_prod1
* @return string
*/
public function getDoserec_prod1 () {}

/**
* Définir le/la doserec_prod1
* @param $doserec_prod1 string
* @param $execute bool Exécuter la requête update ?
* @return bool Opération réussie ?
*/
public function setDoserec_prod1 ($doserec_prod1,
                                  $execute = true) {}

/**
* Récupérer le/la doserec_prod2
* @return string
*/
public function getDoserec_prod2 () {}

/**
* Définir le/la doserec_prod2
* @param $doserec_prod2 string
* @param $execute bool Exécuter la requête update ?
* @return bool Opération réussie ?
*/
public function setDoserec_prod2 ($doserec_prod2,
                                  $execute = true) {}

/**
* Récupérer le/la doserec_prod3

```

```

* @return string
*/
public function getDoserec_prod3 () {}

/**
* Définir le/la doserec_prod3
* @param $doserec_prod3 string
* @param $execute bool Exécuter la requête update ?
* @return bool Opération réussie ?
*/
public function setDoserec_prod3 ($doserec_prod3,
                                $execute = true) {}

/**
* Récupérer le/la id_cli
* @return int
*/
public function getId_cli () {}

/**
* Définir le/la id_cli
* @param $id_cli int
* @param $execute bool Exécuter la requête update ?
* @return bool Opération réussie ?
*/
public function setId_cli ($id_cli,
                           $execute = true) {}

/**
* Récupérer le/la id_entrep
* @return string
*/
public function getId_entrep () {}

/**
* Définir le/la id_entrep
* @param $id_entrep string
* @param $execute bool Exécuter la requête update ?
* @return bool Opération réussie ?
*/
public function setId_entrep ($id_entrep,
                              $execute = true) {}

/**
* Récupérer le/la id_mat
* @return int
*/
public function getId_mat () {}

/**
* Définir le/la id_mat
* @param $id_mat int
* @param $execute bool Exécuter la requête update ?

```

```

* @return bool Opération réussie ?
*/
public function setId_mat ($id_mat,
                          $execute = true) {}

/**
* Récupérer le/la id_emp_chauf
* @return int
*/
public function getId_emp_chauf () {}

/**
* Définir le/la id_emp_chauf
* @param $id_emp_chauf int
* @param $execute bool Exécuter la requête update ?
* @return bool Opération réussie ?
*/
public function setId_emp_chauf ($id_emp_chauf,
                                 $execute = true) {}

/**
* Récupérer le/la id_emp_dec
* @return int
*/
public function getId_emp_dec () {}

/**
* Définir le/la id_emp_dec
* @param $id_emp_dec int
* @param $execute bool Exécuter la requête update ?
* @return bool Opération réussie ?
*/
public function setId_emp_dec ($id_emp_dec,
                              $execute = true) {}

/**
* ToString
* @return string Représentation de chantier sous la forme d'un string
*/
public function __toString () {}

/**
* Sérialiser
* @param $serialize bool Activer la sérialisation ?
* @return string Sérialisation du/de la chantier
*/
public function serialize ($serialize = true) {}

/**
* Désérialiser
* @param $pdo PDO
* @param $string string Sérialisation du/de la chantier
* @param $lazyload bool Activer le chargement fainéant ?

```

```
* @return Chantier
*/
public static function unserialize (PDO $pdo,
                                     $string,
                                     $lazyload = true) {}
}
```

```
+ www > model > classes > Chauffeur.php
```

```
/**
 * @name Base_Chauffeur
 * @version 02/05/2016 (dd/mm/yyyy)
 * @author Axel Jalabert
 */
abstract class Base_Chauffeur {

    const TABLENAME = 'CHAUFFEUR';

    const FIELDNAME_ID_EMP_CHAUF = 'id_emp_chauf';

    const FIELDNAME_NUM_CPHYT_CHAUF = 'num_cphyt_chauf';

    const FIELDNAME_HAB_CHAUF = 'hab_chauf';

    const FIELDNAME_DATE_EXP_HAB_CHAUF = 'date_exp_hab_chauf';

    const FIELDNAME_VISIBLE_CHAUF = 'visible_chauf';

    /** @var PDO */
    protected $_pdo;

    /** @var array tableau pour le chargement fainéant */
    protected static $_lazyload;

    /** @var int */
    protected $_id_emp_chauf;

    /** @var string */
    protected $_num_cphyt_chauf;

    /** @var int */
    protected $_hab_chauf;

    /** @var int */
    protected $_date_exp_hab_chauf;

    /** @var int */
    protected $_visible_chauf;

    /**
     * Construire un(e) chauffeur
     * @param $pdo PDO
     * @param $id_emp_chauf int
     * @param $num_cphyt_chauf string
     * @param $hab_chauf int
     * @param $date_exp_hab_chauf int
     * @param $visible_chauf int
     * @param $lazyload bool Activer le chargement fainéant ?
     */
    protected function __construct (PDO $pdo,
```

```

        $id_emp_chauf,
        $num_cphyt_chauf,
        $hab_chauf,
        $date_exp_hab_chauf,
        $visible_chauf,
        $lazyload = true) {}

/**
 * Créer un(e) chauffeur
 * @param $pdo PDO
 * @param $id_emp_chauf int
 * @param $num_cphyt_chauf string
 * @param $hab_chauf int
 * @param $date_exp_hab_chauf int
 * @param $visible_chauf int
 * @param $lazyload bool Activer le chargement fainéant ?
 * @return Chauffeur
 */
public static function create (PDO $pdo,
                               $id_emp_chauf,
                               $num_cphyt_chauf,
                               $hab_chauf,
                               $date_exp_hab_chauf,
                               $visible_chauf,
                               $lazyload = true) {}

/**
 * Compter les chauffeurs
 * @param $pdo PDO
 * @return int Nombre de chauffeurs
 */
public static function count (PDO $pdo) {}

/**
 * Requête de sélection
 * @param $pdo PDO
 * @param $where string|array
 * @param $orderby string|array
 * @param $limit string|array
 * @param $from string|array
 * @return PDOStatement
 */
protected static function _select (PDO $pdo,
                                    $where = null,
                                    $orderby = null,
                                    $limit = null,
                                    $from = null) {}

/**
 * Charger un(e) chauffeur
 * @param $pdo PDO
 * @param $id_emp_chauf int
 * @param $lazyload bool Activer le chargement fainéant ?

```

```

* @return Chauffeur
*/
public static function load (PDO $pdo,
                             $id_emp_chauf,
                             $lazyload = true) {}

/**
 * Recharger les données depuis la base de données
 */
public function reload () {}

/**
 * Charger tous/toutes les chauffeurs
 * @param $pdo PDO
 * @param $lazyload bool Activer le chargement fainéant ?
 * @return Chauffeur[] Tableau de chauffeurs
 */
public static function loadAll (PDO $pdo,
                                $lazyload = true) {}

/**
 * Sélectionner tous/toutes les chauffeurs
 * @param $pdo PDO
 * @return PDOStatement
 */
public static function selectAll (PDO $pdo) {}

/**
 * Récupérer le/la chauffeur suivant(e) d'un jeu de résultats
 * @param $pdo PDO
 * @param $pdoStatement PDOStatement
 * @param $lazyload bool Activer le chargement fainéant ?
 * @return Chauffeur
 */
public static function fetch (PDO $pdo,
                              PDOStatement $pdoStatement,
                              $lazyload = true) {}

/**
 * Récupérer tous/toutes les chauffeurs d'un jeu de résultats
 * @param $pdo PDO
 * @param $pdoStatement PDOStatement
 * @param $lazyload bool Activer le chargement fainéant ?
 * @return Chauffeur[] Tableau de chauffeurs
 */
public static function fetchAll (PDO $pdo,
                                  PDOStatement $pdoStatement,
                                  $lazyload = true) {}

/**
 * Test d'égalité
 * @param $chauffeur Chauffeur
 * @return bool Les objets sont-ils égaux ?

```

```

*/
public function equals ($chauffeur) {}

/**
 * Vérifier que le/la chauffeur existe en base de données
 * @return bool Le/La chauffeur existe en base de données ?
 */
public function exists () {}

/**
 * Supprimer le/la chauffeur
 * @return bool Opération réussie ?
 */
public function delete () {}

/**
 * Mettre à jour un champ dans la base de données
 * @param $fields array
 * @param $values array
 * @return bool Opération réussie ?
 */
protected function _set ($fields,
                        $values) {}

/**
 * Mettre à jour tous les champs dans la base de données
 * @return bool Opération réussie ?
 */
public function update () {}

/**
 * Récupérer le/la id_emp_chauf
 * @return int
 */
public function getId_emp_chauf () {}

/**
 * Récupérer le/la num_cphyt_chauf
 * @return string
 */
public function getNum_cphyt_chauf () {}

/**
 * Définir le/la num_cphyt_chauf
 * @param $num_cphyt_chauf string
 * @param $execute bool Exécuter la requête update ?
 * @return bool Opération réussie ?
 */
public function setNum_cphyt_chauf ($num_cphyt_chauf,
                                    $execute = true) {}

/**
 * Récupérer le/la hab_chauf

```

```

* @return int
*/
public function getHab_chauf () {}

/**
* Définir le/la hab_chauf
* @param $hab_chauf int
* @param $execute bool Exécuter la requête update ?
* @return bool Opération réussie ?
*/
public function setHab_chauf ($hab_chauf,
                             $execute = true) {}

/**
* Récupérer le/la date_exp_hab_chauf
* @return int
*/
public function getDate_exp_hab_chauf () {}

/**
* Définir le/la date_exp_hab_chauf
* @param $date_exp_hab_chauf int
* @param $execute bool Exécuter la requête update ?
* @return bool Opération réussie ?
*/
public function setDate_exp_hab_chauf ($date_exp_hab_chauf,
                                       $execute = true) {}

/**
* Récupérer le/la visible_chauf
* @return int
*/
public function getVisible_chauf () {}

/**
* Définir le/la visible_chauf
* @param $visible_chauf int
* @param $execute bool Exécuter la requête update ?
* @return bool Opération réussie ?
*/
public function setVisible_chauf ($visible_chauf,
                                  $execute = true) {}

/**
* ToString
* @return string Représentation de chauffeur sous la forme d'un string
*/
public function __toString () {}

/**
* Sérialiser
* @param $serialize bool Activer la sérialisation ?
* @return string Sérialisation du/de la chauffeur

```

```
*/
public function serialize ($serialize = true) {}

/**
 * Désérialiser
 * @param $pdo PDO
 * @param $string string Sérialisation du/de la chauffeur
 * @param $lazyload bool Activer le chargement fainéant ?
 * @return Chauffeur
 */
public static function unserialize (PDO $pdo,
                                     $string,
                                     $lazyload = true) {}

}
```

```
+ www > model > classes > Chauffeur.php
```

```
/**
 * @name Chauffeur
 * @version 02/05/2016 (dd/mm/yyyy)
 */
class Chauffeur extends Base_Chauffeur {

    const TABLENAME = 'CHAUFFEUR';

    const FIELDNAME_ID_EMP_CHAUF = 'id_emp_chauf';

    const FIELDNAME_NUM_CPHYT_CHAUF = 'num_cphyt_chauf';

    const FIELDNAME_HAB_CHAUF = 'hab_chauf';

    const FIELDNAME_DATE_EXP_HAB_CHAUF = 'date_exp_hab_chauf';

    const FIELDNAME_VISIBLE_CHAUF = 'visible_chauf';

    public static function chauffeurList (PDO $pdo,
                                           $entrep,
                                           $role = false) {}

    public static function chauffeurDelete (PDO $pdo,
                                             $id) {}

    public static function chauffeurInfo (PDO $pdo,
                                           $id,
                                           $entrep) {}

    public static function chauffeurAjoutAffichage (PDO $pdo,
                                                     $entrep,
                                                     $urlchauff) {}

    public static function chauffeurModif (Database $db,
                                           $entrep,
                                           $urlchauff,
                                           $quote) {}

    public static function chauffeurAjout (PDO $pdo,
                                           $entrep,
                                           $quote) {}

    public static function supprDef ($chauffeur) {}

}
```

```
+ www > model > classes > Chef_Entrep.php
```

```
/**
 * @name Base_Chef_Entrep
 * @version 06/05/2016 (dd/mm/yyyy)
 * @author Axel Jalabert
 */
abstract class Base_Chef_Entrep {

    const TABLENAME = 'CHEF_ENTREP';

    const FIELDNAME_ID_EMP_CE = 'id_emp_ce';

    const FIELDNAME_ADRES1_CE = 'adres1_ce';

    const FIELDNAME_ADRES2_CE = 'adres2_ce';

    const FIELDNAME_CP_CE = 'cp_ce';

    const FIELDNAME_VILLE_CE = 'ville_ce';

    /** @var PDO */
    protected $_pdo;

    /** @var array tableau pour le chargement fainéant */
    protected static $_lazyload;

    /** @var int */
    protected $_id_emp_ce;

    /** @var string */
    protected $_adres1_ce;

    /** @var string */
    protected $_adres2_ce;

    /** @var int */
    protected $_cp_ce;

    /** @var string */
    protected $_ville_ce;

    /**
     * Construire un(e) chef_entrep
     * @param $pdo PDO
     * @param $id_emp_ce int
     * @param $adres1_ce string
     * @param $adres2_ce string
     * @param $cp_ce int
     * @param $ville_ce string
     * @param $lazyload bool Activer le chargement fainéant ?
     */
    protected function __construct (PDO $pdo,
```

```

        $id_emp_ce,
        $adres1_ce,
        $adres2_ce,
        $cp_ce,
        $ville_ce,
        $lazyload = true) {}

/**
 * Créer un(e) chef_entrep
 * @param $pdo PDO
 * @param $id_emp_ce int
 * @param $adres1_ce string
 * @param $adres2_ce string
 * @param $cp_ce int
 * @param $ville_ce string
 * @param $lazyload bool Activer le chargement fainéant ?
 * @return Chef_Entrep
 */
public static function create (PDO $pdo,
                               $id_emp_ce,
                               $adres1_ce,
                               $adres2_ce,
                               $cp_ce,
                               $ville_ce,
                               $lazyload = true) {}

/**
 * Compter les chef_entreps
 * @param $pdo PDO
 * @return int Nombre de chef_entreps
 */
public static function count (PDO $pdo) {}

/**
 * Requête de sélection
 * @param $pdo PDO
 * @param $where string|array
 * @param $orderby string|array
 * @param $limit string|array
 * @param $from string|array
 * @return PDOStatement
 */
protected static function _select (PDO $pdo,
                                    $where = null,
                                    $orderby = null,
                                    $limit = null,
                                    $from = null) {}

/**
 * Charger un(e) chef_entrep
 * @param $pdo PDO
 * @param $id_emp_ce int
 * @param $lazyload bool Activer le chargement fainéant ?

```

```

* @return Chef_Entrep
*/
public static function load (PDO $pdo,
                             $id_emp_ce,
                             $lazyload = true) {}

/**
 * Recharger les données depuis la base de données
 */
public function reload () {}

/**
 * Charger tous/toutes les chef_entreps
 * @param $pdo PDO
 * @param $lazyload bool Activer le chargement fainéant ?
 * @return Chef_Entrep[] Tableau de chef_entreps
 */
public static function loadAll (PDO $pdo,
                                $lazyload = true) {}

/**
 * Sélectionner tous/toutes les chef_entreps
 * @param $pdo PDO
 * @return PDOStatement
 */
public static function selectAll (PDO $pdo) {}

/**
 * Récupérer le/la chef_entrep suivant(e) d'un jeu de résultats
 * @param $pdo PDO
 * @param $pdoStatement PDOStatement
 * @param $lazyload bool Activer le chargement fainéant ?
 * @return Chef_Entrep
 */
public static function fetch (PDO $pdo,
                              PDOStatement $pdoStatement,
                              $lazyload = true) {}

/**
 * Récupérer tous/toutes les chef_entreps d'un jeu de résultats
 * @param $pdo PDO
 * @param $pdoStatement PDOStatement
 * @param $lazyload bool Activer le chargement fainéant ?
 * @return Chef_Entrep[] Tableau de chef_entreps
 */
public static function fetchAll (PDO $pdo,
                                 PDOStatement $pdoStatement,
                                 $lazyload = true) {}

/**
 * Test d'égalité
 * @param $chef_entrep Chef_Entrep
 * @return bool Les objets sont-ils égaux ?

```

```

*/
public function equals ($chef_entrep) {}

/**
 * Vérifier que le/la chef_entrep existe en base de données
 * @return bool Le/La chef_entrep existe en base de données ?
 */
public function exists () {}

/**
 * Supprimer le/la chef_entrep
 * @return bool Opération réussie ?
 */
public function delete () {}

/**
 * Mettre à jour un champ dans la base de données
 * @param $fields array
 * @param $values array
 * @return bool Opération réussie ?
 */
protected function _set ($fields,
                        $values) {}

/**
 * Mettre à jour tous les champs dans la base de données
 * @return bool Opération réussie ?
 */
public function update () {}

/**
 * Récupérer le/la id_emp_ce
 * @return int
 */
public function getId_emp_ce () {}

/**
 * Récupérer le/la adres1_ce
 * @return string
 */
public function getAdres1_ce () {}

/**
 * Définir le/la adres1_ce
 * @param $adres1_ce string
 * @param $execute bool Exécuter la requête update ?
 * @return bool Opération réussie ?
 */
public function setAdres1_ce ($adres1_ce,
                            $execute = true) {}

/**
 * Récupérer le/la adres2_ce

```

```

* @return string
*/
public function getAdres2_ce () {}

/**
* Définir le/la adres2_ce
* @param $adres2_ce string
* @param $execute bool Exécuter la requête update ?
* @return bool Opération réussie ?
*/
public function setAdres2_ce ($adres2_ce,
                             $execute = true) {}

/**
* Récupérer le/la cp_ce
* @return int
*/
public function getCp_ce () {}

/**
* Définir le/la cp_ce
* @param $cp_ce int
* @param $execute bool Exécuter la requête update ?
* @return bool Opération réussie ?
*/
public function setCp_ce ($cp_ce,
                          $execute = true) {}

/**
* Récupérer le/la ville_ce
* @return string
*/
public function getVille_ce () {}

/**
* Définir le/la ville_ce
* @param $ville_ce string
* @param $execute bool Exécuter la requête update ?
* @return bool Opération réussie ?
*/
public function setVille_ce ($ville_ce,
                             $execute = true) {}

/**
* ToString
* @return string Représentation de chef_entrep sous la forme d'un string
*/
public function __toString () {}

/**
* Sérialiser
* @param $serialize bool Activer la sérialisation ?
* @return string Sérialisation du/de la chef_entrep

```

```
*/
public function serialize ($serialize = true) {}

/**
 * Désérialiser
 * @param $pdo PDO
 * @param $string string Sérialisation du/de la chef_entrep
 * @param $lazyload bool Activer le chargement fainéant ?
 * @return Chef_Entrep
 */
public static function unserialize (PDO $pdo,
                                     $string,
                                     $lazyload = true) {}

}
```

```
+ www > model > classes > Chef_Entrep.php
```

```
/**
 * @name Chef_Entrep
 * @version 04/05/2016 (dd/mm/yyyy)
 */
class Chef_Entrep extends Base_Chef_Entrep {

    const TABLENAME = 'CHEF_ENTREP';

    const FIELDNAME_ID_EMP_CE = 'id_emp_ce';

    const FIELDNAME_ADRES1_CE = 'adres1_ce';

    const FIELDNAME_ADRES2_CE = 'adres2_ce';

    const FIELDNAME_CP_CE = 'cp_ce';

    const FIELDNAME_VILLE_CE = 'ville_ce';

    public static function chefEntrepriseOrganigramme (PDO $pdo,
                                                         $id_entrep) {}

    public static function chefEntrepriseAffichage (PDO $pdo,
                                                     $id_ce) {}

    public static function chefEntrepriseModif (PDO $pdo,
                                                $id_ce,
                                                $quote) {}

    public static function recup_prenom_ce (PDO $pdo,
                                             $id_entrep) {}

}
```

```
+ www > model > classes > Client.php
```

```
/**
 * @name Base_Client
 * @version 10/05/2016 (dd/mm/yyyy)
 * @author Axel Jalabert
 */
abstract class Base_Client {

    const TABLENAME = 'CLIENT';

    const FIELDNAME_ID_CLI = 'id_cli';

    const FIELDNAME_NOM_CLI = 'nom_cli';

    const FIELDNAME_ADRES1_CLI = 'adres1_cli';

    const FIELDNAME_ADRES2_CLI = 'adres2_cli';

    const FIELDNAME_CP_CLI = 'cp_cli';

    const FIELDNAME_VILLE_CLI = 'ville_cli';

    const FIELDNAME_TEL_FIX_CLI = 'tel_fix_cli';

    const FIELDNAME_TEL_PORT_CLI = 'tel_port_cli';

    const FIELDNAME_FAX_CLI = 'fax_cli';

    const FIELDNAME_MAIL_CLI = 'mail_cli';

    const FIELDNAME_NUM_PAC_CLI = 'num_pac_cli';

    const FIELDNAME_VISIBLE_CLI = 'visible_cli';

    /** @var PDO */
    protected $_pdo;

    /** @var array tableau pour le chargement fainéant */
    protected static $_lazyload;

    /** @var int */
    protected $_id_cli;

    /** @var string */
    protected $_nom_cli;

    /** @var string */
    protected $_adres1_cli;

    /** @var string */
    protected $_adres2_cli;
}
```

```

/** @var string */
protected $_cp_cli;

/** @var string */
protected $_ville_cli;

/** @var string */
protected $_tel_fix_cli;

/** @var string */
protected $_tel_port_cli;

/** @var string */
protected $_fax_cli;

/** @var string */
protected $_mail_cli;

/** @var int */
protected $_num_pac_cli;

/** @var int */
protected $_visible_cli;

/**
 * Construire un(e) client
 * @param $pdo PDO
 * @param $id_cli int
 * @param $nom_cli string
 * @param $adres1_cli string
 * @param $adres2_cli string
 * @param $cp_cli string
 * @param $ville_cli string
 * @param $tel_fix_cli string
 * @param $tel_port_cli string
 * @param $fax_cli string
 * @param $mail_cli string
 * @param $num_pac_cli int
 * @param $visible_cli int
 * @param $lazyload bool Activer le chargement fainéant ?
 */
protected function __construct (PDO $pdo,
                                $id_cli,
                                $nom_cli,
                                $adres1_cli,
                                $adres2_cli,
                                $cp_cli,
                                $ville_cli,
                                $tel_fix_cli,
                                $tel_port_cli,
                                $fax_cli,
                                $mail_cli,
                                $num_pac_cli,

```

```

        $visible_cli,
        $lazyload = true) {}

/**
 * Créer un(e) client
 * @param $pdo PDO
 * @param $id_cli int
 * @param $nom_cli string
 * @param $adres1_cli string
 * @param $adres2_cli string
 * @param $cp_cli string
 * @param $ville_cli string
 * @param $tel_fix_cli string
 * @param $tel_port_cli string
 * @param $fax_cli string
 * @param $mail_cli string
 * @param $num_pac_cli int
 * @param $visible_cli int
 * @param $lazyload bool Activer le chargement fainéant ?
 * @return Client
 */
public static function create (PDO $pdo,
                               $id_cli,
                               $nom_cli,
                               $adres1_cli,
                               $adres2_cli,
                               $cp_cli,
                               $ville_cli,
                               $tel_fix_cli,
                               $tel_port_cli,
                               $fax_cli,
                               $mail_cli,
                               $num_pac_cli,
                               $visible_cli,
                               $lazyload = true) {}

/**
 * Compter les clients
 * @param $pdo PDO
 * @return int Nombre de clients
 */
public static function count (PDO $pdo) {}

/**
 * Requête de sélection
 * @param $pdo PDO
 * @param $where string|array
 * @param $orderby string|array
 * @param $limit string|array
 * @param $from string|array
 * @return PDOStatement
 */
protected static function _select (PDO $pdo,

```

```

        $where = null,
        $orderby = null,
        $limit = null,
        $from = null) {}

/**
 * Charger un(e) client
 * @param $pdo PDO
 * @param $id_cli int
 * @param $lazyload bool Activer le chargement fainéant ?
 * @return Client
 */
public static function load (PDO $pdo,
                            $id_cli,
                            $lazyload = true) {}

/**
 * Recharger les données depuis la base de données
 */
public function reload () {}

/**
 * Charger tous/toutes les clients
 * @param $pdo PDO
 * @param $lazyload bool Activer le chargement fainéant ?
 * @return Client[] Tableau de clients
 */
public static function loadAll (PDO $pdo,
                               $lazyload = true) {}

/**
 * Sélectionner tous/toutes les clients
 * @param $pdo PDO
 * @return PDOStatement
 */
public static function selectAll (PDO $pdo) {}

/**
 * Récupérer le/la client suivant(e) d'un jeu de résultats
 * @param $pdo PDO
 * @param $pdoStatement PDOStatement
 * @param $lazyload bool Activer le chargement fainéant ?
 * @return Client
 */
public static function fetch (PDO $pdo,
                             PDOStatement $pdoStatement,
                             $lazyload = true) {}

/**
 * Récupérer tous/toutes les clients d'un jeu de résultats
 * @param $pdo PDO
 * @param $pdoStatement PDOStatement
 * @param $lazyload bool Activer le chargement fainéant ?

```

```

* @return Client[] Tableau de clients
*/
public static function fetchAll (PDO $pdo,
                                PDOStatement $pdoStatement,
                                $lazyload = true) {}

/**
 * Test d'égalité
 * @param $client Client
 * @return bool Les objets sont-ils égaux ?
 */
public function equals ($client) {}

/**
 * Vérifier que le/la client existe en base de données
 * @return bool Le/La client existe en base de données ?
 */
public function exists () {}

/**
 * Supprimer le/la client
 * @return bool Opération réussie ?
 */
public function delete () {}

/**
 * Mettre à jour un champ dans la base de données
 * @param $fields array
 * @param $values array
 * @return bool Opération réussie ?
 */
protected function _set ($fields,
                        $values) {}

/**
 * Mettre à jour tous les champs dans la base de données
 * @return bool Opération réussie ?
 */
public function update () {}

/**
 * Récupérer le/la id_cli
 * @return int
 */
public function getId_cli () {}

/**
 * Récupérer le/la nom_cli
 * @return string
 */
public function getNom_cli () {}

/**

```

```

* Définir le/la nom_cli
* @param $nom_cli string
* @param $execute bool Exécuter la requête update ?
* @return bool Opération réussie ?
*/
public function setNom_cli ($nom_cli,
                           $execute = true) {}

/**
* Récupérer le/la adres1_cli
* @return string
*/
public function getAdres1_cli () {}

/**
* Définir le/la adres1_cli
* @param $adres1_cli string
* @param $execute bool Exécuter la requête update ?
* @return bool Opération réussie ?
*/
public function setAdres1_cli ($adres1_cli,
                              $execute = true) {}

/**
* Récupérer le/la adres2_cli
* @return string
*/
public function getAdres2_cli () {}

/**
* Définir le/la adres2_cli
* @param $adres2_cli string
* @param $execute bool Exécuter la requête update ?
* @return bool Opération réussie ?
*/
public function setAdres2_cli ($adres2_cli,
                              $execute = true) {}

/**
* Récupérer le/la cp_cli
* @return string
*/
public function getCp_cli () {}

/**
* Définir le/la cp_cli
* @param $cp_cli string
* @param $execute bool Exécuter la requête update ?
* @return bool Opération réussie ?
*/
public function setCp_cli ($cp_cli,
                          $execute = true) {}

```

```

/**
 * Récupérer le/la ville_cli
 * @return string
 */
public function getVille_cli () {}

/**
 * Définir le/la ville_cli
 * @param $ville_cli string
 * @param $execute bool Exécuter la requête update ?
 * @return bool Opération réussie ?
 */
public function setVille_cli ($ville_cli,
                             $execute = true) {}

/**
 * Récupérer le/la tel_fix_cli
 * @return string
 */
public function getTel_fix_cli () {}

/**
 * Définir le/la tel_fix_cli
 * @param $tel_fix_cli string
 * @param $execute bool Exécuter la requête update ?
 * @return bool Opération réussie ?
 */
public function setTel_fix_cli ($tel_fix_cli,
                               $execute = true) {}

/**
 * Récupérer le/la tel_port_cli
 * @return string
 */
public function getTel_port_cli () {}

/**
 * Définir le/la tel_port_cli
 * @param $tel_port_cli string
 * @param $execute bool Exécuter la requête update ?
 * @return bool Opération réussie ?
 */
public function setTel_port_cli ($tel_port_cli,
                                 $execute = true) {}

/**
 * Récupérer le/la fax_cli
 * @return string
 */
public function getFax_cli () {}

/**
 * Définir le/la fax_cli

```

```

* @param $fax_cli string
* @param $execute bool Exécuter la requête update ?
* @return bool Opération réussie ?
*/
public function setFax_cli ($fax_cli,
                           $execute = true) {}

/**
* Récupérer le/la mail_cli
* @return string
*/
public function getMail_cli () {}

/**
* Définir le/la mail_cli
* @param $mail_cli string
* @param $execute bool Exécuter la requête update ?
* @return bool Opération réussie ?
*/
public function setMail_cli ($mail_cli,
                             $execute = true) {}

/**
* Récupérer le/la num_pac_cli
* @return int
*/
public function getNum_pac_cli () {}

/**
* Définir le/la num_pac_cli
* @param $num_pac_cli int
* @param $execute bool Exécuter la requête update ?
* @return bool Opération réussie ?
*/
public function setNum_pac_cli ($num_pac_cli,
                                $execute = true) {}

/**
* Récupérer le/la visible_cli
* @return int
*/
public function getVisible_cli () {}

/**
* Définir le/la visible_cli
* @param $visible_cli int
* @param $execute bool Exécuter la requête update ?
* @return bool Opération réussie ?
*/
public function setVisible_cli ($visible_cli,
                                $execute = true) {}

/**

```

```

* ToString
* @return string Représentation de client sous la forme d'un string
*/
public function __toString () {}

/**
* Sériialiser
* @param $serialize bool Activer la sérialisation ?
* @return string Sérialisation du/de la client
*/
public function serialize ($serialize = true) {}

/**
* Désériialiser
* @param $pdo PDO
* @param $string string Sérialisation du/de la client
* @param $lazyload bool Activer le chargement fainéant ?
* @return Client
*/
public static function unserialize (PDO $pdo,
                                     $string,
                                     $lazyload = true) {}
}

```

```
+ www > model > classes > Clp_Phyto.php
```

```
/**
 * @name Base_Clp_Phyto
 * @version 26/05/2016 (dd/mm/yyyy)
 * @author Axel Jalabert
 */
abstract class Base_Clp_Phyto {

    const TABLENAME = 'CLP_PHYTO';

    const FIELDNAME_CODE_PROD = 'code_prod';

    const FIELDNAME_CODE_PHYTO = 'code_phyto';

    /** @var PDO */
    protected $_pdo;

    /** @var array tableau pour le chargement fainéant */
    protected static $_lazyload;

    /** @var string */
    protected $_code_prod;

    /** @var string */
    protected $_code_phyto;

    /**
     * Construire un(e) clp_phyto
     * @param $pdo PDO
     * @param $code_prod string
     * @param $code_phyto string
     * @param $lazyload bool Activer le chargement fainéant ?
     */
    protected function __construct (PDO $pdo,
                                     $code_prod,
                                     $code_phyto,
                                     $lazyload = true) {}

    /**
     * Créer un(e) clp_phyto
     * @param $pdo PDO
     * @param $code_prod string
     * @param $code_phyto string
     * @param $lazyload bool Activer le chargement fainéant ?
     * @return Clp_Phyto
     */
    public static function create (PDO $pdo,
                                   $code_prod,
                                   $code_phyto,
                                   $lazyload = true) {}

    /**
```

```

* Compter les clp_phytos
* @param $pdo PDO
* @return int Nombre de clp_phytos
*/
public static function count (PDO $pdo) {}

/**
* Requête de sélection
* @param $pdo PDO
* @param $where string|array
* @param $orderby string|array
* @param $limit string|array
* @param $from string|array
* @return PDOStatement
*/
protected static function _select (PDO $pdo,
                                     $where = null,
                                     $orderby = null,
                                     $limit = null,
                                     $from = null) {}

/**
* Charger un(e) clp_phyto
* @param $pdo PDO
* @param $code_prod string
* @param $lazyload bool Activer le chargement fainéant ?
* @return Clp_Phyto
*/
public static function load (PDO $pdo,
                             $code_prod,
                             $lazyload = true) {}

/**
* Recharger les données depuis la base de données
*/
public function reload () {}

/**
* Charger tous/toutes les clp_phytos
* @param $pdo PDO
* @param $lazyload bool Activer le chargement fainéant ?
* @return Clp_Phyto[] Tableau de clp_phytos
*/
public static function loadAll (PDO $pdo,
                                $lazyload = true) {}

/**
* Sélectionner tous/toutes les clp_phytos
* @param $pdo PDO
* @return PDOStatement
*/
public static function selectAll (PDO $pdo) {}

```

```

/**
 * Récupérer le/la clp_phyto suivant(e) d'un jeu de résultats
 * @param $pdo PDO
 * @param $pdoStatement PDOStatement
 * @param $lazyload bool Activer le chargement fainéant ?
 * @return Clp_Phyto
 */
public static function fetch (PDO $pdo,
                             PDOStatement $pdoStatement,
                             $lazyload = true) {}

/**
 * Récupérer tous/toutes les clp_phytos d'un jeu de résultats
 * @param $pdo PDO
 * @param $pdoStatement PDOStatement
 * @param $lazyload bool Activer le chargement fainéant ?
 * @return Clp_Phyto[] Tableau de clp_phytos
 */
public static function fetchAll (PDO $pdo,
                                 PDOStatement $pdoStatement,
                                 $lazyload = true) {}

/**
 * Test d'égalité
 * @param $clp_phyto Clp_Phyto
 * @return bool Les objets sont-ils égaux ?
 */
public function equals ($clp_phyto) {}

/**
 * Vérifier que le/la clp_phyto existe en base de données
 * @return bool Le/La clp_phyto existe en base de données ?
 */
public function exists () {}

/**
 * Supprimer le/la clp_phyto
 * @return bool Opération réussie ?
 */
public function delete () {}

/**
 * Mettre à jour un champ dans la base de données
 * @param $fields array
 * @param $values array
 * @return bool Opération réussie ?
 */
protected function _set ($fields,
                        $values) {}

/**
 * Mettre à jour tous les champs dans la base de données
 * @return bool Opération réussie ?
 */

```

```

*/
public function update () {}

/**
 * Récupérer le/la code_prod
 * @return string
 */
public function getCode_prod () {}

/**
 * Récupérer le/la code_phyto
 * @return string
 */
public function getCode_phyto () {}

/**
 * Définir le/la code_phyto
 * @param $code_phyto string
 * @param $execute bool Exécuter la requête update ?
 * @return bool Opération réussie ?
 */
public function setCode_phyto ($code_phyto,
                               $execute = true) {}

/**
 * ToString
 * @return string Représentation de clp_phyto sous la forme d'un string
 */
public function __toString () {}

/**
 * Sérialiser
 * @param $serialize bool Activer la sérialisation ?
 * @return string Sérialisation du/de la clp_phyto
 */
public function serialize ($serialize = true) {}

/**
 * Désérialiser
 * @param $pdo PDO
 * @param $string string Sérialisation du/de la clp_phyto
 * @param $lazyload bool Activer le chargement fainéant ?
 * @return Clp_Phyto
 */
public static function unserialize (PDO $pdo,
                                     $string,
                                     $lazyload = true) {}
}

```

```
+ www > model > classes > Clp_Phyto.php
```

```
/**
 * @name Clp_Phyto
 * @version 26/05/2016 (dd/mm/yyyy)
 */
class Clp_Phyto extends Base_Clp_Phyto {

    const TABLENAME = 'CLP_PHYTO';

    const FIELDNAME_CODE_PROD = 'code_prod';

    const FIELDNAME_CODE_PHYTO = 'code_phyto';

    public static function select_clp (PDO $pdo,
                                        $codeP) {}

    public static function infoClp (PDO $pdo,
                                    $prod,
                                    $num = 1) {}

}
```

```
+ www > model > classes > Commune.php
```

```
class Commune {  
  
    private $code_insee_commune;  
  
    private $code_postal_commune;  
  
    private $nom_commune;  
  
    public function setCodeInseeCommune ($code_insee_commune) {}  
  
    public function getCodeInseeCommune () {}  
  
    public function setCodePostalCommune ($code_postal_commune) {}  
  
    public function getCodePostalCommune () {}  
  
    public function setNomCommune ($nom_commune) {}  
  
    public function getNomCommune () {}  
  
}
```

```
+ www > model > classes > ContratChauf.php
```

```
/**
 * @name Model_Base_Contrat_Chauf
 * @version 02/05/2016 (dd/mm/yyyy)
 * @author Axel Jalabert
 */
abstract class Model_Base_Contrat_Chauf {

    const TABLENAME = 'CONTRAT_CHAUF';

    const FIELDNAME_NUM_CONTR_C = 'num_contr_c';

    const FIELDNAME_DATE_EMBAUCHE_CONTR_C = 'date_embauche_contr_c';

    const FIELDNAME_DATE_FIN_CONTR_C = 'date_fin_contr_c';

    const FIELDNAME_ID_ENTREP = 'id_entrep';

    const FIELDNAME_ID_EMP_CHAUF = 'id_emp_chauf';

    const FIELDNAME_LIB_TCONT = 'lib_tcont';

    /** @var PDO */
    protected $_pdo;

    /** @var array tableau pour le chargement fainéant */
    protected static $_lazyload;

    /** @var int */
    protected $_num_contr_c;

    /** @var int */
    protected $_date_embauche_contr_c;

    /** @var int */
    protected $_date_fin_contr_c;

    /** @var string */
    protected $_id_entrep;

    /** @var int */
    protected $_id_emp_chauf;

    /** @var string */
    protected $_lib_tcont;

    /**
     * Construire un(e) contrat_chauf
     * @param $pdo PDO
     * @param $num_contr_c int
     * @param $date_embauche_contr_c int
     * @param $date_fin_contr_c int
     */
}
```

```

* @param $id_entrep string
* @param $id_emp_chauf int
* @param $lib_tcont string
* @param $lazyload bool Activer le chargement fainéant ?
*/
protected function __construct (PDO $pdo,
                                $num_contr_c,
                                $date_embauche_contr_c,
                                $date_fin_contr_c,
                                $id_entrep,
                                $id_emp_chauf,
                                $lib_tcont,
                                $lazyload = true) {}

/**
 * Créer un(e) contrat_chauf
 * @param $pdo PDO
 * @param $num_contr_c int
 * @param $date_embauche_contr_c int
 * @param $date_fin_contr_c int
 * @param $id_entrep string
 * @param $id_emp_chauf int
 * @param $lib_tcont string
 * @param $lazyload bool Activer le chargement fainéant ?
 * @return Model_Contrat_Chauf
 */
public static function create (PDO $pdo,
                              $num_contr_c,
                              $date_embauche_contr_c,
                              $date_fin_contr_c,
                              $id_entrep,
                              $id_emp_chauf,
                              $lib_tcont,
                              $lazyload = true) {}

/**
 * Compter les contrat_chaufs
 * @param $pdo PDO
 * @return int Nombre de contrat_chaufs
 */
public static function count (PDO $pdo) {}

/**
 * Requête de sélection
 * @param $pdo PDO
 * @param $where string|array
 * @param $orderby string|array
 * @param $limit string|array
 * @param $from string|array
 * @return PDOStatement
 */
protected static function _select (PDO $pdo,
                                   $where = null,

```

```

        $orderby = null,
        $limit = null,
        $from = null) {}

/**
 * Charger un(e) contrat_chauf
 * @param $pdo PDO
 * @param $num_contr_c int
 * @param $lazyload bool Activer le chargement fainéant ?
 * @return Model_Contrat_Chauf
 */
public static function load (PDO $pdo,
                            $num_contr_c,
                            $lazyload = true) {}

/**
 * Recharger les données depuis la base de données
 */
public function reload () {}

/**
 * Charger tous/toutes les contrat_chaufs
 * @param $pdo PDO
 * @param $lazyload bool Activer le chargement fainéant ?
 * @return Model_Contrat_Chauf[] Tableau de contrat_chaufs
 */
public static function loadAll (PDO $pdo,
                               $lazyload = true) {}

/**
 * Sélectionner tous/toutes les contrat_chaufs
 * @param $pdo PDO
 * @return PDOStatement
 */
public static function selectAll (PDO $pdo) {}

/**
 * Récupérer le/la contrat_chauf suivant(e) d'un jeu de résultats
 * @param $pdo PDO
 * @param $pdoStatement PDOStatement
 * @param $lazyload bool Activer le chargement fainéant ?
 * @return Model_Contrat_Chauf
 */
public static function fetch (PDO $pdo,
                             PDOStatement $pdoStatement,
                             $lazyload = true) {}

/**
 * Récupérer tous/toutes les contrat_chaufs d'un jeu de résultats
 * @param $pdo PDO
 * @param $pdoStatement PDOStatement
 * @param $lazyload bool Activer le chargement fainéant ?
 * @return Model_Contrat_Chauf[] Tableau de contrat_chaufs

```

```

*/
public static function fetchAll (PDO $pdo,
                                PDOStatement $pdoStatement,
                                $lazyload = true) {}

/**
 * Test d'égalité
 * @param $contrat_chauf Model_Contrat_Chauf
 * @return bool Les objets sont-ils égaux ?
 */
public function equals ($contrat_chauf) {}

/**
 * Vérifier que le/la contrat_chauf existe en base de données
 * @return bool Le/La contrat_chauf existe en base de données ?
 */
public function exists () {}

/**
 * Supprimer le/la contrat_chauf
 * @return bool Opération réussie ?
 */
public function delete () {}

/**
 * Mettre à jour un champ dans la base de données
 * @param $fields array
 * @param $values array
 * @return bool Opération réussie ?
 */
protected function _set ($fields,
                          $values) {}

/**
 * Mettre à jour tous les champs dans la base de données
 * @return bool Opération réussie ?
 */
public function update () {}

/**
 * Récupérer le/la num_contr_c
 * @return int
 */
public function getNum_contr_c () {}

/**
 * Récupérer le/la date_embauche_contr_c
 * @return int
 */
public function getDate_embauche_contr_c () {}

/**
 * Définir le/la date_embauche_contr_c

```

```

* @param $date_embauche_contr_c int
* @param $execute bool Exécuter la requête update ?
* @return bool Opération réussie ?
*/
public function setDate_embauche_contr_c ($date_embauche_contr_c,
                                         $execute = true) {}

/**
* Récupérer le/la date_fin_contr_c
* @return int
*/
public function getDate_fin_contr_c () {}

/**
* Définir le/la date_fin_contr_c
* @param $date_fin_contr_c int
* @param $execute bool Exécuter la requête update ?
* @return bool Opération réussie ?
*/
public function setDate_fin_contr_c ($date_fin_contr_c,
                                     $execute = true) {}

/**
* Récupérer le/la id_entrep
* @return string
*/
public function getId_entrep () {}

/**
* Définir le/la id_entrep
* @param $id_entrep string
* @param $execute bool Exécuter la requête update ?
* @return bool Opération réussie ?
*/
public function setId_entrep ($id_entrep,
                              $execute = true) {}

/**
* Récupérer le/la id_emp_chauf
* @return int
*/
public function getId_emp_chauf () {}

/**
* Définir le/la id_emp_chauf
* @param $id_emp_chauf int
* @param $execute bool Exécuter la requête update ?
* @return bool Opération réussie ?
*/
public function setId_emp_chauf ($id_emp_chauf,
                                 $execute = true) {}

/**

```

```

* Récupérer le/la lib_tcont
* @return string
*/
public function getLib_tcont () {}

/**
* Définir le/la lib_tcont
* @param $lib_tcont string
* @param $execute bool Exécuter la requête update ?
* @return bool Opération réussie ?
*/
public function setLib_tcont ($lib_tcont,
                             $execute = true) {}

/**
* ToString
* @return string Représentation de contrat_chauf sous la forme d'un string
*/
public function __toString () {}

/**
* Sérialiser
* @param $serialize bool Activer la sérialisation ?
* @return string Sérialisation du/de la contrat_chauf
*/
public function serialize ($serialize = true) {}

/**
* Désérialiser
* @param $pdo PDO
* @param $string string Sérialisation du/de la contrat_chauf
* @param $lazyload bool Activer le chargement fainéant ?
* @return Model_Contrat_Chauf
*/
public static function unserialize (PDO $pdo,
                                    $string,
                                    $lazyload = true) {}

}

```

```
+ www > model > classes > ContratChauf.php
```

```
/**  
 * @name Model_Contrat_Chauf  
 * @version 02/05/2016 (dd/mm/yyyy)  
 */  
class Model_Contrat_Chauf extends Model_Base_Contrat_Chauf {  
  
    const TABLENAME = 'CONTRAT_CHAUF';  
  
    const FIELDNAME_NUM_CONTR_C = 'num_contr_c';  
  
    const FIELDNAME_DATE_EMBAUCHE_CONTR_C = 'date_embauche_contr_c';  
  
    const FIELDNAME_DATE_FIN_CONTR_C = 'date_fin_contr_c';  
  
    const FIELDNAME_ID_ENTREP = 'id_entrep';  
  
    const FIELDNAME_ID_EMP_CHAUF = 'id_emp_chauf';  
  
    const FIELDNAME_LIB_TCONT = 'lib_tcont';  
  
}
```

```
+ www > model > classes > ContratDesc.php
```

```
/**
 * @name Base_Contrat_Dec
 * @version 04/05/2016 (dd/mm/yyyy)
 * @author Axel Jalabert
 */
abstract class Base_Contrat_Dec {

    const TABLENAME = 'CONTRAT_DEC';

    const FIELDNAME_NUM_CONTR_D = 'num_contr_d';

    const FIELDNAME_DATE_EMBAUCHE_CONTR_D = 'date_embauche_contr_d';

    const FIELDNAME_DATE_FIN_CONTR_D = 'date_fin_contr_d';

    const FIELDNAME_LIB_TCONT = 'lib_tcont';

    const FIELDNAME_ID_ENTREP = 'id_entrep';

    const FIELDNAME_ID_EMP_CHAUF_DEC = 'id_emp_chauf_dec';

    /** @var PDO */
    protected $_pdo;

    /** @var array tableau pour le chargement fainéant */
    protected static $_lazyload;

    /** @var int */
    protected $_num_contr_d;

    /** @var string */
    protected $_date_embauche_contr_d;

    /** @var string */
    protected $_date_fin_contr_d;

    /** @var string */
    protected $_lib_tcont;

    /** @var string */
    protected $_id_entrep;

    /** @var int */
    protected $_id_emp_chauf_dec;

    /**
     * Construire un(e) contrat_dec
     * @param $pdo PDO
     * @param $num_contr_d int
     * @param $date_embauche_contr_d string
     * @param $date_fin_contr_d string
     */
}
```

```

* @param $lib_tcont string
* @param $id_entrep string
* @param $id_emp_chauf_dec int
* @param $lazyload bool Activer le chargement fainéant ?
*/
protected function __construct (PDO $pdo,
                                $num_contr_d,
                                $date_embauche_contr_d,
                                $date_fin_contr_d,
                                $lib_tcont,
                                $id_entrep,
                                $id_emp_chauf_dec,
                                $lazyload = true) {}

/**
 * Créer un(e) contrat_dec
 * @param $pdo PDO
 * @param $num_contr_d int
 * @param $date_embauche_contr_d string
 * @param $date_fin_contr_d string
 * @param $lib_tcont string
 * @param $id_entrep string
 * @param $id_emp_chauf_dec int
 * @param $lazyload bool Activer le chargement fainéant ?
 * @return Contrat_Dec
 */
public static function create (PDO $pdo,
                              $num_contr_d,
                              $date_embauche_contr_d,
                              $date_fin_contr_d,
                              $lib_tcont,
                              $id_entrep,
                              $id_emp_chauf_dec,
                              $lazyload = true) {}

/**
 * Compter les contrat_decs
 * @param $pdo PDO
 * @return int Nombre de contrat_decs
 */
public static function count (PDO $pdo) {}

/**
 * Requête de sélection
 * @param $pdo PDO
 * @param $where string|array
 * @param $orderby string|array
 * @param $limit string|array
 * @param $from string|array
 * @return PDOStatement
 */
protected static function _select (PDO $pdo,
                                   $where = null,

```

```

        $orderby = null,
        $limit = null,
        $from = null) {}

/**
 * Charger un(e) contrat_dec
 * @param $pdo PDO
 * @param $num_contr_d int
 * @param $lazyload bool Activer le chargement fainéant ?
 * @return Contrat_Dec
 */
public static function load (PDO $pdo,
                            $num_contr_d,
                            $lazyload = true) {}

/**
 * Recharger les données depuis la base de données
 */
public function reload () {}

/**
 * Charger tous/toutes les contrat_dec
 * @param $pdo PDO
 * @param $lazyload bool Activer le chargement fainéant ?
 * @return Contrat_Dec[] Tableau de contrat_dec
 */
public static function loadAll (PDO $pdo,
                               $lazyload = true) {}

/**
 * Sélectionner tous/toutes les contrat_dec
 * @param $pdo PDO
 * @return PDOStatement
 */
public static function selectAll (PDO $pdo) {}

/**
 * Récupérer le/la contrat_dec suivant(e) d'un jeu de résultats
 * @param $pdo PDO
 * @param $pdoStatement PDOStatement
 * @param $lazyload bool Activer le chargement fainéant ?
 * @return Contrat_Dec
 */
public static function fetch (PDO $pdo,
                             PDOStatement $pdoStatement,
                             $lazyload = true) {}

/**
 * Récupérer tous/toutes les contrat_dec d'un jeu de résultats
 * @param $pdo PDO
 * @param $pdoStatement PDOStatement
 * @param $lazyload bool Activer le chargement fainéant ?
 * @return Contrat_Dec[] Tableau de contrat_dec

```

```

*/
public static function fetchAll (PDO $pdo,
                                PDOStatement $pdoStatement,
                                $lazyload = true) {}

/**
 * Test d'égalité
 * @param $contrat_dec Contrat_Dec
 * @return bool Les objets sont-ils égaux ?
 */
public function equals ($contrat_dec) {}

/**
 * Vérifier que le/la contrat_dec existe en base de données
 * @return bool Le/La contrat_dec existe en base de données ?
 */
public function exists () {}

/**
 * Supprimer le/la contrat_dec
 * @return bool Opération réussie ?
 */
public function delete () {}

/**
 * Mettre à jour un champ dans la base de données
 * @param $fields array
 * @param $values array
 * @return bool Opération réussie ?
 */
protected function _set ($fields,
                        $values) {}

/**
 * Mettre à jour tous les champs dans la base de données
 * @return bool Opération réussie ?
 */
public function update () {}

/**
 * Récupérer le/la num_contr_d
 * @return int
 */
public function getNum_contr_d () {}

/**
 * Récupérer le/la date_embauche_contr_d
 * @return string
 */
public function getDate_embauche_contr_d () {}

/**
 * Définir le/la date_embauche_contr_d

```

```

* @param $date_embauche_contr_d string
* @param $execute bool Exécuter la requête update ?
* @return bool Opération réussie ?
*/
public function setDate_embauche_contr_d ($date_embauche_contr_d,
                                         $execute = true) {}

/**
* Récupérer le/la date_fin_contr_d
* @return string
*/
public function getDate_fin_contr_d () {}

/**
* Définir le/la date_fin_contr_d
* @param $date_fin_contr_d string
* @param $execute bool Exécuter la requête update ?
* @return bool Opération réussie ?
*/
public function setDate_fin_contr_d ($date_fin_contr_d,
                                     $execute = true) {}

/**
* Récupérer le/la lib_tcont
* @return string
*/
public function getLib_tcont () {}

/**
* Définir le/la lib_tcont
* @param $lib_tcont string
* @param $execute bool Exécuter la requête update ?
* @return bool Opération réussie ?
*/
public function setLib_tcont ($lib_tcont,
                             $execute = true) {}

/**
* Récupérer le/la id_entrep
* @return string
*/
public function getId_entrep () {}

/**
* Définir le/la id_entrep
* @param $id_entrep string
* @param $execute bool Exécuter la requête update ?
* @return bool Opération réussie ?
*/
public function setId_entrep ($id_entrep,
                              $execute = true) {}

/**

```

```

* Récupérer le/la id_emp_chauf_dec
* @return int
*/
public function getId_emp_chauf_dec () {}

/**
* Définir le/la id_emp_chauf_dec
* @param $id_emp_chauf_dec int
* @param $execute bool Exécuter la requête update ?
* @return bool Opération réussie ?
*/
public function setId_emp_chauf_dec ($id_emp_chauf_dec,
                                     $execute = true) {}

/**
* ToString
* @return string Représentation de contrat_dec sous la forme d'un string
*/
public function __toString () {}

/**
* Sérialiser
* @param $serialize bool Activer la sérialisation ?
* @return string Sérialisation du/de la contrat_dec
*/
public function serialize ($serialize = true) {}

/**
* Désérialiser
* @param $pdo PDO
* @param $string string Sérialisation du/de la contrat_dec
* @param $lazyload bool Activer le chargement fainéant ?
* @return Contrat_Dec
*/
public static function unserialize (PDO $pdo,
                                    $string,
                                    $lazyload = true) {}

}

```

```
+ www > model > classes > ContratDesc.php
```

```
/**
 * @name Contrat_Dec
 * @version 04/05/2016 (dd/mm/yyyy)
 */
class Contrat_Dec extends Base_Contrat_Dec {

    const TABLENAME = 'CONTRAT_DEC';

    const FIELDNAME_NUM_CONTR_D = 'num_contr_d';

    const FIELDNAME_DATE_EMBAUCHE_CONTR_D = 'date_embauche_contr_d';

    const FIELDNAME_DATE_FIN_CONTR_D = 'date_fin_contr_d';

    const FIELDNAME_LIB_TCONT = 'lib_tcont';

    const FIELDNAME_ID_ENTREP = 'id_entrep';

    const FIELDNAME_ID_EMP_CHAUF_DEC = 'id_emp_chauf_dec';

}
```

```
+ www > model > classes > Cookie.php
```

```
class Cookie {  
  
    private $id_cookie;  
  
    public function setIdCookie ($id_cookie) {}  
  
    public function getIdCookie () {}  
  
}
```

```
+ www > model > classes > Culture.php
```

```
class Culture {

    private $numero_culture;

    private $surface_culture;

    public function setNumeroCulture ($numero_culture) {}

    public function getNumeroCulture () {}

    public function setSurfaceCulture ($surface_culture) {}

    public function getSurfaceCulture () {}

    public static function supprDef ($cult) {}

    /**
     * NUM_CAD: Numéro cadastral de la parcelle
     * ID_ILOT: Id dans la base de données de l'ilot parent
     * NUM_CULT: Numéro de la culture
     * SURF_CULT: Surface de la culture
     * INSEE_COM: Numéro INSEE de la commune
     * CP_COMM: Code postal de la commune
     * NOM_COMM: Nom de la commune
     * VAR_CULT: Variété de la plantation de la culture
     */
    public static function cultureAjout (PDO $pdo,
                                         $quote) {}

}
```

```
+ www > model > classes > Date_Var.php
```

```
/**
 * @name Base_Date_Var
 * @version 19/07/2016 (dd/mm/yyyy)
 * @author Axel Jalabert
 */
abstract class Base_Date_Var {

    const TABLENAME = 'DATE_VAR';

    const FIELDNAME_IDDATE_VAR = 'iddate_var';

    const FIELDNAME_ID_CULT = 'id_cult';

    const FIELDNAME_CODE_VAR = 'code_var';

    const FIELDNAME_LIB_VAR = 'lib_var';

    const FIELDNAME_DATE_DV = 'date_dv';

    /** @var PDO */
    protected $_pdo;

    /** @var array tableau pour le chargement fainéant */
    protected static $_lazyload;

    /** @var int */
    protected $_idDate_var;

    /** @var string */
    protected $_id_cult;

    /** @var int */
    protected $_code_var;

    /** @var string */
    protected $_lib_var;

    /** @var string */
    protected $_date_dv;

    /**
     * Construire un(e) date_var
     * @param $pdo PDO
     * @param $idDate_var int
     * @param $id_cult string
     * @param $code_var int
     * @param $lib_var string
     * @param $date_dv string
     * @param $lazyload bool Activer le chargement fainéant ?
     */
    protected function __construct (PDO $pdo,
```

```

        $idDate_var,
        $id_cult,
        $code_var,
        $lib_var,
        $date_dv,
        $lazyload = true) {}

/**
 * Créer un(e) date_var
 * @param $pdo PDO
 * @param $id_cult string
 * @param $code_var int
 * @param $lib_var string
 * @param $date_dv string
 * @param $lazyload bool Activer le chargement fainéant ?
 * @return Date_Var
 */
public static function create (PDO $pdo,
                               $id_cult,
                               $code_var,
                               $lib_var,
                               $date_dv,
                               $lazyload = true) {}

/**
 * Compter les date_vars
 * @param $pdo PDO
 * @return int Nombre de date_vars
 */
public static function count (PDO $pdo) {}

/**
 * Requête de sélection
 * @param $pdo PDO
 * @param $where string|array
 * @param $orderby string|array
 * @param $limit string|array
 * @param $from string|array
 * @return PDOStatement
 */
protected static function _select (PDO $pdo,
                                    $where = null,
                                    $orderby = null,
                                    $limit = null,
                                    $from = null) {}

/**
 * Charger un(e) date_var
 * @param $pdo PDO
 * @param $idDate_var int
 * @param $lazyload bool Activer le chargement fainéant ?
 * @return Date_Var
 */

```

```

public static function load (PDO $pdo,
                            $idDate_var,
                            $lazyload = true) {}

/**
 * Recharger les données depuis la base de données
 */
public function reload () {}

/**
 * Charger tous/toutes les date_vars
 * @param $pdo PDO
 * @param $lazyload bool Activer le chargement fainéant ?
 * @return Date_Var[] Tableau de date_vars
 */
public static function loadAll (PDO $pdo,
                               $lazyload = true) {}

/**
 * Sélectionner tous/toutes les date_vars
 * @param $pdo PDO
 * @return PDOStatement
 */
public static function selectAll (PDO $pdo) {}

/**
 * Récupérer le/la date_var suivant(e) d'un jeu de résultats
 * @param $pdo PDO
 * @param $pdoStatement PDOStatement
 * @param $lazyload bool Activer le chargement fainéant ?
 * @return Date_Var
 */
public static function fetch (PDO $pdo,
                             PDOStatement $pdoStatement,
                             $lazyload = true) {}

/**
 * Récupérer tous/toutes les date_vars d'un jeu de résultats
 * @param $pdo PDO
 * @param $pdoStatement PDOStatement
 * @param $lazyload bool Activer le chargement fainéant ?
 * @return Date_Var[] Tableau de date_vars
 */
public static function fetchAll (PDO $pdo,
                                PDOStatement $pdoStatement,
                                $lazyload = true) {}

/**
 * Test d'égalité
 * @param $date_var Date_Var
 * @return bool Les objets sont-ils égaux ?
 */
public function equals ($date_var) {}

```

```

/**
 * Vérifier que le/la date_var existe en base de données
 * @return bool Le/La date_var existe en base de données ?
 */
public function exists () {}

/**
 * Supprimer le/la date_var
 * @return bool Opération réussie ?
 */
public function delete () {}

/**
 * Mettre à jour un champ dans la base de données
 * @param $fields array
 * @param $values array
 * @return bool Opération réussie ?
 */
protected function _set ($fields,
                          $values) {}

/**
 * Mettre à jour tous les champs dans la base de données
 * @return bool Opération réussie ?
 */
public function update () {}

/**
 * Récupérer le/la idDate_var
 * @return int
 */
public function getIdDate_var () {}

/**
 * Récupérer le/la id_cult
 * @return string
 */
public function getId_cult () {}

/**
 * Définir le/la id_cult
 * @param $id_cult string
 * @param $execute bool Exécuter la requête update ?
 * @return bool Opération réussie ?
 */
public function setId_cult ($id_cult,
                            $execute = true) {}

/**
 * Récupérer le/la code_var
 * @return int
 */
public function getCode_var () {}

```

```

/**
 * Définir le/la code_var
 * @param $code_var int
 * @param $execute bool Exécuter la requête update ?
 * @return bool Opération réussie ?
 */
public function setCode_var ($code_var,
                             $execute = true) {}

/**
 * Récupérer le/la lib_var
 * @return string
 */
public function getLib_var () {}

/**
 * Définir le/la lib_var
 * @param $lib_var string
 * @param $execute bool Exécuter la requête update ?
 * @return bool Opération réussie ?
 */
public function setLib_var ($lib_var,
                             $execute = true) {}

/**
 * Récupérer le/la date_dv
 * @return string
 */
public function getDate_dv () {}

/**
 * Définir le/la date_dv
 * @param $date_dv string
 * @param $execute bool Exécuter la requête update ?
 * @return bool Opération réussie ?
 */
public function setDate_dv ($date_dv,
                             $execute = true) {}

/**
 * ToString
 * @return string Représentation de date_var sous la forme d'un string
 */
public function __toString () {}

/**
 * Sérialiser
 * @param $serialize bool Activer la sérialisation ?
 * @return string Sérialisation du/de la date_var
 */
public function serialize ($serialize = true) {}

/**

```

```
* Désérialiser
* @param $pdo PDO
* @param $string string Sérialisation du/de la date_var
* @param $lazyload bool Activer le chargement fainéant ?
* @return Date_Var
*/
```

```
public static function unserialize (PDO $pdo,
                                     $string,
                                     $lazyload = true) {}
```

```
}
```

```
+ www > model > classes > Date_Var.php
```

```
/**
 * @name Date_Var
 * @version 19/07/2016 (dd/mm/yyyy)
 */
class Date_Var extends Base_Date_Var {

    const TABLENAME = 'DATE_VAR';

    const FIELDNAME_IDDATE_VAR = 'iddate_var';

    const FIELDNAME_ID_CULT = 'id_cult';

    const FIELDNAME_CODE_VAR = 'code_var';

    const FIELDNAME_LIB_VAR = 'lib_var';

    const FIELDNAME_DATE_DV = 'date_dv';

    public static function recupCodeVar (PDO $pdo,
                                         $id) {}

}
```

```
+ www > model > classes > Decideur.php
```

```
/**
 * @name Base_Decideur
 * @version 06/05/2016 (dd/mm/yyyy)
 * @author Axel Jalabert
 */
abstract class Base_Decideur {

    const TABLENAME = 'DECIDEUR';

    const FIELDNAME_ID_EMP_DEC = 'id_emp_dec';

    const FIELDNAME_NUM_CET_DEC = 'num_cet_dec';

    const FIELDNAME_DATE_VAL_CERT_DEC = 'date_val_cert_dec';

    const FIELDNAME_VISIBLE_DEC = 'visible_dec';

    /** @var PDO */
    protected $_pdo;

    /** @var array tableau pour le chargement fainéant */
    protected static $_lazyload;

    /** @var int */
    protected $_id_emp_dec;

    /** @var string */
    protected $_num_cet_dec;

    /** @var string */
    protected $_date_val_cert_dec;

    /** @var int */
    protected $_visible_dec;

    /**
     * Construire un(e) decideur
     * @param $pdo PDO
     * @param $id_emp_dec int
     * @param $num_cet_dec string
     * @param $date_val_cert_dec string
     * @param $visible_dec int
     * @param $lazyload bool Activer le chargement fainéant ?
     */
    protected function __construct (PDO $pdo,
                                     $id_emp_dec,
                                     $num_cet_dec,
                                     $date_val_cert_dec,
                                     $visible_dec,
                                     $lazyload = true) {}
}
```

```

/**
 * Créer un(e) decideur
 * @param $pdo PDO
 * @param $id_emp_dec int
 * @param $num_cet_dec string
 * @param $date_val_cert_dec string
 * @param $visible_dec int
 * @param $lazyload bool Activer le chargement fainéant ?
 * @return Decideur
 */
public static function create (PDO $pdo,
                               $id_emp_dec,
                               $num_cet_dec,
                               $date_val_cert_dec,
                               $visible_dec,
                               $lazyload = true) {}

/**
 * Compter les decideurs
 * @param $pdo PDO
 * @return int Nombre de decideurs
 */
public static function count (PDO $pdo) {}

/**
 * Requête de sélection
 * @param $pdo PDO
 * @param $where string|array
 * @param $orderby string|array
 * @param $limit string|array
 * @param $from string|array
 * @return PDOStatement
 */
protected static function _select (PDO $pdo,
                                    $where = null,
                                    $orderby = null,
                                    $limit = null,
                                    $from = null) {}

/**
 * Charger un(e) decideur
 * @param $pdo PDO
 * @param $id_emp_dec int
 * @param $lazyload bool Activer le chargement fainéant ?
 * @return Decideur
 */
public static function load (PDO $pdo,
                             $id_emp_dec,
                             $lazyload = true) {}

/**
 * Recharger les données depuis la base de données
 */

```

```

public function reload () {}

/**
 * Charger tous/toutes les decideurs
 * @param $pdo PDO
 * @param $lazyload bool Activer le chargement fainéant ?
 * @return Decideur[] Tableau de decideurs
 */
public static function loadAll (PDO $pdo,
                                $lazyload = true) {}

/**
 * Sélectionner tous/toutes les decideurs
 * @param $pdo PDO
 * @return PDOStatement
 */
public static function selectAll (PDO $pdo) {}

/**
 * Récupérer le/la decideur suivant(e) d'un jeu de résultats
 * @param $pdo PDO
 * @param $pdoStatement PDOStatement
 * @param $lazyload bool Activer le chargement fainéant ?
 * @return Decideur
 */
public static function fetch (PDO $pdo,
                              PDOStatement $pdoStatement,
                              $lazyload = true) {}

/**
 * Récupérer tous/toutes les decideurs d'un jeu de résultats
 * @param $pdo PDO
 * @param $pdoStatement PDOStatement
 * @param $lazyload bool Activer le chargement fainéant ?
 * @return Decideur[] Tableau de decideurs
 */
public static function fetchAll (PDO $pdo,
                                 PDOStatement $pdoStatement,
                                 $lazyload = true) {}

/**
 * Test d'égalité
 * @param $decideur Decideur
 * @return bool Les objets sont-ils égaux ?
 */
public function equals ($decideur) {}

/**
 * Vérifier que le/la decideur existe en base de données
 * @return bool Le/La decideur existe en base de données ?
 */
public function exists () {}

```

```

/**
 * Supprimer le/la decideur
 * @return bool Opération réussie ?
 */
public function delete () {}

/**
 * Mettre à jour un champ dans la base de données
 * @param $fields array
 * @param $values array
 * @return bool Opération réussie ?
 */
protected function _set ($fields,
                        $values) {}

/**
 * Mettre à jour tous les champs dans la base de données
 * @return bool Opération réussie ?
 */
public function update () {}

/**
 * Récupérer le/la id_emp_dec
 * @return int
 */
public function getId_emp_dec () {}

/**
 * Récupérer le/la num_cet_dec
 * @return string
 */
public function getNum_cet_dec () {}

/**
 * Définir le/la num_cet_dec
 * @param $num_cet_dec string
 * @param $execute bool Exécuter la requête update ?
 * @return bool Opération réussie ?
 */
public function setNum_cet_dec ($num_cet_dec,
                              $execute = true) {}

/**
 * Récupérer le/la date_val_cert_dec
 * @return string
 */
public function getDate_val_cert_dec () {}

/**
 * Définir le/la date_val_cert_dec
 * @param $date_val_cert_dec string
 * @param $execute bool Exécuter la requête update ?
 * @return bool Opération réussie ?

```

```

*/
public function setDate_val_cert_dec ($date_val_cert_dec,
                                     $execute = true) {}

/**
 * Récupérer le/la visible_dec
 * @return int
 */
public function getVisible_dec () {}

/**
 * Définir le/la visible_dec
 * @param $visible_dec int
 * @param $execute bool Exécuter la requête update ?
 * @return bool Opération réussie ?
 */
public function setVisible_dec ($visible_dec,
                               $execute = true) {}

/**
 * ToString
 * @return string Représentation de decideur sous la forme d'un string
 */
public function __toString () {}

/**
 * Sérialiser
 * @param $serialize bool Activer la sérialisation ?
 * @return string Sérialisation du/de la decideur
 */
public function serialize ($serialize = true) {}

/**
 * Désérialiser
 * @param $pdo PDO
 * @param $string string Sérialisation du/de la decideur
 * @param $lazyload bool Activer le chargement fainéant ?
 * @return Decideur
 */
public static function unserialize (PDO $pdo,
                                    $string,
                                    $lazyload = true) {}

}

```

```
+ www > model > classes > Decideur.php
```

```
/**
 * @name Decideur
 * @version 06/05/2016 (dd/mm/yyyy)
 */
class Decideur extends Base_Decideur {

    const TABLENAME = 'DECIDEUR';

    const FIELDNAME_ID_EMP_DEC = 'id_emp_dec';

    const FIELDNAME_NUM_CET_DEC = 'num_cet_dec';

    const FIELDNAME_DATE_VAL_CERT_DEC = 'date_val_cert_dec';

    const FIELDNAME_VISIBLE_DEC = 'visible_dec';

    public static function decideurList (PDO $pdo,
                                         $entrep,
                                         $role = false) {}

    public static function decideurDelete (PDO $pdo,
                                           $id) {}

    public static function decideurInfo (PDO $pdo,
                                         $id,
                                         $entrep) {}

    public static function decideurAjoutAffichage (PDO $pdo,
                                                    $entrep,
                                                    $urldec) {}

    public static function decideurModif ($db,
                                         $entrep,
                                         $urldec,
                                         $quote) {}

    public static function decideurAjout (PDO $pdo,
                                         $entrep,
                                         $quote) {}

    public static function supprDef ($decideur) {}

}
```

```
+ www > model > classes > Dosage.php
```

```
class Dosage {  
  
    private $dose_max_dosage;  
  
    private $nombre_appli_autorise_dosage;  
  
    public function setDoseMaxDosage ($dose_max_dosage) {}  
  
    public function getDoseMaxDosage () {}  
  
    public function setNombreAppliAutoriseDosage ($nombre_appli_autorise_dosage) {}  
  
    public function getCodeNombreAppliAutoriseDosage () {}  
  
}
```

```
+ www > model > classes > Employe.php
```

```
/**
 * @name Base_Employe
 * @version 02/05/2016 (dd/mm/yyyy)
 * @author Axel Jalabert
 */
abstract class Base_Employe {

    const TABLENAME = 'EMPLOYEE';

    const FIELDNAME_ID_EMP = 'id_emp';

    const FIELDNAME_NOM_EMP = 'nom_emp';

    const FIELDNAME_PREN_EMP = 'pren_emp';

    const FIELDNAME_DATE_NAISS_EMP = 'date_naiss_emp';

    const FIELDNAME_LIEU_NAISS_EMP = 'lieu_naiss_emp';

    const FIELDNAME_TEL_FIX_EMP = 'tel_fix_emp';

    const FIELDNAME_TEL_PORT_EMP = 'tel_port_emp';

    const FIELDNAME_FAX_EMP = 'fax_emp';

    const FIELDNAME_MAIL_EMP = 'mail_emp';

    /** @var PDO */
    protected $_pdo;

    /** @var array tableau pour le chargement fainéant */
    protected static $_lazyload;

    /** @var int */
    protected $_id_emp;

    /** @var string */
    protected $_nom_emp;

    /** @var string */
    protected $_pren_emp;

    /** @var int */
    protected $_date_naiss_emp;

    /** @var string */
    protected $_lieu_naiss_emp;

    /** @var string */
    protected $_tel_fix_emp;
```

```

/** @var string */
protected $_tel_port_emp;

/** @var string */
protected $_fax_emp;

/** @var string */
protected $_mail_emp;

/**
 * Construire un(e) employe
 * @param $pdo PDO
 * @param $id_emp int
 * @param $nom_emp string
 * @param $pren_emp string
 * @param $date_naiss_emp int
 * @param $lieu_naiss_emp string
 * @param $tel_fix_emp string
 * @param $tel_port_emp string
 * @param $fax_emp string
 * @param $mail_emp string
 * @param $lazyload bool Activer le chargement fainéant ?
 */
protected function __construct (PDO $pdo,
                                $id_emp,
                                $nom_emp,
                                $pren_emp,
                                $date_naiss_emp,
                                $lieu_naiss_emp,
                                $tel_fix_emp,
                                $tel_port_emp,
                                $fax_emp,
                                $mail_emp,
                                $lazyload = true) {}

/**
 * Créer un(e) employe
 * @param $pdo PDO
 * @param $id_emp int
 * @param $nom_emp string
 * @param $pren_emp string
 * @param $date_naiss_emp int
 * @param $lieu_naiss_emp string
 * @param $tel_fix_emp string
 * @param $tel_port_emp string
 * @param $fax_emp string
 * @param $mail_emp string
 * @param $lazyload bool Activer le chargement fainéant ?
 * @return Employe
 */
public static function create (PDO $pdo,
                              $id_emp,
                              $nom_emp,

```

```

        $pren_emp,
        $date_naiss_emp,
        $lieu_naiss_emp,
        $tel_fix_emp,
        $tel_port_emp,
        $fax_emp,
        $mail_emp,
        $lazyload = true) {}

/**
 * Compter les employes
 * @param $pdo PDO
 * @return int Nombre de employes
 */
public static function count (PDO $pdo) {}

/**
 * Requête de sélection
 * @param $pdo PDO
 * @param $where string|array
 * @param $orderby string|array
 * @param $limit string|array
 * @param $from string|array
 * @return PDOStatement
 */
protected static function _select (PDO $pdo,
                                    $where = null,
                                    $orderby = null,
                                    $limit = null,
                                    $from = null) {}

/**
 * Charger un(e) employe
 * @param $pdo PDO
 * @param $id_emp int
 * @param $lazyload bool Activer le chargement fainéant ?
 * @return Employe
 */
public static function load (PDO $pdo,
                            $id_emp,
                            $lazyload = true) {}

/**
 * Recharger les données depuis la base de données
 */
public function reload () {}

/**
 * Charger tous/toutes les employes
 * @param $pdo PDO
 * @param $lazyload bool Activer le chargement fainéant ?
 * @return Employe[] Tableau de employes
 */

```

```

public static function loadAll (PDO $pdo,
                                $lazyload = true) {}

/**
 * Sélectionner tous/toutes les employes
 * @param $pdo PDO
 * @return PDOStatement
 */
public static function selectAll (PDO $pdo) {}

/**
 * Récupérer le/la employe suivant(e) d'un jeu de résultats
 * @param $pdo PDO
 * @param $pdoStatement PDOStatement
 * @param $lazyload bool Activer le chargement fainéant ?
 * @return Employe
 */
public static function fetch (PDO $pdo,
                              PDOStatement $pdoStatement,
                              $lazyload = true) {}

/**
 * Récupérer tous/toutes les employes d'un jeu de résultats
 * @param $pdo PDO
 * @param $pdoStatement PDOStatement
 * @param $lazyload bool Activer le chargement fainéant ?
 * @return Employe[] Tableau de employes
 */
public static function fetchAll (PDO $pdo,
                                 PDOStatement $pdoStatement,
                                 $lazyload = true) {}

/**
 * Test d'égalité
 * @param $employe Employe
 * @return bool Les objets sont-ils égaux ?
 */
public function equals ($employe) {}

/**
 * Vérifier que le/la employe existe en base de données
 * @return bool Le/La employe existe en base de données ?
 */
public function exists () {}

/**
 * Supprimer le/la employe
 * @return bool Opération réussie ?
 */
public function delete () {}

/**
 * Mettre à jour un champ dans la base de données

```

```

* @param $fields array
* @param $values array
* @return bool Opération réussie ?
*/
protected function _set ($fields,
                        $values) {}

/**
 * Mettre à jour tous les champs dans la base de données
 * @return bool Opération réussie ?
 */
public function update () {}

/**
 * Récupérer le/la id_emp
 * @return int
 */
public function getId_emp () {}

/**
 * Récupérer le/la nom_emp
 * @return string
 */
public function getNom_emp () {}

/**
 * Définir le/la nom_emp
 * @param $nom_emp string
 * @param $execute bool Exécuter la requête update ?
 * @return bool Opération réussie ?
 */
public function setNom_emp ($nom_emp,
                          $execute = true) {}

/**
 * Récupérer le/la pren_emp
 * @return string
 */
public function getPren_emp () {}

/**
 * Définir le/la pren_emp
 * @param $pren_emp string
 * @param $execute bool Exécuter la requête update ?
 * @return bool Opération réussie ?
 */
public function setPren_emp ($pren_emp,
                          $execute = true) {}

/**
 * Récupérer le/la date_naiss_emp
 * @return int
 */

```

```

public function getDate_naiss_emp () {}

/**
 * Définir le/la date_naiss_emp
 * @param $date_naiss_emp int
 * @param $execute bool Exécuter la requête update ?
 * @return bool Opération réussie ?
 */
public function setDate_naiss_emp ($date_naiss_emp,
                                   $execute = true) {}

/**
 * Récupérer le/la lieu_naiss_emp
 * @return string
 */
public function getLieu_naiss_emp () {}

/**
 * Définir le/la lieu_naiss_emp
 * @param $lieu_naiss_emp string
 * @param $execute bool Exécuter la requête update ?
 * @return bool Opération réussie ?
 */
public function setLieu_naiss_emp ($lieu_naiss_emp,
                                   $execute = true) {}

/**
 * Récupérer le/la tel_fix_emp
 * @return string
 */
public function getTel_fix_emp () {}

/**
 * Définir le/la tel_fix_emp
 * @param $tel_fix_emp string
 * @param $execute bool Exécuter la requête update ?
 * @return bool Opération réussie ?
 */
public function setTel_fix_emp ($tel_fix_emp,
                                $execute = true) {}

/**
 * Récupérer le/la tel_port_emp
 * @return string
 */
public function getTel_port_emp () {}

/**
 * Définir le/la tel_port_emp
 * @param $tel_port_emp string
 * @param $execute bool Exécuter la requête update ?
 * @return bool Opération réussie ?
 */

```

```

public function setTel_port_emp ($tel_port_emp,
                                $execute = true) {}

/**
 * Récupérer le/la fax_emp
 * @return string
 */
public function getFax_emp () {}

/**
 * Définir le/la fax_emp
 * @param $fax_emp string
 * @param $execute bool Exécuter la requête update ?
 * @return bool Opération réussie ?
 */
public function setFax_emp ($fax_emp,
                            $execute = true) {}

/**
 * Récupérer le/la mail_emp
 * @return string
 */
public function getMail_emp () {}

/**
 * Définir le/la mail_emp
 * @param $mail_emp string
 * @param $execute bool Exécuter la requête update ?
 * @return bool Opération réussie ?
 */
public function setMail_emp ($mail_emp,
                             $execute = true) {}

/**
 * ToString
 * @return string Représentation de employe sous la forme d'un string
 */
public function __toString () {}

/**
 * Sérialiser
 * @param $serialize bool Activer la sérialisation ?
 * @return string Sérialisation du/de la employe
 */
public function serialize ($serialize = true) {}

/**
 * Désérialiser
 * @param $pdo PDO
 * @param $string string Sérialisation du/de la employe
 * @param $lazyload bool Activer le chargement fainéant ?
 * @return Employe
 */

```

```
public static function unserialize (PDO $pdo,  
    $string,  
    $lazyload = true) {}  
  
}
```

```
+ www > model > classes > Employe.php
```

```
/**
 * @name Employe
 * @version 02/05/2016 (dd/mm/yyyy)
 */
class Employe extends Base_Employe {

    const TABLENAME = 'EMPLOYEE';

    const FIELDNAME_ID_EMP = 'id_emp';

    const FIELDNAME_NOM_EMP = 'nom_emp';

    const FIELDNAME_PREN_EMP = 'pren_emp';

    const FIELDNAME_DATE_NAISS_EMP = 'date_naiss_emp';

    const FIELDNAME_LIEU_NAISS_EMP = 'lieu_naiss_emp';

    const FIELDNAME_TEL_FIX_EMP = 'tel_fix_emp';

    const FIELDNAME_TEL_PORT_EMP = 'tel_port_emp';

    const FIELDNAME_FAX_EMP = 'fax_emp';

    const FIELDNAME_MAIL_EMP = 'mail_emp';

    public static function employeAffichage (PDO $pdo,
                                             $id_ce) {}

    public static function employeModif (PDO $pdo,
                                         $id_ce,
                                         $quote) {}

    public static function info_ce (PDO $pdo,
                                    $id_emp_ce) {}

}
```

```
+ www > model > classes > Entreprise.php
```

```
/**
 * @name Base_Entreprise
 * @version 02/06/2016 (dd/mm/yyyy)
 * @author Axel Jalabert
 */
abstract class Base_Entreprise {

    const TABLENAME = 'ENTREPRISE';

    const FIELDNAME_ID_ENTREP = 'id_entrep';

    const FIELDNAME_RS_ENTREP = 'rs_entrep';

    const FIELDNAME_SIRET_ENTREP = 'siret_entrep';

    const FIELDNAME_TVA_ENTREP = 'tva_entrep';

    const FIELDNAME_AGR_ENTREP = 'agr_entrep';

    const FIELDNAME_ADRES1_ENTREP = 'adres1_entrep';

    const FIELDNAME_ADRES2_ENTREP = 'adres2_entrep';

    const FIELDNAME_CP_ENTREP = 'cp_entrep';

    const FIELDNAME_VILLE_ENTREP = 'ville_entrep';

    const FIELDNAME_TEL_FIX_ENTREP = 'tel_fix_entrep';

    const FIELDNAME_TEL_PORT_ENTREP = 'tel_port_entrep';

    const FIELDNAME_FAX_ENTREP = 'fax_entrep';

    const FIELDNAME_MAIL_ENTREP = 'mail_entrep';

    const FIELDNAME_MDP_ENTREP = 'mdp_entrep';

    const FIELDNAME_RIB_ENTREP = 'rib_entrep';

    const FIELDNAME_ID_EMP_CE = 'id_emp_ce';

    const FIELDNAME_ROLE_ENTREP = 'role_entrep';

    /** @var PDO */
    protected $_pdo;

    /** @var array tableau pour le chargement fainéant */
    protected static $_lazyload;

    /** @var string */
    protected $_id_entrep;
```

```

/** @var string */
protected $_rs_entrep;

/** @var int */
protected $_siret_entrep;

/** @var string */
protected $_tva_entrep;

/** @var string */
protected $_agr_entrep;

/** @var string */
protected $_adres1_entrep;

/** @var string */
protected $_adres2_entrep;

/** @var int */
protected $_cp_entrep;

/** @var string */
protected $_ville_entrep;

/** @var string */
protected $_tel_fix_entrep;

/** @var string */
protected $_tel_port_entrep;

/** @var string */
protected $_fax_entrep;

/** @var string */
protected $_mail_entrep;

/** @var string */
protected $_mdp_entrep;

/** @var string */
protected $_rib_entrep;

/** @var int */
protected $_id_emp_ce;

/** @var int */
protected $_role_entrep;

/**
 * Construire un(e) entreprise
 * @param $pdo PDO
 * @param $id_entrep string
 * @param $rs_entrep string

```

```

* @param $siret_entrep int
* @param $tva_entrep string
* @param $agr_entrep string
* @param $adres1_entrep string
* @param $adres2_entrep string
* @param $cp_entrep int
* @param $ville_entrep string
* @param $tel_fix_entrep string
* @param $tel_port_entrep string
* @param $fax_entrep string
* @param $mail_entrep string
* @param $mdp_entrep string
* @param $rib_entrep string
* @param $id_emp_ce int
* @param $role_entrep int
* @param $lazyload bool Activer le chargement fainéant ?
*/

```

```

protected function __construct (PDO $pdo,
                                $id_entrep,
                                $rs_entrep,
                                $siret_entrep,
                                $tva_entrep,
                                $agr_entrep,
                                $adres1_entrep,
                                $adres2_entrep,
                                $cp_entrep,
                                $ville_entrep,
                                $tel_fix_entrep,
                                $tel_port_entrep,
                                $fax_entrep,
                                $mail_entrep,
                                $mdp_entrep,
                                $rib_entrep,
                                $id_emp_ce,
                                $role_entrep,
                                $lazyload = true) {}

```

```

/**

```

```

* Créer un(e) entreprise
* @param $pdo PDO
* @param $id_entrep string
* @param $rs_entrep string
* @param $siret_entrep int
* @param $tva_entrep string
* @param $agr_entrep string
* @param $adres1_entrep string
* @param $adres2_entrep string
* @param $cp_entrep int
* @param $ville_entrep string
* @param $tel_fix_entrep string
* @param $tel_port_entrep string
* @param $fax_entrep string
* @param $mail_entrep string

```

```

* @param $mdp_entrep string
* @param $rib_entrep string
* @param $id_emp_ce int
* @param $role_entrep int
* @param $lazyload bool Activer le chargement fainéant ?
* @return Entreprise
*/
public static function create (PDO $pdo,
                               $id_entrep,
                               $rs_entrep,
                               $siret_entrep,
                               $tva_entrep,
                               $agr_entrep,
                               $adres1_entrep,
                               $adres2_entrep,
                               $cp_entrep,
                               $ville_entrep,
                               $tel_fix_entrep,
                               $tel_port_entrep,
                               $fax_entrep,
                               $mail_entrep,
                               $mdp_entrep,
                               $rib_entrep,
                               $id_emp_ce,
                               $role_entrep,
                               $lazyload = true) {}

/**
 * Compter les entreprises
 * @param $pdo PDO
 * @return int Nombre de entreprises
 */
public static function count (PDO $pdo) {}

/**
 * Requête de sélection
 * @param $pdo PDO
 * @param $where string|array
 * @param $orderby string|array
 * @param $limit string|array
 * @param $from string|array
 * @return PDOStatement
 */
protected static function _select (PDO $pdo,
                                    $where = null,
                                    $orderby = null,
                                    $limit = null,
                                    $from = null) {}

/**
 * Charger un(e) entreprise
 * @param $pdo PDO
 * @param $id_entrep string

```

```

* @param $lazyload bool Activer le chargement fainéant ?
* @return Entreprise
*/
public static function load (PDO $pdo,
                            $id_entrep,
                            $lazyload = true) {}

/**
* Recharger les données depuis la base de données
*/
public function reload () {}

/**
* Charger tous/toutes les entreprises
* @param $pdo PDO
* @param $lazyload bool Activer le chargement fainéant ?
* @return Entreprise[] Tableau de entreprises
*/
public static function loadAll (PDO $pdo,
                               $lazyload = true) {}

/**
* Sélectionner tous/toutes les entreprises
* @param $pdo PDO
* @return PDOStatement
*/
public static function selectAll (PDO $pdo) {}

/**
* Récupérer le/la entreprise suivant(e) d'un jeu de résultats
* @param $pdo PDO
* @param $pdoStatement PDOStatement
* @param $lazyload bool Activer le chargement fainéant ?
* @return Entreprise
*/
public static function fetch (PDO $pdo,
                             PDOStatement $pdoStatement,
                             $lazyload = true) {}

/**
* Récupérer tous/toutes les entreprises d'un jeu de résultats
* @param $pdo PDO
* @param $pdoStatement PDOStatement
* @param $lazyload bool Activer le chargement fainéant ?
* @return Entreprise[] Tableau de entreprises
*/
public static function fetchAll (PDO $pdo,
                                PDOStatement $pdoStatement,
                                $lazyload = true) {}

/**
* Test d'égalité
* @param $entreprise Entreprise

```

```

* @return bool Les objets sont-ils égaux ?
*/
public function equals ($entreprise) {}

/**
 * Vérifier que le/la entreprise existe en base de données
 * @return bool Le/La entreprise existe en base de données ?
 */
public function exists () {}

/**
 * Supprimer le/la entreprise
 * @return bool Opération réussie ?
 */
public function delete () {}

/**
 * Mettre à jour un champ dans la base de données
 * @param $fields array
 * @param $values array
 * @return bool Opération réussie ?
 */
protected function _set ($fields,
                        $values) {}

/**
 * Mettre à jour tous les champs dans la base de données
 * @return bool Opération réussie ?
 */
public function update () {}

/**
 * Récupérer le/la id_entrep
 * @return string
 */
public function getId_entrep () {}

/**
 * Récupérer le/la rs_entrep
 * @return string
 */
public function getRs_entrep () {}

/**
 * Définir le/la rs_entrep
 * @param $rs_entrep string
 * @param $execute bool Exécuter la requête update ?
 * @return bool Opération réussie ?
 */
public function setRs_entrep ($rs_entrep,
                            $execute = true) {}

/**

```

```

* Récupérer le/la siret_entrep
* @return int
*/
public function getSiret_entrep () {}

/**
* Définir le/la siret_entrep
* @param $siret_entrep int
* @param $execute bool Exécuter la requête update ?
* @return bool Opération réussie ?
*/
public function setSiret_entrep ($siret_entrep,
                                $execute = true) {}

/**
* Récupérer le/la tva_entrep
* @return string
*/
public function getTva_entrep () {}

/**
* Définir le/la tva_entrep
* @param $tva_entrep string
* @param $execute bool Exécuter la requête update ?
* @return bool Opération réussie ?
*/
public function setTva_entrep ($tva_entrep,
                                $execute = true) {}

/**
* Récupérer le/la agr_entrep
* @return string
*/
public function getAgr_entrep () {}

/**
* Définir le/la agr_entrep
* @param $agr_entrep string
* @param $execute bool Exécuter la requête update ?
* @return bool Opération réussie ?
*/
public function setAgr_entrep ($agr_entrep,
                                $execute = true) {}

/**
* Récupérer le/la adres1_entrep
* @return string
*/
public function getAdres1_entrep () {}

/**
* Définir le/la adres1_entrep
* @param $adres1_entrep string

```

```

* @param $execute bool Exécuter la requête update ?
* @return bool Opération réussie ?
*/
public function setAdres1_entrep ($adres1_entrep,
                                $execute = true) {}

/**
* Récupérer le/la adres2_entrep
* @return string
*/
public function getAdres2_entrep () {}

/**
* Définir le/la adres2_entrep
* @param $adres2_entrep string
* @param $execute bool Exécuter la requête update ?
* @return bool Opération réussie ?
*/
public function setAdres2_entrep ($adres2_entrep,
                                $execute = true) {}

/**
* Récupérer le/la cp_entrep
* @return int
*/
public function getCp_entrep () {}

/**
* Définir le/la cp_entrep
* @param $cp_entrep int
* @param $execute bool Exécuter la requête update ?
* @return bool Opération réussie ?
*/
public function setCp_entrep ($cp_entrep,
                              $execute = true) {}

/**
* Récupérer le/la ville_entrep
* @return string
*/
public function getVille_entrep () {}

/**
* Définir le/la ville_entrep
* @param $ville_entrep string
* @param $execute bool Exécuter la requête update ?
* @return bool Opération réussie ?
*/
public function setVille_entrep ($ville_entrep,
                                $execute = true) {}

/**
* Récupérer le/la tel_fix_entrep

```

```

* @return string
*/
public function getTel_fix_entrep () {}

/**
* Définir le/la tel_fix_entrep
* @param $tel_fix_entrep string
* @param $execute bool Exécuter la requête update ?
* @return bool Opération réussie ?
*/
public function setTel_fix_entrep ($tel_fix_entrep,
                                $execute = true) {}

/**
* Récupérer le/la tel_port_entrep
* @return string
*/
public function getTel_port_entrep () {}

/**
* Définir le/la tel_port_entrep
* @param $tel_port_entrep string
* @param $execute bool Exécuter la requête update ?
* @return bool Opération réussie ?
*/
public function setTel_port_entrep ($tel_port_entrep,
                                $execute = true) {}

/**
* Récupérer le/la fax_entrep
* @return string
*/
public function getFax_entrep () {}

/**
* Définir le/la fax_entrep
* @param $fax_entrep string
* @param $execute bool Exécuter la requête update ?
* @return bool Opération réussie ?
*/
public function setFax_entrep ($fax_entrep,
                              $execute = true) {}

/**
* Récupérer le/la mail_entrep
* @return string
*/
public function getMail_entrep () {}

/**
* Définir le/la mail_entrep
* @param $mail_entrep string
* @param $execute bool Exécuter la requête update ?

```

```

* @return bool Opération réussie ?
*/
public function setMail_entrep ($mail_entrep,
                               $execute = true) {}

/**
* Récupérer le/la mdp_entrep
* @return string
*/
public function getMdp_entrep () {}

/**
* Définir le/la mdp_entrep
* @param $mdp_entrep string
* @param $execute bool Exécuter la requête update ?
* @return bool Opération réussie ?
*/
public function setMdp_entrep ($mdp_entrep,
                               $execute = true) {}

/**
* Récupérer le/la rib_entrep
* @return string
*/
public function getRib_entrep () {}

/**
* Définir le/la rib_entrep
* @param $rib_entrep string
* @param $execute bool Exécuter la requête update ?
* @return bool Opération réussie ?
*/
public function setRib_entrep ($rib_entrep,
                               $execute = true) {}

/**
* Récupérer le/la id_emp_ce
* @return int
*/
public function getId_emp_ce () {}

/**
* Définir le/la id_emp_ce
* @param $id_emp_ce int
* @param $execute bool Exécuter la requête update ?
* @return bool Opération réussie ?
*/
public function setId_emp_ce ($id_emp_ce,
                              $execute = true) {}

/**
* Récupérer le/la role_entrep
* @return int

```

```

*/
public function getRole_entrep () {}

/**
 * Définir le/la role_entrep
 * @param $role_entrep int
 * @param $execute bool Exécuter la requête update ?
 * @return bool Opération réussie ?
 */
public function setRole_entrep ($role_entrep,
                                $execute = true) {}

/**
 * ToString
 * @return string Représentation de entreprise sous la forme d'un string
 */
public function __toString () {}

/**
 * Sérialiser
 * @param $serialize bool Activer la sérialisation ?
 * @return string Sérialisation du/de la entreprise
 */
public function serialize ($serialize = true) {}

/**
 * Désérialiser
 * @param $pdo PDO
 * @param $string string Sérialisation du/de la entreprise
 * @param $lazyload bool Activer le chargement fainéant ?
 * @return Entreprise
 */
public static function unserialize (PDO $pdo,
                                    $string,
                                    $lazyload = true) {}
}

```

```
+ www > model > classes > Entreprise.php
```

```
/**
 * @name Entreprise
 * @version 02/06/2016 (dd/mm/yyyy)
 */
class Entreprise extends Base_Entreprise {

    const TABLENAME = 'ENTREPRISE';

    const FIELDNAME_ID_ENTREP = 'id_entrep';

    const FIELDNAME_RS_ENTREP = 'rs_entrep';

    const FIELDNAME_SIRET_ENTREP = 'siret_entrep';

    const FIELDNAME_TVA_ENTREP = 'tva_entrep';

    const FIELDNAME_AGR_ENTREP = 'agr_entrep';

    const FIELDNAME_ADRES1_ENTREP = 'adres1_entrep';

    const FIELDNAME_ADRES2_ENTREP = 'adres2_entrep';

    const FIELDNAME_CP_ENTREP = 'cp_entrep';

    const FIELDNAME_VILLE_ENTREP = 'ville_entrep';

    const FIELDNAME_TEL_FIX_ENTREP = 'tel_fix_entrep';

    const FIELDNAME_TEL_PORT_ENTREP = 'tel_port_entrep';

    const FIELDNAME_FAX_ENTREP = 'fax_entrep';

    const FIELDNAME_MAIL_ENTREP = 'mail_entrep';

    const FIELDNAME_MDP_ENTREP = 'mdp_entrep';

    const FIELDNAME_RIB_ENTREP = 'rib_entrep';

    const FIELDNAME_ID_EMP_CE = 'id_emp_ce';

    const FIELDNAME_ROLE_ENTREP = 'role_entrep';

    public static function prestataireAffichage (PDO $pdo,
                                                $entrep) {}

    public static function recup_agr_entrep (PDO $pdo,
                                             $id_entrep) {}

    public static function prestataireModif (PDO $pdo,
                                             $entrep,
                                             $quote) {}
}
```

```
public static function create_entreprise (PDO $pdo,  
                                         $num_entrep,  
                                         $rs_entrep,  
                                         $siret_entrep,  
                                         $tva_entrep,  
                                         $agr_entrep,  
                                         $adres1_entrep,  
                                         $adres2_entrep,  
                                         $cp_entrep,  
                                         $ville_entrep,  
                                         $tel_fix_entrep,  
                                         $tel_port_entrep,  
                                         $fax_entrep,  
                                         $mail_entrep,  
                                         $mdp_entrep,  
                                         $rib_entrep,  
                                         $id_emp_ce,  
                                         $role_entrep) {}
```

```
public static function affichage_info_entrep (PDO $pdo) {}
```

```
public static function recup_mdp (PDO $pdo,  
                                  $id_entrep) {}
```

```
public static function recup_mail (PDO $pdo,  
                                   $id_entrep) {}
```

```
public static function recup_role_entrep (PDO $pdo,  
                                           $id_entrep) {}
```

```
public static function recup_rs_entrep (PDO $pdo,  
                                         $id_entrep) {}
```

```
public static function recup_id_emp_ce (PDO $pdo,  
                                         $id_entrep) {}
```

```
public static function supprDef ($entrep) {}
```

```
}
```

```
+ www > model > classes > Entretien.php
```

```
class Entretien {

    private $date_entretien;

    private $libelle_entretien;

    private $entretien_realise_par;

    private $numero_facture_entretien;

    public function setDateEntretien ($date_entretien) {}

    public function getDateEntretien () {}

    public function setLibelleEntretien ($libelle_entretien) {}

    public function getLibelleEntretien () {}

    public function setEntretienRealisePar ($entretien_realise_par) {}

    public function getEntretienRealisePar () {}

    public function setNumeroFactureEntretien ($numero_facture_entretien) {}

    public function getNumeroFactureEntretien () {}

    public static function supprDef ($entretien) {}

    /**
     * DATE_ENT: Date de l'entretien
     * LIB_ENT: Titre de l'entretien
     * REALIS_ENT: Nom de la personne qui a réalisé l'entretien
     * FACT_ENT: Numéro de facture de l'entretien
     * NATURE_ENT: Type d'entretien
     * ID_MAT: Numéro du matériel dans la base de données
     */
    public static function entretienAjout (PDO $pdo,
                                           $quote) {}

}
```

```
+ www > model > classes > Exploitation.php
```

```
/**
 * @name Base_Exploitation
 * @version 11/05/2016 (dd/mm/yyyy)
 * @author Axel Jalabert
 */
abstract class Base_Exploitation {

    const TABLENAME = 'EXPLOITATION';

    const FIELDNAME_SIRET_EXP = 'siret_exp';

    const FIELDNAME_NOM_EXP = 'nom_exp';

    const FIELDNAME_ADRES1_EXP = 'adres1_exp';

    const FIELDNAME_ADRES2_EXP = 'adres2_exp';

    const FIELDNAME_CP_EXP = 'cp_exp';

    const FIELDNAME_VILLE_EXP = 'ville_exp';

    const FIELDNAME_ID_CLI = 'id_cli';

    const FIELDNAME_VISIBLE_EXP = 'visible_exp';

    /** @var PDO */
    protected $_pdo;

    /** @var array tableau pour le chargement fainéant */
    protected static $_lazyload;

    /** @var int */
    protected $_siret_exp;

    /** @var string */
    protected $_nom_exp;

    /** @var string */
    protected $_adres1_exp;

    /** @var string */
    protected $_adres2_exp;

    /** @var int */
    protected $_cp_exp;

    /** @var string */
    protected $_ville_exp;

    /** @var int */
    protected $_id_cli;
```

```

/** @var int */
protected $_visible_exp;

/**
 * Construire un(e) exploitation
 * @param $pdo PDO
 * @param $siret_exp int
 * @param $nom_exp string
 * @param $adres1_exp string
 * @param $adres2_exp string
 * @param $cp_exp int
 * @param $ville_exp string
 * @param $id_cli int
 * @param $visible_exp int
 * @param $lazyload bool Activer le chargement fainéant ?
 */
protected function __construct (PDO $pdo,
                                $siret_exp,
                                $nom_exp,
                                $adres1_exp,
                                $adres2_exp,
                                $cp_exp,
                                $ville_exp,
                                $id_cli,
                                $visible_exp,
                                $lazyload = true) {}

/**
 * Créer un(e) exploitation
 * @param $pdo PDO
 * @param $siret_exp int
 * @param $nom_exp string
 * @param $adres1_exp string
 * @param $adres2_exp string
 * @param $cp_exp int
 * @param $ville_exp string
 * @param $id_cli int
 * @param $visible_exp int
 * @param $lazyload bool Activer le chargement fainéant ?
 * @return Exploitation
 */
public static function create (PDO $pdo,
                              $siret_exp,
                              $nom_exp,
                              $adres1_exp,
                              $adres2_exp,
                              $cp_exp,
                              $ville_exp,
                              $id_cli,
                              $visible_exp,
                              $lazyload = true) {}

/**

```

```

* Compter les exploitations
* @param $pdo PDO
* @return int Nombre de exploitations
*/
public static function count (PDO $pdo) {}

/**
* Requête de sélection
* @param $pdo PDO
* @param $where string|array
* @param $orderby string|array
* @param $limit string|array
* @param $from string|array
* @return PDOStatement
*/
protected static function _select (PDO $pdo,
                                     $where = null,
                                     $orderby = null,
                                     $limit = null,
                                     $from = null) {}

/**
* Charger un(e) exploitation
* @param $pdo PDO
* @param $siret_exp int
* @param $lazyload bool Activer le chargement fainéant ?
* @return Exploitation
*/
public static function load (PDO $pdo,
                             $siret_exp,
                             $lazyload = true) {}

/**
* Recharger les données depuis la base de données
*/
public function reload () {}

/**
* Charger tous/toutes les exploitations
* @param $pdo PDO
* @param $lazyload bool Activer le chargement fainéant ?
* @return Exploitation[] Tableau de exploitations
*/
public static function loadAll (PDO $pdo,
                                $lazyload = true) {}

/**
* Sélectionner tous/toutes les exploitations
* @param $pdo PDO
* @return PDOStatement
*/
public static function selectAll (PDO $pdo) {}

```

```

/**
 * Récupérer le/la exploitation suivant(e) d'un jeu de résultats
 * @param $pdo PDO
 * @param $pdoStatement PDOStatement
 * @param $lazyload bool Activer le chargement fainéant ?
 * @return Exploitation
 */
public static function fetch (PDO $pdo,
                             PDOStatement $pdoStatement,
                             $lazyload = true) {}

/**
 * Récupérer tous/toutes les exploitations d'un jeu de résultats
 * @param $pdo PDO
 * @param $pdoStatement PDOStatement
 * @param $lazyload bool Activer le chargement fainéant ?
 * @return Exploitation[] Tableau de exploitations
 */
public static function fetchAll (PDO $pdo,
                                PDOStatement $pdoStatement,
                                $lazyload = true) {}

/**
 * Test d'égalité
 * @param $exploitation Exploitation
 * @return bool Les objets sont-ils égaux ?
 */
public function equals ($exploitation) {}

/**
 * Vérifier que le/la exploitation existe en base de données
 * @return bool Le/La exploitation existe en base de données ?
 */
public function exists () {}

/**
 * Supprimer le/la exploitation
 * @return bool Opération réussie ?
 */
public function delete () {}

/**
 * Mettre à jour un champ dans la base de données
 * @param $fields array
 * @param $values array
 * @return bool Opération réussie ?
 */
protected function _set ($fields,
                        $values) {}

/**
 * Mettre à jour tous les champs dans la base de données
 * @return bool Opération réussie ?

```

```

*/
public function update () {}

/**
 * Récupérer le/la siret_exp
 * @return int
 */
public function getSiret_exp () {}

/**
 * Récupérer le/la nom_exp
 * @return string
 */
public function getNom_exp () {}

/**
 * Définir le/la nom_exp
 * @param $nom_exp string
 * @param $execute bool Exécuter la requête update ?
 * @return bool Opération réussie ?
 */
public function setNom_exp ($nom_exp,
                           $execute = true) {}

/**
 * Récupérer le/la adres1_exp
 * @return string
 */
public function getAdres1_exp () {}

/**
 * Définir le/la adres1_exp
 * @param $adres1_exp string
 * @param $execute bool Exécuter la requête update ?
 * @return bool Opération réussie ?
 */
public function setAdres1_exp ($adres1_exp,
                              $execute = true) {}

/**
 * Récupérer le/la adres2_exp
 * @return string
 */
public function getAdres2_exp () {}

/**
 * Définir le/la adres2_exp
 * @param $adres2_exp string
 * @param $execute bool Exécuter la requête update ?
 * @return bool Opération réussie ?
 */
public function setAdres2_exp ($adres2_exp,
                              $execute = true) {}

```

```

/**
 * Récupérer le/la cp_exp
 * @return int
 */
public function getCp_exp () {}

/**
 * Définir le/la cp_exp
 * @param $cp_exp int
 * @param $execute bool Exécuter la requête update ?
 * @return bool Opération réussie ?
 */
public function setCp_exp ($cp_exp,
                          $execute = true) {}

/**
 * Récupérer le/la ville_exp
 * @return string
 */
public function getVille_exp () {}

/**
 * Définir le/la ville_exp
 * @param $ville_exp string
 * @param $execute bool Exécuter la requête update ?
 * @return bool Opération réussie ?
 */
public function setVille_exp ($ville_exp,
                              $execute = true) {}

/**
 * Récupérer le/la id_cli
 * @return int
 */
public function getId_cli () {}

/**
 * Définir le/la id_cli
 * @param $id_cli int
 * @param $execute bool Exécuter la requête update ?
 * @return bool Opération réussie ?
 */
public function setId_cli ($id_cli,
                           $execute = true) {}

/**
 * Récupérer le/la visible_exp
 * @return int
 */
public function getVisible_exp () {}

/**
 * Définir le/la visible_exp

```

```

* @param $visible_exp int
* @param $execute bool Exécuter la requête update ?
* @return bool Opération réussie ?
*/
public function setVisible_exp ($visible_exp,
                               $execute = true) {}

/**
* ToString
* @return string Représentation de exploitation sous la forme d'un string
*/
public function __toString () {}

/**
* Sérialiser
* @param $serialize bool Activer la sérialisation ?
* @return string Sérialisation du/de la exploitation
*/
public function serialize ($serialize = true) {}

/**
* Désérialiser
* @param $pdo PDO
* @param $string string Sérialisation du/de la exploitation
* @param $lazyload bool Activer le chargement fainéant ?
* @return Exploitation
*/
public static function unserialize (PDO $pdo,
                                    $string,
                                    $lazyload = true) {}

}

```

```
+ www > model > classes > Exploitation.php
```

```
/**
 * @name Exploitation
 * @version 11/05/2016 (dd/mm/yyyy)
 */
class Exploitation extends Base_Exploitation {

    const TABLENAME = 'EXPLOITATION';

    const FIELDNAME_SIRET_EXP = 'siret_exp';

    const FIELDNAME_NOM_EXP = 'nom_exp';

    const FIELDNAME_ADRES1_EXP = 'adres1_exp';

    const FIELDNAME_ADRES2_EXP = 'adres2_exp';

    const FIELDNAME_CP_EXP = 'cp_exp';

    const FIELDNAME_VILLE_EXP = 'ville_exp';

    const FIELDNAME_ID_CLI = 'id_cli';

    const FIELDNAME_VISIBLE_EXP = 'visible_exp';

    public static function ExploitByClientID (PDO $pdo,
                                              $cli,
                                              $entrep) {}

    public static function supprDef ($exploit) {}

    /**
     * SIRET_EXP: Siret de l'exploitation
     * NOM_EXP: Nom de l'exploitation
     * ADRES1_EXP: Adresse 1 de l'exploitation
     * ADRES2_EXP: Adresse 2 de l'exploitation
     * CP_EXP: Code postal de l'exploitation
     * VILLE_EXP: Ville de l'exploitation
     * ID_CLI: Numéro du client propriétaire dans la base de données
     */
    public static function exploitAjout (PDO $pdo,
                                         $quote) {}

}
```

```
+ www > model > classes > Exposition.php
```

```
/**
 * @name Base_Exposition
 * @version 26/05/2016 (dd/mm/yyyy)
 * @author Axel Jalabert
 */
abstract class Base_Exposition {

    const TABLENAME = 'EXPOSITION';

    const FIELDNAME_ID_EXPO = 'id_expo';

    const FIELDNAME_ID_EMP_CHAUF = 'id_emp_chauf';

    const FIELDNAME_DATE_EXPO = 'date_expo';

    /** @var PDO */
    protected $_pdo;

    /** @var array tableau pour le chargement fainéant */
    protected static $_lazyload;

    /** @var int */
    protected $_id_expo;

    /** @var int */
    protected $_id_emp_chauf;

    /** @var string */
    protected $_date_expo;

    /**
     * Construire un(e) exposition
     * @param $pdo PDO
     * @param $id_expo int
     * @param $id_emp_chauf int
     * @param $date_expo string
     * @param $lazyload bool Activer le chargement fainéant ?
     */
    protected function __construct (PDO $pdo,
                                     $id_expo,
                                     $id_emp_chauf,
                                     $date_expo,
                                     $lazyload = true) {}

    /**
     * Créer un(e) exposition
     * @param $pdo PDO
     * @param $id_expo int
     * @param $id_emp_chauf int
     * @param $date_expo string
     * @param $lazyload bool Activer le chargement fainéant ?
     */
}
```

```

* @return Exposition
*/
public static function create (PDO $pdo,
                               $id_expo,
                               $id_emp_chauf,
                               $date_expo,
                               $lazyload = true) {}

/**
 * Compter les expositions
 * @param $pdo PDO
 * @return int Nombre de expositions
 */
public static function count (PDO $pdo) {}

/**
 * Requête de sélection
 * @param $pdo PDO
 * @param $where string|array
 * @param $orderby string|array
 * @param $limit string|array
 * @param $from string|array
 * @return PDOStatement
 */
protected static function _select (PDO $pdo,
                                    $where = null,
                                    $orderby = null,
                                    $limit = null,
                                    $from = null) {}

/**
 * Charger un(e) exposition
 * @param $pdo PDO
 * @param $id_expo int
 * @param $lazyload bool Activer le chargement fainéant ?
 * @return Exposition
 */
public static function load (PDO $pdo,
                             $id_expo,
                             $lazyload = true) {}

/**
 * Recharger les données depuis la base de données
 */
public function reload () {}

/**
 * Charger tous/toutes les expositions
 * @param $pdo PDO
 * @param $lazyload bool Activer le chargement fainéant ?
 * @return Exposition[] Tableau de expositions
 */
public static function loadAll (PDO $pdo,

```

```

        $lazyload = true) {}

/**
 * Sélectionner tous/toutes les expositions
 * @param $pdo PDO
 * @return PDOStatement
 */
public static function selectAll (PDO $pdo) {}

/**
 * Récupérer le/la exposition suivant(e) d'un jeu de résultats
 * @param $pdo PDO
 * @param $pdoStatement PDOStatement
 * @param $lazyload bool Activer le chargement fainéant ?
 * @return Exposition
 */
public static function fetch (PDO $pdo,
                             PDOStatement $pdoStatement,
                             $lazyload = true) {}

/**
 * Récupérer tous/toutes les expositions d'un jeu de résultats
 * @param $pdo PDO
 * @param $pdoStatement PDOStatement
 * @param $lazyload bool Activer le chargement fainéant ?
 * @return Exposition[] Tableau de expositions
 */
public static function fetchAll (PDO $pdo,
                                 PDOStatement $pdoStatement,
                                 $lazyload = true) {}

/**
 * Test d'égalité
 * @param $exposition Exposition
 * @return bool Les objets sont-ils égaux ?
 */
public function equals ($exposition) {}

/**
 * Vérifier que le/la exposition existe en base de données
 * @return bool Le/La exposition existe en base de données ?
 */
public function exists () {}

/**
 * Supprimer le/la exposition
 * @return bool Opération réussie ?
 */
public function delete () {}

/**
 * Mettre à jour un champ dans la base de données
 * @param $fields array

```

```

* @param $values array
* @return bool Opération réussie ?
*/
protected function _set ($fields,
                        $values) {}

/**
* Mettre à jour tous les champs dans la base de données
* @return bool Opération réussie ?
*/
public function update () {}

/**
* Récupérer le/la id_expo
* @return int
*/
public function getId_expo () {}

/**
* Récupérer le/la id_emp_chauf
* @return int
*/
public function getId_emp_chauf () {}

/**
* Définir le/la id_emp_chauf
* @param $id_emp_chauf int
* @param $execute bool Exécuter la requête update ?
* @return bool Opération réussie ?
*/
public function setId_emp_chauf ($id_emp_chauf,
                                $execute = true) {}

/**
* Récupérer le/la date_expo
* @return string
*/
public function getDate_expo () {}

/**
* Définir le/la date_expo
* @param $date_expo string
* @param $execute bool Exécuter la requête update ?
* @return bool Opération réussie ?
*/
public function setDate_expo ($date_expo,
                              $execute = true) {}

/**
* ToString
* @return string Représentation de exposition sous la forme d'un string
*/
public function __toString () {}

```

```
/**
 * S rialiser
 * @param $serialize bool Activer la s rialisation ?
 * @return string S rialisation du/de la exposition
 */
public function serialize ($serialize = true) {}

/**
 * D s rialiser
 * @param $pdo PDO
 * @param $string string S rialisation du/de la exposition
 * @param $lazyload bool Activer le chargement fain ant ?
 * @return Exposition
 */
public static function unserialize (PDO $pdo,
                                     $string,
                                     $lazyload = true) {}
}
```

```
+ www > model > classes > Exposition.php
```

```
/**  
 * @name Exposition  
 * @version 26/05/2016 (dd/mm/yyyy)  
 */  
class Exposition extends Base_Exposition {  
  
    const TABLENAME = 'EXPOSITION';  
  
    const FIELDNAME_ID_EXPO = 'id_expo';  
  
    const FIELDNAME_ID_EMP_CHAUF = 'id_emp_chauf';  
  
    const FIELDNAME_DATE_EXPO = 'date_expo';  
  
    public static function verification_exposition ($clp_phyto) {}  
  
}
```

```
+ www > model > classes > Fournisseur.php
```

```
class Fournisseur {  
  
    private $nom_fournisseur;  
  
    public function setNomFournisseur ($nom_fournisseur) {}  
  
    public function getNomFournisseur () {}  
  
}
```

```
+ www > model > classes > Ilot.php
```

```
class Ilot {  
  
    private $numero_ilot;  
  
    public function setNumeroIlot ($numero_ilot) {}  
  
    public function getNumeroIlot () {}  
  
    public static function supprDef ($ilot) {}  
  
    /**  
     * NUM_ILOT: Numéro ilot  
     * ID_EXP: Id dans la base de données de l'exploitation parente  
     * ADRES_ILOT: Adresse de l'ilot  
     * CP_ILOT: Code postal ilot  
     * VILLE_ILOT: Nom ville ilot  
     */  
    public static function ilotCreation (PDO $pdo,  
                                         $quote) {}  
  
}
```

```
+ www > model > classes > Img_Entrep.php
```

```
/**
 * @name Base_Img_Entrep
 * @version 31/05/2016 (dd/mm/yyyy)
 * @author Axel Jalabert
 */
abstract class Base_Img_Entrep {

    const TABLENAME = 'IMG_ENTREP';

    const FIELDNAME_ID_IMG = 'id_img';

    const FIELDNAME_ID_ENTREP = 'id_entrep';

    const FIELDNAME_URL_IMG = 'url_img';

    /** @var PDO */
    protected $_pdo;

    /** @var array tableau pour le chargement fainéant */
    protected static $_lazyload;

    /** @var int */
    protected $_id_img;

    /** @var int */
    protected $_id_entrep;

    /** @var string */
    protected $_url_img;

    /**
     * Construire un(e) img_entrep
     * @param $pdo PDO
     * @param $id_img int
     * @param $id_entrep int
     * @param $url_img string
     * @param $lazyload bool Activer le chargement fainéant ?
     */
    protected function __construct (PDO $pdo,
                                     $id_img,
                                     $id_entrep,
                                     $url_img,
                                     $lazyload = true) {}

    /**
     * Créer un(e) img_entrep
     * @param $pdo PDO
     * @param $id_img int
     * @param $id_entrep int
     * @param $url_img string
     * @param $lazyload bool Activer le chargement fainéant ?
     */
}
```

```

* @return Img_Entrep
*/
public static function create (PDO $pdo,
                               $id_img,
                               $id_entrep,
                               $url_img,
                               $lazyload = true) {}

/**
 * Compter les img_entreps
 * @param $pdo PDO
 * @return int Nombre de img_entreps
 */
public static function count (PDO $pdo) {}

/**
 * Requête de sélection
 * @param $pdo PDO
 * @param $where string|array
 * @param $orderby string|array
 * @param $limit string|array
 * @param $from string|array
 * @return PDOStatement
 */
protected static function _select (PDO $pdo,
                                    $where = null,
                                    $orderby = null,
                                    $limit = null,
                                    $from = null) {}

/**
 * Charger un(e) img_entrep
 * @param $pdo PDO
 * @param $id_img int
 * @param $lazyload bool Activer le chargement fainéant ?
 * @return Img_Entrep
 */
public static function load (PDO $pdo,
                             $id_img,
                             $lazyload = true) {}

/**
 * Recharger les données depuis la base de données
 */
public function reload () {}

/**
 * Charger tous/toutes les img_entreps
 * @param $pdo PDO
 * @param $lazyload bool Activer le chargement fainéant ?
 * @return Img_Entrep[] Tableau de img_entreps
 */
public static function loadAll (PDO $pdo,

```

```

        $lazyload = true) {}

/**
 * Sélectionner tous/toutes les img_entreps
 * @param $pdo PDO
 * @return PDOStatement
 */
public static function selectAll (PDO $pdo) {}

/**
 * Récupérer le/la img_entrep suivant(e) d'un jeu de résultats
 * @param $pdo PDO
 * @param $pdoStatement PDOStatement
 * @param $lazyload bool Activer le chargement fainéant ?
 * @return Img_Entrep
 */
public static function fetch (PDO $pdo,
                             PDOStatement $pdoStatement,
                             $lazyload = true) {}

/**
 * Récupérer tous/toutes les img_entreps d'un jeu de résultats
 * @param $pdo PDO
 * @param $pdoStatement PDOStatement
 * @param $lazyload bool Activer le chargement fainéant ?
 * @return Img_Entrep[] Tableau de img_entreps
 */
public static function fetchAll (PDO $pdo,
                                 PDOStatement $pdoStatement,
                                 $lazyload = true) {}

/**
 * Test d'égalité
 * @param $img_entrep Img_Entrep
 * @return bool Les objets sont-ils égaux ?
 */
public function equals ($img_entrep) {}

/**
 * Vérifier que le/la img_entrep existe en base de données
 * @return bool Le/La img_entrep existe en base de données ?
 */
public function exists () {}

/**
 * Supprimer le/la img_entrep
 * @return bool Opération réussie ?
 */
public function delete () {}

/**
 * Mettre à jour un champ dans la base de données
 * @param $fields array

```

```

* @param $values array
* @return bool Opération réussie ?
*/
protected function _set ($fields,
                        $values) {}

/**
* Mettre à jour tous les champs dans la base de données
* @return bool Opération réussie ?
*/
public function update () {}

/**
* Récupérer le/la id_img
* @return int
*/
public function getId_img () {}

/**
* Récupérer le/la id_entrep
* @return int
*/
public function getId_entrep () {}

/**
* Définir le/la id_entrep
* @param $id_entrep int
* @param $execute bool Exécuter la requête update ?
* @return bool Opération réussie ?
*/
public function setId_entrep ($id_entrep,
                             $execute = true) {}

/**
* Récupérer le/la url_img
* @return string
*/
public function getUrl_img () {}

/**
* Définir le/la url_img
* @param $url_img string
* @param $execute bool Exécuter la requête update ?
* @return bool Opération réussie ?
*/
public function setUrl_img ($url_img,
                           $execute = true) {}

/**
* ToString
* @return string Représentation de img_entrep sous la forme d'un string
*/
public function __toString () {}

```

```
/**
 * S rialiser
 * @param $serialize bool Activer la s rialisation ?
 * @return string S rialisation du/de la img_entrep
 */
public function serialize ($serialize = true) {}

/**
 * D s rialiser
 * @param $pdo PDO
 * @param $string string S rialisation du/de la img_entrep
 * @param $lazyload bool Activer le chargement fain ant ?
 * @return Img_Entrep
 */
public static function unserialize (PDO $pdo,
                                     $string,
                                     $lazyload = true) {}
}
```

```
+ www > model > classes > InfoAD.php
```

```
class InfoAD {  
  
    private $id_infoAD;  
  
    private $libelle_infoAD;  
  
    public function setIdInfoAD ($id_infoAD) {}  
  
    public function getIdInfoAD () {}  
  
    public function setLibelleInfoAD ($libelle_infoAD) {}  
  
    public function getLibelleInfoAD () {}  
  
}
```

```
+ www > model > classes > Intervention.php
```

```
/**
 * @name Base_Intervention
 * @version 19/05/2016 (dd/mm/yyyy)
 * @author Axel Jalabert
 */
abstract class Base_Intervention {

    const TABLENAME = 'INTERVENTION';

    const FIELDNAME_ID_I = 'id_i';

    const FIELDNAME_ID_CHANT = 'id_chant';

    const FIELDNAME_CIBLE_I = 'cible_i';

    const FIELDNAME_TYPE_I = 'type_i';

    const FIELDNAME_CATEGORIE_I = 'categorie_i';

    const FIELDNAME_PARTIC_I = 'partic_i';

    /** @var PDO */
    protected $_pdo;

    /** @var array tableau pour le chargement fainéant */
    protected static $_lazyload;

    /** @var int */
    protected $_id_i;

    /** @var int */
    protected $_id_chant;

    /** @var string */
    protected $_cible_i;

    /** @var string */
    protected $_type_i;

    /** @var string */
    protected $_categorie_i;

    /** @var string */
    protected $_partic_i;

    /**
     * Construire un(e) intervention
     * @param $pdo PDO
     * @param $id_i int
     * @param $id_chant int
     * @param $cible_i string
     */
}
```

```

* @param $type_i string
* @param $categorie_i string
* @param $partic_i string
* @param $lazyload bool Activer le chargement fainéant ?
*/
protected function __construct (PDO $pdo,
                                $id_i,
                                $id_chant,
                                $cible_i,
                                $type_i,
                                $categorie_i,
                                $partic_i,
                                $lazyload = true) {}

/**
* Créer un(e) intervention
* @param $pdo PDO
* @param $id_i int
* @param $id_chant int
* @param $cible_i string
* @param $type_i string
* @param $categorie_i string
* @param $partic_i string
* @param $lazyload bool Activer le chargement fainéant ?
* @return Intervention
*/
public static function create (PDO $pdo,
                              $id_i,
                              $id_chant,
                              $cible_i,
                              $type_i,
                              $categorie_i,
                              $partic_i,
                              $lazyload = true) {}

/**
* Compter les interventions
* @param $pdo PDO
* @return int Nombre de interventions
*/
public static function count (PDO $pdo) {}

/**
* Requête de sélection
* @param $pdo PDO
* @param $where string|array
* @param $orderby string|array
* @param $limit string|array
* @param $from string|array
* @return PDOStatement
*/
protected static function _select (PDO $pdo,
                                   $where = null,

```

```

        $orderby = null,
        $limit = null,
        $from = null) {}

/**
 * Charger un(e) intervention
 * @param $pdo PDO
 * @param $id_i int
 * @param $lazyload bool Activer le chargement fainéant ?
 * @return Intervention
 */
public static function load (PDO $pdo,
                            $id_i,
                            $lazyload = true) {}

/**
 * Recharger les données depuis la base de données
 */
public function reload () {}

/**
 * Charger tous/toutes les interventions
 * @param $pdo PDO
 * @param $lazyload bool Activer le chargement fainéant ?
 * @return Intervention[] Tableau de interventions
 */
public static function loadAll (PDO $pdo,
                               $lazyload = true) {}

/**
 * Sélectionner tous/toutes les interventions
 * @param $pdo PDO
 * @return PDOStatement
 */
public static function selectAll (PDO $pdo) {}

/**
 * Récupérer le/la intervention suivant(e) d'un jeu de résultats
 * @param $pdo PDO
 * @param $pdoStatement PDOStatement
 * @param $lazyload bool Activer le chargement fainéant ?
 * @return Intervention
 */
public static function fetch (PDO $pdo,
                             PDOStatement $pdoStatement,
                             $lazyload = true) {}

/**
 * Récupérer tous/toutes les interventions d'un jeu de résultats
 * @param $pdo PDO
 * @param $pdoStatement PDOStatement
 * @param $lazyload bool Activer le chargement fainéant ?
 * @return Intervention[] Tableau de interventions

```

```

*/
public static function fetchAll (PDO $pdo,
                                PDOStatement $pdoStatement,
                                $lazyload = true) {}

/**
 * Test d'égalité
 * @param $intervention Intervention
 * @return bool Les objets sont-ils égaux ?
 */
public function equals ($intervention) {}

/**
 * Vérifier que le/la intervention existe en base de données
 * @return bool Le/La intervention existe en base de données ?
 */
public function exists () {}

/**
 * Supprimer le/la intervention
 * @return bool Opération réussie ?
 */
public function delete () {}

/**
 * Mettre à jour un champ dans la base de données
 * @param $fields array
 * @param $values array
 * @return bool Opération réussie ?
 */
protected function _set ($fields,
                          $values) {}

/**
 * Mettre à jour tous les champs dans la base de données
 * @return bool Opération réussie ?
 */
public function update () {}

/**
 * Récupérer le/la id_i
 * @return int
 */
public function getId_i () {}

/**
 * Récupérer le/la id_chant
 * @return int
 */
public function getId_chant () {}

/**
 * Définir le/la id_chant

```

```

* @param $id_chant int
* @param $execute bool Exécuter la requête update ?
* @return bool Opération réussie ?
*/
public function setId_chant ($id_chant,
                             $execute = true) {}

/**
* Récupérer le/la cible_i
* @return string
*/
public function getCible_i () {}

/**
* Définir le/la cible_i
* @param $cible_i string
* @param $execute bool Exécuter la requête update ?
* @return bool Opération réussie ?
*/
public function setCible_i ($cible_i,
                             $execute = true) {}

/**
* Récupérer le/la type_i
* @return string
*/
public function getType_i () {}

/**
* Définir le/la type_i
* @param $type_i string
* @param $execute bool Exécuter la requête update ?
* @return bool Opération réussie ?
*/
public function setType_i ($type_i,
                             $execute = true) {}

/**
* Récupérer le/la categorie_i
* @return string
*/
public function getCategorie_i () {}

/**
* Définir le/la categorie_i
* @param $categorie_i string
* @param $execute bool Exécuter la requête update ?
* @return bool Opération réussie ?
*/
public function setCategorie_i ($categorie_i,
                                 $execute = true) {}

/**

```

```

* Récupérer le/la partic_i
* @return string
*/
public function getPartic_i () {}

/**
* Définir le/la partic_i
* @param $partic_i string
* @param $execute bool Exécuter la requête update ?
* @return bool Opération réussie ?
*/
public function setPartic_i ($partic_i,
                            $execute = true) {}

/**
* ToString
* @return string Représentation de intervention sous la forme d'un string
*/
public function __toString () {}

/**
* Sérialiser
* @param $serialize bool Activer la sérialisation ?
* @return string Sérialisation du/de la intervention
*/
public function serialize ($serialize = true) {}

/**
* Désérialiser
* @param $pdo PDO
* @param $string string Sérialisation du/de la intervention
* @param $lazyload bool Activer le chargement fainéant ?
* @return Intervention
*/
public static function unserialize (PDO $pdo,
                                    $string,
                                    $lazyload = true) {}

}

```

```
+ www > model > classes > Intervention.php
```

```
/**
 * @name Intervention
 * @version 19/05/2016 (dd/mm/yyyy)
 */
class Intervention extends Base_Intervention {

    const TABLENAME = 'INTERVENTION';

    const FIELDNAME_ID_I = 'id_i';

    const FIELDNAME_ID_CHANT = 'id_chant';

    const FIELDNAME_CIBLE_I = 'cible_i';

    const FIELDNAME_TYPE_I = 'type_i';

    const FIELDNAME_CATEGORIE_I = 'categorie_i';

    const FIELDNAME_PARTIC_I = 'partic_i';

    public static function infoIntervention (PDO $pdo,
                                             $fiche) {}

}
```

```
+ www > model > classes > Materiel.php
```

```
/**
 * @name Base_Materiel
 * @version 18/05/2016 (dd/mm/yyyy)
 * @author Axel Jalabert
 */
abstract class Base_Materiel {

    const TABLENAME = 'MATERIEL';

    const FIELDNAME_ID_MAT = 'id_mat';

    const FIELDNAME_LIB_MAT = 'lib_mat';

    const FIELDNAME_SERIE_MAT = 'serie_mat';

    const FIELDNAME_MARQUE_MAT = 'marque_mat';

    const FIELDNAME_MODELE_MAT = 'modele_mat';

    const FIELDNAME_CUVE_MAT = 'cuve_mat';

    const FIELDNAME_ACHAT_MAT = 'achat_mat';

    const FIELDNAME_ID_ENTREP = 'id_entrep';

    const FIELDNAME_VISIBLE_MAT = 'visible_mat';

    /** @var PDO */
    protected $_pdo;

    /** @var array tableau pour le chargement fainéant */
    protected static $_lazyload;

    /** @var int */
    protected $_id_mat;

    /** @var string */
    protected $_lib_mat;

    /** @var string */
    protected $_serie_mat;

    /** @var string */
    protected $_marque_mat;

    /** @var string */
    protected $_modele_mat;

    /** @var string */
    protected $_cuve_mat;
}
```

```

/** @var string */
protected $_achat_mat;

/** @var string */
protected $_id_entrep;

/** @var int */
protected $_visible_mat;

/**
 * Construire un(e) materiel
 * @param $pdo PDO
 * @param $id_mat int
 * @param $lib_mat string
 * @param $serie_mat string
 * @param $marque_mat string
 * @param $modele_mat string
 * @param $cuve_mat string
 * @param $achat_mat string
 * @param $id_entrep string
 * @param $visible_mat int
 * @param $lazyload bool Activer le chargement fainéant ?
 */
protected function __construct (PDO $pdo,
                                $id_mat,
                                $lib_mat,
                                $serie_mat,
                                $marque_mat,
                                $modele_mat,
                                $cuve_mat,
                                $achat_mat,
                                $id_entrep,
                                $visible_mat,
                                $lazyload = true) {}

/**
 * Créer un(e) materiel
 * @param $pdo PDO
 * @param $id_mat int
 * @param $lib_mat string
 * @param $serie_mat string
 * @param $marque_mat string
 * @param $modele_mat string
 * @param $cuve_mat string
 * @param $achat_mat string
 * @param $id_entrep string
 * @param $visible_mat int
 * @param $lazyload bool Activer le chargement fainéant ?
 * @return Materiel
 */
public static function create (PDO $pdo,
                              $id_mat,
                              $lib_mat,

```

```

        $serie_mat,
        $marque_mat,
        $modele_mat,
        $cuve_mat,
        $sachat_mat,
        $id_entrep,
        $visible_mat,
        $lazyload = true) {}

/**
 * Compter les materiels
 * @param $pdo PDO
 * @return int Nombre de materiels
 */
public static function count (PDO $pdo) {}

/**
 * Requête de sélection
 * @param $pdo PDO
 * @param $where string|array
 * @param $orderby string|array
 * @param $limit string|array
 * @param $from string|array
 * @return PDOStatement
 */
protected static function _select (PDO $pdo,
                                    $where = null,
                                    $orderby = null,
                                    $limit = null,
                                    $from = null) {}

/**
 * Charger un(e) materiel
 * @param $pdo PDO
 * @param $id_mat int
 * @param $lazyload bool Activer le chargement fainéant ?
 * @return Materiel
 */
public static function load (PDO $pdo,
                            $id_mat,
                            $lazyload = true) {}

/**
 * Recharger les données depuis la base de données
 */
public function reload () {}

/**
 * Charger tous/toutes les materiels
 * @param $pdo PDO
 * @param $lazyload bool Activer le chargement fainéant ?
 * @return Materiel[] Tableau de materiels
 */

```

```

public static function loadAll (PDO $pdo,
                               $lazyload = true) {}

/**
 * Sélectionner tous/toutes les materiels
 * @param $pdo PDO
 * @return PDOStatement
 */
public static function selectAll (PDO $pdo) {}

/**
 * Récupérer le/la materiel suivant(e) d'un jeu de résultats
 * @param $pdo PDO
 * @param $pdoStatement PDOStatement
 * @param $lazyload bool Activer le chargement fainéant ?
 * @return Materiel
 */
public static function fetch (PDO $pdo,
                              PDOStatement $pdoStatement,
                              $lazyload = true) {}

/**
 * Récupérer tous/toutes les materiels d'un jeu de résultats
 * @param $pdo PDO
 * @param $pdoStatement PDOStatement
 * @param $lazyload bool Activer le chargement fainéant ?
 * @return Materiel[] Tableau de materiels
 */
public static function fetchAll (PDO $pdo,
                                 PDOStatement $pdoStatement,
                                 $lazyload = true) {}

/**
 * Test d'égalité
 * @param $materiel Materiel
 * @return bool Les objets sont-ils égaux ?
 */
public function equals ($materiel) {}

/**
 * Vérifier que le/la materiel existe en base de données
 * @return bool Le/La materiel existe en base de données ?
 */
public function exists () {}

/**
 * Supprimer le/la materiel
 * @return bool Opération réussie ?
 */
public function delete () {}

/**
 * Mettre à jour un champ dans la base de données

```

```

* @param $fields array
* @param $values array
* @return bool Opération réussie ?
*/
protected function _set ($fields,
                        $values) {}

/**
 * Mettre à jour tous les champs dans la base de données
 * @return bool Opération réussie ?
 */
public function update () {}

/**
 * Récupérer le/la id_mat
 * @return int
 */
public function getId_mat () {}

/**
 * Récupérer le/la lib_mat
 * @return string
 */
public function getLib_mat () {}

/**
 * Définir le/la lib_mat
 * @param $lib_mat string
 * @param $execute bool Exécuter la requête update ?
 * @return bool Opération réussie ?
 */
public function setLib_mat ($lib_mat,
                          $execute = true) {}

/**
 * Récupérer le/la serie_mat
 * @return string
 */
public function getSerie_mat () {}

/**
 * Définir le/la serie_mat
 * @param $serie_mat string
 * @param $execute bool Exécuter la requête update ?
 * @return bool Opération réussie ?
 */
public function setSerie_mat ($serie_mat,
                             $execute = true) {}

/**
 * Récupérer le/la marque_mat
 * @return string
 */

```

```

public function getMarque_mat () {}

/**
 * Définir le/la marque_mat
 * @param $marque_mat string
 * @param $execute bool Exécuter la requête update ?
 * @return bool Opération réussie ?
 */
public function setMarque_mat ($marque_mat,
                              $execute = true) {}

/**
 * Récupérer le/la modele_mat
 * @return string
 */
public function getModele_mat () {}

/**
 * Définir le/la modele_mat
 * @param $modele_mat string
 * @param $execute bool Exécuter la requête update ?
 * @return bool Opération réussie ?
 */
public function setModele_mat ($modele_mat,
                              $execute = true) {}

/**
 * Récupérer le/la cuve_mat
 * @return string
 */
public function getCuve_mat () {}

/**
 * Définir le/la cuve_mat
 * @param $cuve_mat string
 * @param $execute bool Exécuter la requête update ?
 * @return bool Opération réussie ?
 */
public function setCuve_mat ($cuve_mat,
                              $execute = true) {}

/**
 * Récupérer le/la achat_mat
 * @return string
 */
public function getAchat_mat () {}

/**
 * Définir le/la achat_mat
 * @param $achat_mat string
 * @param $execute bool Exécuter la requête update ?
 * @return bool Opération réussie ?
 */

```

```

public function setAchat_mat ($achat_mat,
                             $execute = true) {}

/**
 * Récupérer le/la id_entrep
 * @return string
 */
public function getId_entrep () {}

/**
 * Définir le/la id_entrep
 * @param $id_entrep string
 * @param $execute bool Exécuter la requête update ?
 * @return bool Opération réussie ?
 */
public function setId_entrep ($id_entrep,
                              $execute = true) {}

/**
 * Récupérer le/la visible_mat
 * @return int
 */
public function getVisible_mat () {}

/**
 * Définir le/la visible_mat
 * @param $visible_mat int
 * @param $execute bool Exécuter la requête update ?
 * @return bool Opération réussie ?
 */
public function setVisible_mat ($visible_mat,
                                $execute = true) {}

/**
 * ToString
 * @return string Représentation de materiel sous la forme d'un string
 */
public function __toString () {}

/**
 * Sérialiser
 * @param $serialize bool Activer la sérialisation ?
 * @return string Sérialisation du/de la materiel
 */
public function serialize ($serialize = true) {}

/**
 * Désérialiser
 * @param $pdo PDO
 * @param $string string Sérialisation du/de la materiel
 * @param $lazyload bool Activer le chargement fainéant ?
 * @return Materiel
 */

```

```
public static function unserialize (PDO $pdo,  
    $string,  
    $lazyload = true) {}  
  
}
```

```
+ www > model > classes > Materiel.php
```

```
/**
 * @name Materiel
 * @version 18/05/2016 (dd/mm/yyyy)
 */
class Materiel extends Base_Materiel {

    const TABLENAME = 'MATERIEL';

    const FIELDNAME_ID_MAT = 'id_mat';

    const FIELDNAME_LIB_MAT = 'lib_mat';

    const FIELDNAME_SERIE_MAT = 'serie_mat';

    const FIELDNAME_MARQUE_MAT = 'marque_mat';

    const FIELDNAME_MODELE_MAT = 'modele_mat';

    const FIELDNAME_CUVE_MAT = 'cuve_mat';

    const FIELDNAME_ACHAT_MAT = 'achat_mat';

    const FIELDNAME_ID_ENTREP = 'id_entrep';

    const FIELDNAME_VISIBLE_MAT = 'visible_mat';

    public static function materielList (PDO $pdo,
                                         $entrep) {}

    public static function supprDef ($materiel) {}

    /**
     * LIB_MAT: Nom du matériel
     * SERIE_MAT: Numéro de série
     * MARQUE_MAT: Marque du matériel
     * MODELE_MAT: Modèle du matériel
     * CUVE_MAT: Capacité de la cuve du matériel
     * ACHAT_MAT: Date d'achat du matériel
     */
    public static function materielAjout (PDO $pdo,
                                         $entrep,
                                         $quote) {}

}
```

```
+ www > model > classes > Notification.php
```

```
class Notification {  
  
    private $descr_court_notification;  
  
    private $descr_long_notification;  
  
    public function setDescrCourtNotification ($descr_court_notification) {}  
  
    public function getDescrCourtNotification () {}  
  
    public function setDescrLongNotification ($descr_long_notification) {}  
  
    public function getDescrLongNotification () {}  
  
}
```

```
+ www > model > classes > NumCli.php
```

```
class NumCli {  
  
    private $num_cli;  
  
    private $num_contrat_cli;  
  
    public function setNumCli ($num_cli) {}  
  
    public function getNumCli () {}  
  
    public function setNumContratCli ($num_contrat_cli) {}  
  
    public function getNumContratCli () {}  
  
}
```

```
+ www > model > classes > Organigramme.php
```

```
/**  
 * Permet de générer le PDF de l'Organigramme  
 * Avec seulement le header et le footer type.  
 * Les images sont insérées dans la classe Organigramme.  
 */  
class OrganigrammeFPDF extends FPDF {  
  
    public function Header () {}  
  
    public function Footer () {}  
  
}
```

```
+ www > model > classes > Organigramme.php
```

```
/**
 * Permet d'utiliser la librairie GD.
 *
 * @date 2018-04-27 (yyyy/mm/dd)
 * @author BOURA Tristan
 */
class GDImage {

    /**
     * type de l'IMAGE
     * @var int
     */
    private $type;

    /**
     * Nom de l'IMAGE
     * @var string
     */
    private $name;

    /**
     * l'IMAGE
     * @var resource
     */
    private $image;

    /**
     * Largeur de l'IMAGE
     * @var integer
     */
    private $width;

    /**
     * Hauteur de l'IMAGE
     * @var integer
     */
    private $height;

    /**
     * Couleur de remplissge
     * @var integer
     */
    private $fillcolor;

    /**
     * Couleur de bordure
     * @var integer
     */
    private $filloutline;

    /**
```

```

* Chemin de la font
* @var string
*/
private $font;

/**
* Taille de la font
* @var integer
*/
private $size;

/**
* Chemin absolue de l'IMAGE sauvegardée
* @var string
*/
private $root;

/**
* Chemin d'intégration HTML de l'IMAGE sauvegardée
* @var string
*/
private $webroot;

public function getWidth () {}

public function getHeight () {}

public function getRoot () {}

public function getWebroot () {}

public function getImage () {}

public function getName () {}

public function getType () {}

/**
* Constructeur de la classe qui prend en paramètre
* le nom et la dimension de l'image.
* Puis initialise les paramètre de la classe par défaut.
*
* @param string $name Nom de l'Image
* @param integer $width Largeur de l'Image en pixel
* @param integer $height Hauteur de l'Image en pixel
*/
public function __construct ($name,
                            $width,
                            $height,
                            $type) {}

/**
* Sauvegarde l'IMAGE au format jpg || png au choix, et initialise

```

```

* les variables $webroot et $root.
*
* @param string $dir Dossier d'export dans le dossier "view/img"
* @param string $type L'extension de l'IMAGE : jpg || png
* @return boolean Retourne true si l'exécution est réussite, sinon false
*/
public function save ($dir = false,
                    $type = 'jpg') {}

/**
* Enregistre la couleur de remplissage.
* @param integer $r Red
* @param integer $g Green
* @param integer $b Blue
*/
public function fill ($r = false,
                    $g = false,
                    $b = false) {}

public function black () {}

public function white () {}

/**
* Enregistre la couleur de bordure.
* @param integer $r Red
* @param integer $g Green
* @param integer $b Blue
*/
public function outline ($r = false,
                       $g = false,
                       $b = false) {}

/**
* Initialise la largeur de la bordure.
* @param integer $w Largeur de la bordure
*/
public function stroke ($w) {}

/**
* Dessine un rectangle sur l'IMAGE.
* @param integer $x Position x du rectangle
* @param integer $y Position y du rectangle
* @param integer $w Largeur du rectangle
* @param integer $h Hauteur du rectangle
*/
public function rectangle ($x,
                          $y,
                          $w,
                          $h) {}

/**
* Dessine une ligne sur l'IMAGE.

```

```

* @param integer $x Position x
* @param integer $y Position y
* @param integer $w Position relative à partir de $x
* @param integer $h Position relative à partir de $y
*/
public function line ($x,
                    $y,
                    $w,
                    $h) {}

/**
 * Enregistre la font envoyée en paramètre
 * @param string $dir Chemin de la font
 */
public function font ($dir) {}

/**
 * Enregistre la taille de la font
 * @param integer $size Taille de la font
 */
public function size ($size) {}

/**
 * Retourne les dimensions du texte envoyée en paramètre
 * dans une liste ["left": px, "top": px, "width": px, "height": px].
 * @param string $t Texte
 * @param boolean $v Texte horizontal si True sinon texte vertical
 * @return array Dimensions du texte
 */
public function box ($t,
                  $v = false) {}

/**
 * Ecrit sur l'image
 * @param integer $x Position x
 * @param integer $y Position y
 * @param string $t Texte
 * @param boolean $v Text horizontal si True sinon vertical
 */
public function write ($x,
                    $y,
                    $t,
                    $v = false) {}

/**
 * Dessine un polygone
 * @param array $p Liste des points
 */
public function polygon ($p) {}

```

```

}
```

```
+ www > model > classes > Pdf_Chantier.php
```

```
/**
 * @name Base_Pdf_Chantier
 * @version 27/05/2016 (dd/mm/yyyy)
 * @author Axel Chantier
 */
abstract class Base_Pdf_Chantier {

    const TABLENAME = 'PDF_CHANTIER';

    const FIELDNAME_ID_PDF = 'id_pdf';

    const FIELDNAME_ID_CHANT = 'id_chant';

    const FIELDNAME_URL_PDF = 'url_pdf';

    /** @var PDO */
    protected $_pdo;

    /** @var array tableau pour le chargement fainéant */
    protected static $_lazyload;

    /** @var int */
    protected $_id_pdf;

    /** @var int */
    protected $_id_chant;

    /** @var string */
    protected $_url_pdf;

    /**
     * Construire un(e) pdf_chantier
     * @param $pdo PDO
     * @param $id_pdf int
     * @param $id_chant int
     * @param $url_pdf string
     * @param $lazyload bool Activer le chargement fainéant ?
     */
    protected function __construct (PDO $pdo,
                                     $id_pdf,
                                     $id_chant,
                                     $url_pdf,
                                     $lazyload = true) {}

    /**
     * Créer un(e) pdf_chantier
     * @param $pdo PDO
     * @param $id_pdf int
     * @param $id_chant int
     * @param $url_pdf string
     * @param $lazyload bool Activer le chargement fainéant ?
     */
}
```

```

* @return Pdf_Chantier
*/
public static function create (PDO $pdo,
                               $id_pdf,
                               $id_chant,
                               $url_pdf,
                               $lazyload = true) {}

/**
 * Compter les pdf_chantiers
 * @param $pdo PDO
 * @return int Nombre de pdf_chantiers
 */
public static function count (PDO $pdo) {}

/**
 * Requête de sélection
 * @param $pdo PDO
 * @param $where string|array
 * @param $orderby string|array
 * @param $limit string|array
 * @param $from string|array
 * @return PDOStatement
 */
protected static function _select (PDO $pdo,
                                    $where = null,
                                    $orderby = null,
                                    $limit = null,
                                    $from = null) {}

/**
 * Charger un(e) pdf_chantier
 * @param $pdo PDO
 * @param $id_pdf int
 * @param $lazyload bool Activer le chargement fainéant ?
 * @return Pdf_Chantier
 */
public static function load (PDO $pdo,
                             $id_pdf,
                             $lazyload = true) {}

/**
 * Recharger les données depuis la base de données
 */
public function reload () {}

/**
 * Charger tous/toutes les pdf_chantiers
 * @param $pdo PDO
 * @param $lazyload bool Activer le chargement fainéant ?
 * @return Pdf_Chantier[] Tableau de pdf_chantiers
 */
public static function loadAll (PDO $pdo,

```

```

        $lazyload = true) {}

/**
 * Sélectionner tous/toutes les pdf_chantiers
 * @param $pdo PDO
 * @return PDOStatement
 */
public static function selectAll (PDO $pdo) {}

/**
 * Récupérer le/la pdf_chantier suivant(e) d'un jeu de résultats
 * @param $pdo PDO
 * @param $pdoStatement PDOStatement
 * @param $lazyload bool Activer le chargement fainéant ?
 * @return Pdf_Chantier
 */
public static function fetch (PDO $pdo,
                             PDOStatement $pdoStatement,
                             $lazyload = true) {}

/**
 * Récupérer tous/toutes les pdf_chantiers d'un jeu de résultats
 * @param $pdo PDO
 * @param $pdoStatement PDOStatement
 * @param $lazyload bool Activer le chargement fainéant ?
 * @return Pdf_Chantier[] Tableau de pdf_chantiers
 */
public static function fetchAll (PDO $pdo,
                                 PDOStatement $pdoStatement,
                                 $lazyload = true) {}

/**
 * Test d'égalité
 * @param $pdf_chantier Pdf_Chantier
 * @return bool Les objets sont-ils égaux ?
 */
public function equals ($pdf_chantier) {}

/**
 * Vérifier que le/la pdf_chantier existe en base de données
 * @return bool Le/La pdf_chantier existe en base de données ?
 */
public function exists () {}

/**
 * Supprimer le/la pdf_chantier
 * @return bool Opération réussie ?
 */
public function delete () {}

/**
 * Mettre à jour un champ dans la base de données
 * @param $fields array

```

```

* @param $values array
* @return bool Opération réussie ?
*/
protected function _set ($fields,
                        $values) {}

/**
* Mettre à jour tous les champs dans la base de données
* @return bool Opération réussie ?
*/
public function update () {}

/**
* Récupérer le/la id_pdf
* @return int
*/
public function getId_pdf () {}

/**
* Récupérer le/la id_chant
* @return int
*/
public function getId_chant () {}

/**
* Définir le/la id_chant
* @param $id_chant int
* @param $execute bool Exécuter la requête update ?
* @return bool Opération réussie ?
*/
public function setId_chant ($id_chant,
                            $execute = true) {}

/**
* Récupérer le/la url_pdf
* @return string
*/
public function getUrl_pdf () {}

/**
* Définir le/la url_pdf
* @param $url_pdf string
* @param $execute bool Exécuter la requête update ?
* @return bool Opération réussie ?
*/
public function setUrl_pdf ($url_pdf,
                           $execute = true) {}

/**
* ToString
* @return string Représentation de pdf_chantier sous la forme d'un string
*/
public function __toString () {}

```

```
/**
 * S rialiser
 * @param $serialize bool Activer la s rialisation ?
 * @return string S rialisation du/de la pdf_chantier
 */
public function serialize ($serialize = true) {}

/**
 * D s rialiser
 * @param $pdo PDO
 * @param $string string S rialisation du/de la pdf_chantier
 * @param $lazyload bool Activer le chargement fain ant ?
 * @return Pdf_Chantier
 */
public static function unserialize (PDO $pdo,
                                     $string,
                                     $lazyload = true) {}
}
```

```
+ www > model > classes > Pdf_Chantier.php
```

```
/**  
 * @name Pdf_Chantier  
 * @version 27/05/2016 (dd/mm/yyyy)  
 */  
class Pdf_Chantier extends Base_Pdf_Chantier {  
  
    const TABLENAME = 'PDF_CHANTIER';  
  
    const FIELDNAME_ID_PDF = 'id_pdf';  
  
    const FIELDNAME_ID_CHANT = 'id_chant';  
  
    const FIELDNAME_URL_PDF = 'url_pdf';  
  
    public static function infoPDF (PDO $pdo,  
                                     $fiche) {}  
  
}
```

```
+ www > model > classes > Periode.php
```

```
/**
 * @name Base_Periode
 * @version 03/05/2016 (dd/mm/yyyy)
 * @author Axel Jalabert
 */
abstract class Base_Periode {

    const TABLENAME = 'PERIODE';

    const FIELDNAME_ID_PERIODE = 'id_periode';

    const FIELDNAME_DEBUT_PERIODE = 'debut_periode';

    const FIELDNAME_FIN_PERIODE = 'fin_periode';

    const FIELDNAME_ID_TPER = 'id_tper';

    /** @var PDO */
    protected $_pdo;

    /** @var array tableau pour le chargement fainéant */
    protected static $_lazyload;

    /** @var int */
    protected $_id_periode;

    /** @var string */
    protected $_debut_periode;

    /** @var string */
    protected $_fin_periode;

    /** @var int */
    protected $_id_tper;

    /**
     * Construire un(e) periode
     * @param $pdo PDO
     * @param $id_periode int
     * @param $debut_periode string
     * @param $fin_periode string
     * @param $id_tper int
     * @param $lazyload bool Activer le chargement fainéant ?
     */
    protected function __construct (PDO $pdo,
                                     $id_periode,
                                     $debut_periode,
                                     $fin_periode,
                                     $id_tper,
                                     $lazyload = true) {}
}
```

```

/**
 * Créer un(e) periode
 * @param $pdo PDO
 * @param $id_periode int
 * @param $debut_periode string
 * @param $fin_periode string
 * @param $id_tper int
 * @param $lazyload bool Activer le chargement fainéant ?
 * @return Periode
 */
public static function create (PDO $pdo,
                               $id_periode,
                               $debut_periode,
                               $fin_periode,
                               $id_tper,
                               $lazyload = true) {}

/**
 * Compter les periodes
 * @param $pdo PDO
 * @return int Nombre de periodes
 */
public static function count (PDO $pdo) {}

/**
 * Requête de sélection
 * @param $pdo PDO
 * @param $where string|array
 * @param $orderby string|array
 * @param $limit string|array
 * @param $from string|array
 * @return PDOStatement
 */
protected static function _select (PDO $pdo,
                                    $where = null,
                                    $orderby = null,
                                    $limit = null,
                                    $from = null) {}

/**
 * Charger un(e) periode
 * @param $pdo PDO
 * @param $id_periode int
 * @param $lazyload bool Activer le chargement fainéant ?
 * @return Periode
 */
public static function load (PDO $pdo,
                             $id_periode,
                             $lazyload = true) {}

/**
 * Recharger les données depuis la base de données
 */

```

```

public function reload () {}

/**
 * Charger tous/toutes les periodes
 * @param $pdo PDO
 * @param $lazyload bool Activer le chargement fainéant ?
 * @return Periode[] Tableau de periodes
 */
public static function loadAll (PDO $pdo,
                                $lazyload = true) {}

/**
 * Sélectionner tous/toutes les periodes
 * @param $pdo PDO
 * @return PDOStatement
 */
public static function selectAll (PDO $pdo) {}

/**
 * Récupérer le/la periode suivant(e) d'un jeu de résultats
 * @param $pdo PDO
 * @param $pdoStatement PDOStatement
 * @param $lazyload bool Activer le chargement fainéant ?
 * @return Periode
 */
public static function fetch (PDO $pdo,
                              PDOStatement $pdoStatement,
                              $lazyload = true) {}

/**
 * Récupérer tous/toutes les periodes d'un jeu de résultats
 * @param $pdo PDO
 * @param $pdoStatement PDOStatement
 * @param $lazyload bool Activer le chargement fainéant ?
 * @return Periode[] Tableau de periodes
 */
public static function fetchAll (PDO $pdo,
                                 PDOStatement $pdoStatement,
                                 $lazyload = true) {}

/**
 * Test d'égalité
 * @param $periode Periode
 * @return bool Les objets sont-ils égaux ?
 */
public function equals ($periode) {}

/**
 * Vérifier que le/la periode existe en base de données
 * @return bool Le/La periode existe en base de données ?
 */
public function exists () {}

```

```

/**
 * Supprimer le/la periode
 * @return bool Opération réussie ?
 */
public function delete () {}

/**
 * Mettre à jour un champ dans la base de données
 * @param $fields array
 * @param $values array
 * @return bool Opération réussie ?
 */
protected function _set ($fields,
                          $values) {}

/**
 * Mettre à jour tous les champs dans la base de données
 * @return bool Opération réussie ?
 */
public function update () {}

/**
 * Récupérer le/la id_période
 * @return int
 */
public function getId_période () {}

/**
 * Récupérer le/la debut_période
 * @return string
 */
public function getDebut_période () {}

/**
 * Définir le/la debut_période
 * @param $debut_période string
 * @param $execute bool Exécuter la requête update ?
 * @return bool Opération réussie ?
 */
public function setDebut_période ($debut_période,
                                  $execute = true) {}

/**
 * Récupérer le/la fin_période
 * @return string
 */
public function getFin_période () {}

/**
 * Définir le/la fin_période
 * @param $fin_période string
 * @param $execute bool Exécuter la requête update ?
 * @return bool Opération réussie ?

```

```

*/
public function setFin_periode ($fin_periode,
                               $execute = true) {}

/**
 * Récupérer le/la id_tper
 * @return int
 */
public function getId_tper () {}

/**
 * Définir le/la id_tper
 * @param $id_tper int
 * @param $execute bool Exécuter la requête update ?
 * @return bool Opération réussie ?
 */
public function setId_tper ($id_tper,
                            $execute = true) {}

/**
 * ToString
 * @return string Représentation de periode sous la forme d'un string
 */
public function __toString () {}

/**
 * Sérialiser
 * @param $serialize bool Activer la sérialisation ?
 * @return string Sérialisation du/de la periode
 */
public function serialize ($serialize = true) {}

/**
 * Désérialiser
 * @param $pdo PDO
 * @param $string string Sérialisation du/de la periode
 * @param $lazyload bool Activer le chargement fainéant ?
 * @return Periode
 */
public static function unserialize (PDO $pdo,
                                    $string,
                                    $lazyload = true) {}

}

```

```
+ www > model > classes > Periode.php
```

```
/**
 * @name Periode
 * @version 03/05/2016 (dd/mm/yyyy)
 */
class Periode extends Base_Periode {

    const TABLENAME = 'PERIODE';

    const FIELDNAME_ID_PERIODE = 'id_periode';

    const FIELDNAME_DEBUT_PERIODE = 'debut_periode';

    const FIELDNAME_FIN_PERIODE = 'fin_periode';

    const FIELDNAME_ID_TPER = 'id_tper';

    public static function supprDef ($periode) {}

    /**
     * LIB_TPER: Type de période (Mensuelle, Hebdomadaire, ...)
     * DEBUT_PERIODE: Date de début de la période
     * FIN_PERIODE: Date de fin de la période
     * NUM_CONTR_C: Numéro de contrat du chauffeur concerné
     */
    public static function periodeAjout (PDO $pdo,
                                         $quote) {}

}
```

```
+ www > model > classes > Phyto_Stock.php
```

```
/**
 * @name Base_Phyto_Stock
 * @version 25/05/2016 (dd/mm/yyyy)
 * @author Axel Jalabert
 */
abstract class Base_Phyto_Stock {

    const TABLENAME = 'PHYTO_STOCK';

    const FIELDNAME_CODE_PROD = 'code_prod';

    const FIELDNAME_ETAT_PHY_STOCK = 'etat_phy_stock';

    const FIELDNAME_PH_MIN_STOCK = 'ph_min_stock';

    const FIELDNAME_PH_MAX_STOCK = 'ph_max_stock';

    /** @var PDO */
    protected $_pdo;

    /** @var array tableau pour le chargement fainéant */
    protected static $_lazyload;

    /** @var string */
    protected $_code_prod;

    /** @var string */
    protected $_etat_phy_stock;

    /** @var string */
    protected $_ph_min_stock;

    /** @var string */
    protected $_ph_max_stock;

    /**
     * Construire un(e) phyto_stock
     * @param $pdo PDO
     * @param $code_prod string
     * @param $etat_phy_stock string
     * @param $ph_min_stock string
     * @param $ph_max_stock string
     * @param $lazyload bool Activer le chargement fainéant ?
     */
    protected function __construct (PDO $pdo,
                                     $code_prod,
                                     $etat_phy_stock,
                                     $ph_min_stock,
                                     $ph_max_stock,
                                     $lazyload = true) {}
}
```

```

/**
 * Créer un(e) phyto_stock
 * @param $pdo PDO
 * @param $code_prod string
 * @param $etat_phy_stock string
 * @param $ph_min_stock string
 * @param $ph_max_stock string
 * @param $lazyload bool Activer le chargement fainéant ?
 * @return Phyto_Stock
 */
public static function create (PDO $pdo,
                               $code_prod,
                               $etat_phy_stock,
                               $ph_min_stock,
                               $ph_max_stock,
                               $lazyload = true) {}

/**
 * Compter les phyto_stocks
 * @param $pdo PDO
 * @return int Nombre de phyto_stocks
 */
public static function count (PDO $pdo) {}

/**
 * Requête de sélection
 * @param $pdo PDO
 * @param $where string|array
 * @param $orderby string|array
 * @param $limit string|array
 * @param $from string|array
 * @return PDOStatement
 */
protected static function _select (PDO $pdo,
                                    $where = null,
                                    $orderby = null,
                                    $limit = null,
                                    $from = null) {}

/**
 * Charger un(e) phyto_stock
 * @param $pdo PDO
 * @param $code_prod string
 * @param $lazyload bool Activer le chargement fainéant ?
 * @return Phyto_Stock
 */
public static function load (PDO $pdo,
                             $code_prod,
                             $lazyload = true) {}

/**
 * Recharger les données depuis la base de données
 */

```

```

public function reload () {}

/**
 * Charger tous/toutes les phyto_stocks
 * @param $pdo PDO
 * @param $lazyload bool Activer le chargement fainéant ?
 * @return Phyto_Stock[] Tableau de phyto_stocks
 */
public static function loadAll (PDO $pdo,
                                $lazyload = true) {}

/**
 * Sélectionner tous/toutes les phyto_stocks
 * @param $pdo PDO
 * @return PDOStatement
 */
public static function selectAll (PDO $pdo) {}

/**
 * Récupérer le/la phyto_stock suivant(e) d'un jeu de résultats
 * @param $pdo PDO
 * @param $pdoStatement PDOStatement
 * @param $lazyload bool Activer le chargement fainéant ?
 * @return Phyto_Stock
 */
public static function fetch (PDO $pdo,
                              PDOStatement $pdoStatement,
                              $lazyload = true) {}

/**
 * Récupérer tous/toutes les phyto_stocks d'un jeu de résultats
 * @param $pdo PDO
 * @param $pdoStatement PDOStatement
 * @param $lazyload bool Activer le chargement fainéant ?
 * @return Phyto_Stock[] Tableau de phyto_stocks
 */
public static function fetchAll (PDO $pdo,
                                 PDOStatement $pdoStatement,
                                 $lazyload = true) {}

/**
 * Test d'égalité
 * @param $phyto_stock Phyto_Stock
 * @return bool Les objets sont-ils égaux ?
 */
public function equals ($phyto_stock) {}

/**
 * Vérifier que le/la phyto_stock existe en base de données
 * @return bool Le/La phyto_stock existe en base de données ?
 */
public function exists () {}

```

```

/**
 * Supprimer le/la phyto_stock
 * @return bool Opération réussie ?
 */
public function delete () {}

/**
 * Mettre à jour un champ dans la base de données
 * @param $fields array
 * @param $values array
 * @return bool Opération réussie ?
 */
protected function _set ($fields,
                          $values) {}

/**
 * Mettre à jour tous les champs dans la base de données
 * @return bool Opération réussie ?
 */
public function update () {}

/**
 * Récupérer le/la code_prod
 * @return string
 */
public function getCode_prod () {}

/**
 * Récupérer le/la etat_phy_stock
 * @return string
 */
public function getEtat_phy_stock () {}

/**
 * Définir le/la etat_phy_stock
 * @param $etat_phy_stock string
 * @param $execute bool Exécuter la requête update ?
 * @return bool Opération réussie ?
 */
public function setEtat_phy_stock ($etat_phy_stock,
                                   $execute = true) {}

/**
 * Récupérer le/la ph_min_stock
 * @return string
 */
public function getPh_min_stock () {}

/**
 * Définir le/la ph_min_stock
 * @param $ph_min_stock string
 * @param $execute bool Exécuter la requête update ?
 * @return bool Opération réussie ?

```

```

*/
public function setPh_min_stock ($ph_min_stock,
                                $execute = true) {}

/**
 * Récupérer le/la ph_max_stock
 * @return string
 */
public function getPh_max_stock () {}

/**
 * Définir le/la ph_max_stock
 * @param $ph_max_stock string
 * @param $execute bool Exécuter la requête update ?
 * @return bool Opération réussie ?
 */
public function setPh_max_stock ($ph_max_stock,
                                $execute = true) {}

/**
 * ToString
 * @return string Représentation de phyto_stock sous la forme d'un string
 */
public function __toString () {}

/**
 * Sérialiser
 * @param $serialize bool Activer la sérialisation ?
 * @return string Sérialisation du/de la phyto_stock
 */
public function serialize ($serialize = true) {}

/**
 * Désérialiser
 * @param $pdo PDO
 * @param $string string Sérialisation du/de la phyto_stock
 * @param $lazyload bool Activer le chargement fainéant ?
 * @return Phyto_Stock
 */
public static function unserialize (PDO $pdo,
                                    $string,
                                    $lazyload = true) {}

}

```

```
+ www > model > classes > Phyto_Stock.php
```

```
/**
 * @name Phyto_Stock
 * @version 25/05/2016 (dd/mm/yyyy)
 */
class Phyto_Stock extends Base_Phyto_Stock {

    const TABLENAME = 'PHYTO_STOCK';

    const FIELDNAME_CODE_PROD = 'code_prod';

    const FIELDNAME_ETAT_PHY_STOCK = 'etat_phy_stock';

    const FIELDNAME_PH_MIN_STOCK = 'ph_min_stock';

    const FIELDNAME_PH_MAX_STOCK = 'ph_max_stock';

    public static function requete_info_produit_local (PDO $db,
                                                       $codeP) {}

    public static function type_produit_local ($type) {}

    public static function unite_produit_local ($unite) {}

    public static function volume_produit_local ($volume) {}

    public static function clp_produit_local ($clp) {}

    public static function ph_produit_local ($ph_min,
                                             $ph_max) {}

    public static function date_produit_local ($date_produit) {}

    public static function condition_local ($info_produit) {}

}
```

```
+ www > model > classes > Phyto_Uc.php
```

```
/**
 * @name Base_Phyto_Uc
 * @version 23/05/2016 (dd/mm/yyyy)
 * @author Axel Jalabert
 */
abstract class Base_Phyto_Uc {

    const TABLENAME = 'PHYTO_UC';

    const FIELDNAME_CODE_PROD = 'code_prod';

    const FIELDNAME_CODE_UC = 'code_uc';

    const FIELDNAME_QUANTITE_UC = 'quantite_uc';

    /** @var PDO */
    protected $_pdo;

    /** @var array tableau pour le chargement fainéant */
    protected static $_lazyload;

    /** @var string */
    protected $_code_prod;

    /** @var string */
    protected $_code_uc;

    /** @var string */
    protected $_quantite_uc;

    /**
     * Construire un(e) phyto_uc
     * @param $pdo PDO
     * @param $code_prod string
     * @param $code_uc string
     * @param $quantite_uc string
     * @param $lazyload bool Activer le chargement fainéant ?
     */
    protected function __construct (PDO $pdo,
                                     $code_prod,
                                     $code_uc,
                                     $quantite_uc,
                                     $lazyload = true) {}

    /**
     * Créer un(e) phyto_uc
     * @param $pdo PDO
     * @param $code_prod string
     * @param $code_uc string
     * @param $quantite_uc string
     * @param $lazyload bool Activer le chargement fainéant ?
     */
}
```

```

* @return Phyto_Uc
*/
public static function create (PDO $pdo,
                              $code_prod,
                              $code_uc,
                              $quantite_uc,
                              $lazyload = true) {}

/**
 * Compter les phyto_ucs
 * @param $pdo PDO
 * @return int Nombre de phyto_ucs
 */
public static function count (PDO $pdo) {}

/**
 * Requête de sélection
 * @param $pdo PDO
 * @param $where string|array
 * @param $orderby string|array
 * @param $limit string|array
 * @param $from string|array
 * @return PDOStatement
 */
protected static function _select (PDO $pdo,
                                    $where = null,
                                    $orderby = null,
                                    $limit = null,
                                    $from = null) {}

/**
 * Charger un(e) phyto_uc
 * @param $pdo PDO
 * @param $code_prod string
 * @param $lazyload bool Activer le chargement fainéant ?
 * @return Phyto_Uc
 */
public static function load (PDO $pdo,
                            $code_prod,
                            $lazyload = true) {}

/**
 * Recharger les données depuis la base de données
 */
public function reload () {}

/**
 * Charger tous/toutes les phyto_ucs
 * @param $pdo PDO
 * @param $lazyload bool Activer le chargement fainéant ?
 * @return Phyto_Uc[] Tableau de phyto_ucs
 */
public static function loadAll (PDO $pdo,

```

```

        $lazyload = true) {}

/**
 * Sélectionner tous/toutes les phyto_ucs
 * @param $pdo PDO
 * @return PDOStatement
 */
public static function selectAll (PDO $pdo) {}

/**
 * Récupérer le/la phyto_uc suivant(e) d'un jeu de résultats
 * @param $pdo PDO
 * @param $pdoStatement PDOStatement
 * @param $lazyload bool Activer le chargement fainéant ?
 * @return Phyto_Uc
 */
public static function fetch (PDO $pdo,
                             PDOStatement $pdoStatement,
                             $lazyload = true) {}

/**
 * Récupérer tous/toutes les phyto_ucs d'un jeu de résultats
 * @param $pdo PDO
 * @param $pdoStatement PDOStatement
 * @param $lazyload bool Activer le chargement fainéant ?
 * @return Phyto_Uc[] Tableau de phyto_ucs
 */
public static function fetchAll (PDO $pdo,
                                 PDOStatement $pdoStatement,
                                 $lazyload = true) {}

/**
 * Test d'égalité
 * @param $phyto_uc Phyto_Uc
 * @return bool Les objets sont-ils égaux ?
 */
public function equals ($phyto_uc) {}

/**
 * Vérifier que le/la phyto_uc existe en base de données
 * @return bool Le/La phyto_uc existe en base de données ?
 */
public function exists () {}

/**
 * Supprimer le/la phyto_uc
 * @return bool Opération réussie ?
 */
public function delete () {}

/**
 * Mettre à jour un champ dans la base de données
 * @param $fields array

```

```

* @param $values array
* @return bool Opération réussie ?
*/
protected function _set ($fields,
                        $values) {}

/**
* Mettre à jour tous les champs dans la base de données
* @return bool Opération réussie ?
*/
public function update () {}

/**
* Récupérer le/la code_prod
* @return string
*/
public function getCode_prod () {}

/**
* Récupérer le/la code_uc
* @return string
*/
public function getCode_uc () {}

/**
* Définir le/la code_uc
* @param $code_uc string
* @param $execute bool Exécuter la requête update ?
* @return bool Opération réussie ?
*/
public function setCode_uc ($code_uc,
                          $execute = true) {}

/**
* Récupérer le/la quantite_uc
* @return string
*/
public function getQuantite_uc () {}

/**
* Définir le/la quantite_uc
* @param $quantite_uc string
* @param $execute bool Exécuter la requête update ?
* @return bool Opération réussie ?
*/
public function setQuantite_uc ($quantite_uc,
                              $execute = true) {}

/**
* ToString
* @return string Représentation de phyto_uc sous la forme d'un string
*/
public function __toString () {}

```

```

/**
 * S rialiser
 * @param $serialize bool Activer la s rialisation ?
 * @return string S rialisation du/de la phyto_uc
 */
public function serialize ($serialize = true) {}

/**
 * D s rialiser
 * @param $pdo PDO
 * @param $string string S rialisation du/de la phyto_uc
 * @param $lazyload bool Activer le chargement fain ant ?
 * @return Phyto_Uc
 */
public static function unserialize (PDO $pdo,
                                     $string,
                                     $lazyload = true) {}
}

```

```
+ www > model > classes > Phyto_Uc.php
```

```
/**  
 * @name Phyto_Uc  
 * @version 23/05/2016 (dd/mm/yyyy)  
 */  
class Phyto_Uc extends Base_Phyto_Uc {  
  
    const TABLENAME = 'PHYTO_UC';  
  
    const FIELDNAME_CODE_PROD = 'code_prod';  
  
    const FIELDNAME_CODE_UC = 'code_uc';  
  
    const FIELDNAME_QUANTITE_UC = 'quantite_uc';  
  
}
```

```
+ www > model > classes > Produit.php
```

```
/**
 * @name Base_Produit
 * @version 23/05/2016 (dd/mm/yyyy)
 * @author Axel Jalabert
 */
abstract class Base_Produit {

    const TABLENAME = 'PRODUIT';

    const FIELDNAME_CODE_PROD = 'code_prod';

    const FIELDNAME_AMM_PROD = 'amm_prod';

    const FIELDNAME_NOM_PROD = 'nom_prod';

    const FIELDNAME_FDS_PROD = 'fds_prod';

    const FIELDNAME_DREPC_PROD = 'drepc_prod';

    const FIELDNAME_TYPE_PROD = 'type_prod';

    const FIELDNAME_UNITE_PROD = 'unite_prod';

    const FIELDNAME_DATE_FIN_UTIL_PROD = 'date_fin_util_prod';

    /** @var PDO */
    protected $_pdo;

    /** @var array tableau pour le chargement fainéant */
    protected static $_lazyload;

    /** @var string */
    protected $_code_prod;

    /** @var string */
    protected $_amm_prod;

    /** @var string */
    protected $_nom_prod;

    /** @var string */
    protected $_fds_prod;

    /** @var int */
    protected $_drepc_prod;

    /** @var int */
    protected $_type_prod;

    /** @var string */
    protected $_unite_prod;
```

```

/** @var string */
protected $_date_fin_util_prod;

/**
 * Construire un(e) produit
 * @param $pdo PDO
 * @param $code_prod string
 * @param $amm_prod string
 * @param $nom_prod string
 * @param $fds_prod string
 * @param $drepc_prod int
 * @param $type_prod int
 * @param $unite_prod string
 * @param $date_fin_util_prod string
 * @param $lazyload bool Activer le chargement fainéant ?
 */
protected function __construct (PDO $pdo,
                                $code_prod,
                                $amm_prod,
                                $nom_prod,
                                $fds_prod,
                                $drepc_prod,
                                $type_prod,
                                $unite_prod,
                                $date_fin_util_prod,
                                $lazyload = true) {}

/**
 * Créer un(e) produit
 * @param $pdo PDO
 * @param $code_prod string
 * @param $amm_prod string
 * @param $nom_prod string
 * @param $fds_prod string
 * @param $drepc_prod int
 * @param $type_prod int
 * @param $unite_prod string
 * @param $date_fin_util_prod string
 * @param $lazyload bool Activer le chargement fainéant ?
 * @return Produit
 */
public static function create (PDO $pdo,
                              $code_prod,
                              $amm_prod,
                              $nom_prod,
                              $fds_prod,
                              $drepc_prod,
                              $type_prod,
                              $unite_prod,
                              $date_fin_util_prod,
                              $lazyload = true) {}

/**

```

```

* Compter les produits
* @param $pdo PDO
* @return int Nombre de produits
*/
public static function count (PDO $pdo) {}

/**
* Requête de sélection
* @param $pdo PDO
* @param $where string|array
* @param $orderby string|array
* @param $limit string|array
* @param $from string|array
* @return PDOStatement
*/
protected static function _select (PDO $pdo,
                                     $where = null,
                                     $orderby = null,
                                     $limit = null,
                                     $from = null) {}

/**
* Charger un(e) produit
* @param $pdo PDO
* @param $code_prod string
* @param $lazyload bool Activer le chargement fainéant ?
* @return Produit
*/
public static function load (PDO $pdo,
                             $code_prod,
                             $lazyload = true) {}

/**
* Recharger les données depuis la base de données
*/
public function reload () {}

/**
* Charger tous/toutes les produits
* @param $pdo PDO
* @param $lazyload bool Activer le chargement fainéant ?
* @return Produit[] Tableau de produits
*/
public static function loadAll (PDO $pdo,
                                $lazyload = true) {}

/**
* Sélectionner tous/toutes les produits
* @param $pdo PDO
* @return PDOStatement
*/
public static function selectAll (PDO $pdo) {}

```

```

/**
 * Récupérer le/la produit suivant(e) d'un jeu de résultats
 * @param $pdo PDO
 * @param $pdoStatement PDOStatement
 * @param $lazyload bool Activer le chargement fainéant ?
 * @return Produit
 */
public static function fetch (PDO $pdo,
                             PDOStatement $pdoStatement,
                             $lazyload = true) {}

/**
 * Récupérer tous/toutes les produits d'un jeu de résultats
 * @param $pdo PDO
 * @param $pdoStatement PDOStatement
 * @param $lazyload bool Activer le chargement fainéant ?
 * @return Produit[] Tableau de produits
 */
public static function fetchAll (PDO $pdo,
                                 PDOStatement $pdoStatement,
                                 $lazyload = true) {}

/**
 * Test d'égalité
 * @param $produit Produit
 * @return bool Les objets sont-ils égaux ?
 */
public function equals ($produit) {}

/**
 * Vérifier que le/la produit existe en base de données
 * @return bool Le/La produit existe en base de données ?
 */
public function exists () {}

/**
 * Supprimer le/la produit
 * @return bool Opération réussie ?
 */
public function delete () {}

/**
 * Mettre à jour un champ dans la base de données
 * @param $fields array
 * @param $values array
 * @return bool Opération réussie ?
 */
protected function _set ($fields,
                        $values) {}

/**
 * Mettre à jour tous les champs dans la base de données
 * @return bool Opération réussie ?

```

```

*/
public function update () {}

/**
 * Récupérer le/la code_prod
 * @return string
 */
public function getCode_prod () {}

/**
 * Récupérer le/la amm_prod
 * @return string
 */
public function getAmm_prod () {}

/**
 * Définir le/la amm_prod
 * @param $amm_prod string
 * @param $execute bool Exécuter la requête update ?
 * @return bool Opération réussie ?
 */
public function setAmm_prod ($amm_prod,
                            $execute = true) {}

/**
 * Récupérer le/la nom_prod
 * @return string
 */
public function getNom_prod () {}

/**
 * Définir le/la nom_prod
 * @param $nom_prod string
 * @param $execute bool Exécuter la requête update ?
 * @return bool Opération réussie ?
 */
public function setNom_prod ($nom_prod,
                             $execute = true) {}

/**
 * Récupérer le/la fds_prod
 * @return string
 */
public function getFds_prod () {}

/**
 * Définir le/la fds_prod
 * @param $fds_prod string
 * @param $execute bool Exécuter la requête update ?
 * @return bool Opération réussie ?
 */
public function setFds_prod ($fds_prod,
                             $execute = true) {}

```

```

/**
 * Récupérer le/la drepc_prod
 * @return int
 */
public function getDrepc_prod () {}

/**
 * Définir le/la drepc_prod
 * @param $drepc_prod int
 * @param $execute bool Exécuter la requête update ?
 * @return bool Opération réussie ?
 */
public function setDrepc_prod ($drepc_prod,
                              $execute = true) {}

/**
 * Récupérer le/la type_prod
 * @return int
 */
public function getType_prod () {}

/**
 * Définir le/la type_prod
 * @param $type_prod int
 * @param $execute bool Exécuter la requête update ?
 * @return bool Opération réussie ?
 */
public function setType_prod ($type_prod,
                              $execute = true) {}

/**
 * Récupérer le/la unite_prod
 * @return string
 */
public function getUnite_prod () {}

/**
 * Définir le/la unite_prod
 * @param $unite_prod string
 * @param $execute bool Exécuter la requête update ?
 * @return bool Opération réussie ?
 */
public function setUnite_prod ($unite_prod,
                              $execute = true) {}

/**
 * Récupérer le/la date_fin_util_prod
 * @return string
 */
public function getDate_fin_util_prod () {}

/**
 * Définir le/la date_fin_util_prod

```

```

* @param $date_fin_util_prod string
* @param $execute bool Exécuter la requête update ?
* @return bool Opération réussie ?
*/
public function setDate_fin_util_prod ($date_fin_util_prod,
                                       $execute = true) {}

/**
* ToString
* @return string Représentation de produit sous la forme d'un string
*/
public function __toString () {}

/**
* Sérialiser
* @param $serialize bool Activer la sérialisation ?
* @return string Sérialisation du/de la produit
*/
public function serialize ($serialize = true) {}

/**
* Désérialiser
* @param $pdo PDO
* @param $string string Sérialisation du/de la produit
* @param $lazyload bool Activer le chargement fainéant ?
* @return Produit
*/
public static function unserialize (PDO $pdo,
                                    $string,
                                    $lazyload = true) {}
}

```

```
+ www > model > classes > Produit.php
```

```
/**
 * @name Produit
 * @version 18/05/2016 (dd/mm/yyyy)
 */
class Produit extends Base_Produit {

    const TABLENAME = 'PRODUIT';

    const FIELDNAME_CODE_PROD = 'code_prod';

    const FIELDNAME_AMM_PROD = 'amm_prod';

    const FIELDNAME_NOM_PROD = 'nom_prod';

    const FIELDNAME_FDS_PROD = 'fds_prod';

    const FIELDNAME_DREPC_PROD = 'drepc_prod';

    const FIELDNAME_TYPE_PROD = 'type_prod';

    const FIELDNAME_UNITE_PROD = 'unite_prod';

    const FIELDNAME_DATE_FIN_UTIL_PROD = 'date_fin_util_prod';

    public static function comparaisonFinal ($produit1,
                                             $produit2 = false,
                                             $produit3 = false) {}

    public static function compareMelange ($produit1,
                                           $produit2 = false,
                                           $produit3 = false) {}

    public static function compareCategorie ($categorie_prod1,
                                             $categorie_prod2 = false) {}

    public static function definitionCategorieProduit ($produit) {}

    public static function listProduit (PDO $pdo,
                                        $type) {}

    public static function listProduitAll (PDO $pdo) {}

    public static function infoProduit (PDO $pdo,
                                        $prod,
                                        $num = 1,
                                        $code = false) {}

    public static function infoProduit2 (PDO $pdo,
                                        $prod,
                                        $num = 1,
                                        $code = false) {}
}
```

}

```
+ www > model > classes > QteProdCli.php
```

```
class QteProdCli {  
  
    private $quantite_qteprodcli;  
  
    public function setQuantiteQteProdCli ($quantite_qteprodcli) {}  
  
    public function getQuantiteQteProdCli () {}  
  
}
```

```
+ www > model > classes > QteProdEntrep.php
```

```
class QteProdEntrep {  
  
    private $quantite_qteprodentrep;  
  
    public function setQuantiteQteProdEntrep ($quantite_qteprodentrep) {}  
  
    public function getQuantiteQteProdEntrep () {}  
  
}
```

```
+ www > model > classes > Risque.php
```

```
class Risque {  
  
    private $id_risque;  
  
    private $phrase_risque;  
  
    public function setIdRisque ($id_risque) {}  
  
    public function getIdRisque () {}  
  
    public function setPhraseRisque ($phrase_risque) {}  
  
    public function getPhraseRisque () {}  
  
}
```

```
+ www > model > classes > Situer.php
```

```
/**
 * @name Base_Situer
 * @version 19/05/2016 (dd/mm/yyyy)
 * @author Axel Jalabert
 */
abstract class Base_Situer {

    const TABLENAME = 'SITUER';

    const FIELDNAME_NUM_CULT = 'num_cult';

    const FIELDNAME_ID_CHANT = 'id_chant';

    /** @var PDO */
    protected $_pdo;

    /** @var array tableau pour le chargement fainéant */
    protected static $_lazyload;

    /** @var string numéro de la culture associée */
    protected $_num_cult;

    /** @var int id du chantier associé */
    protected $_id_chant;

    /**
     * Construire un(e) situer
     * @param $pdo PDO
     * @param $num_cult string Numéro de la culture associée
     * @param $id_chant int ID du chantier associé
     * @param $lazyload bool Activer le chargement fainéant ?
     */
    protected function __construct (PDO $pdo,
                                     $num_cult,
                                     $id_chant,
                                     $lazyload = true) {}

    /**
     * Créer un(e) situer
     * @param $pdo PDO
     * @param $num_cult string Numéro de la culture associée
     * @param $id_chant int ID du chantier associé
     * @param $lazyload bool Activer le chargement fainéant ?
     * @return Situer
     */
    public static function create (PDO $pdo,
                                   $num_cult,
                                   $id_chant,
                                   $lazyload = true) {}

    /**
```

```

* Compter les situers
* @param $pdo PDO
* @return int Nombre de situers
*/
public static function count (PDO $pdo) {}

/**
* Requête de sélection
* @param $pdo PDO
* @param $where string|array
* @param $orderby string|array
* @param $limit string|array
* @param $from string|array
* @return PDOStatement
*/
protected static function _select (PDO $pdo,
                                     $where = null,
                                     $orderby = null,
                                     $limit = null,
                                     $from = null) {}

/**
* Charger un(e) situer
* @param $pdo PDO
* @param $num_cult string Numéro de la culture associée
* @param $lazyload bool Activer le chargement fainéant ?
* @return Situer
*/
public static function load (PDO $pdo,
                             $num_cult,
                             $lazyload = true) {}

/**
* Recharger les données depuis la base de données
*/
public function reload () {}

/**
* Charger tous/toutes les situers
* @param $pdo PDO
* @param $lazyload bool Activer le chargement fainéant ?
* @return Situer[] Tableau de situers
*/
public static function loadAll (PDO $pdo,
                                $lazyload = true) {}

/**
* Sélectionner tous/toutes les situers
* @param $pdo PDO
* @return PDOStatement
*/
public static function selectAll (PDO $pdo) {}

```

```

/**
 * Récupérer le/la situer suivant(e) d'un jeu de résultats
 * @param $pdo PDO
 * @param $pdoStatement PDOStatement
 * @param $lazyload bool Activer le chargement fainéant ?
 * @return Situer
 */
public static function fetch (PDO $pdo,
                             PDOStatement $pdoStatement,
                             $lazyload = true) {}

/**
 * Récupérer tous/toutes les situers d'un jeu de résultats
 * @param $pdo PDO
 * @param $pdoStatement PDOStatement
 * @param $lazyload bool Activer le chargement fainéant ?
 * @return Situer[] Tableau de situers
 */
public static function fetchAll (PDO $pdo,
                                PDOStatement $pdoStatement,
                                $lazyload = true) {}

/**
 * Test d'égalité
 * @param $siter Situer
 * @return bool Les objets sont-ils égaux ?
 */
public function equals ($siter) {}

/**
 * Vérifier que le/la situer existe en base de données
 * @return bool Le/La situer existe en base de données ?
 */
public function exists () {}

/**
 * Supprimer le/la situer
 * @return bool Opération réussie ?
 */
public function delete () {}

/**
 * Mettre à jour un champ dans la base de données
 * @param $fields array
 * @param $values array
 * @return bool Opération réussie ?
 */
protected function _set ($fields,
                        $values) {}

/**
 * Mettre à jour tous les champs dans la base de données
 * @return bool Opération réussie ?

```

```

*/
public function update () {}

/**
 * Récupérer le/la num_cult
 * @return string Numéro de la culture associée
 */
public function getNum_cult () {}

/**
 * Récupérer le/la id_chant
 * @return int ID du chantier associé
 */
public function getId_chant () {}

/**
 * Définir le/la id_chant
 * @param $id_chant int ID du chantier associé
 * @param $execute bool Exécuter la requête update ?
 * @return bool Opération réussie ?
 */
public function setId_chant ($id_chant,
                             $execute = true) {}

/**
 * ToString
 * @return string Représentation de situer sous la forme d'un string
 */
public function __toString () {}

/**
 * Sérialiser
 * @param $serialize bool Activer la sérialisation ?
 * @return string Sérialisation du/de la situer
 */
public function serialize ($serialize = true) {}

/**
 * Désérialiser
 * @param $pdo PDO
 * @param $string string Sérialisation du/de la situer
 * @param $lazyload bool Activer le chargement fainéant ?
 * @return Situer
 */
public static function unserialize (PDO $pdo,
                                     $string,
                                     $lazyload = true) {}
}

```

```
+ www > model > classes > Situer.php
```

```
/**  
 * @name Situer  
 * @version 19/05/2016 (dd/mm/yyyy)  
 */  
class Situer extends Base_Situer {  
  
    const TABLENAME = 'SITUER';  
  
    const FIELDNAME_NUM_CULT = 'num_cult';  
  
    const FIELDNAME_ID_CHANT = 'id_chant';  
  
    public static function infoSituation (PDO $pdo,  
                                           $fiche) {}  
  
}
```

```
+ www > model > classes > Traiter_Sur.php
```

```
/**
 * @name Base_Traiter_Sur
 * @version 03/05/2016 (dd/mm/yyyy)
 * @author Axel Jalabert
 */
abstract class Base_Traiter_Sur {

    const TABLENAME = 'TRAITER_SUR';

    const FIELDNAME_NUM_CONTR_C = 'num_contr_c';

    const FIELDNAME_ID_PERIODE = 'id_periode';

    /** @var PDO */
    protected $_pdo;

    /** @var array tableau pour le chargement fainéant */
    protected static $_lazyload;

    /** @var int */
    protected $_num_contr_c;

    /** @var int */
    protected $_id_periode;

    /**
     * Construire un(e) traiter_sur
     * @param $pdo PDO
     * @param $num_contr_c int
     * @param $id_periode int
     * @param $lazyload bool Activer le chargement fainéant ?
     */
    protected function __construct (PDO $pdo,
                                     $num_contr_c,
                                     $id_periode,
                                     $lazyload = true) {}

    /**
     * Créer un(e) traiter_sur
     * @param $pdo PDO
     * @param $num_contr_c int
     * @param $id_periode int
     * @param $lazyload bool Activer le chargement fainéant ?
     * @return Traiter_Sur
     */
    public static function create (PDO $pdo,
                                   $num_contr_c,
                                   $id_periode,
                                   $lazyload = true) {}

    /**
```

```

* Compter les traiter_surs
* @param $pdo PDO
* @return int Nombre de traiter_surs
*/
public static function count (PDO $pdo) {}

/**
* Requête de sélection
* @param $pdo PDO
* @param $where string|array
* @param $orderby string|array
* @param $limit string|array
* @param $from string|array
* @return PDOStatement
*/
protected static function _select (PDO $pdo,
                                     $where = null,
                                     $orderby = null,
                                     $limit = null,
                                     $from = null) {}

/**
* Charger un(e) traiter_sur
* @param $pdo PDO
* @param $num_contr_c int
* @param $id_periode int
* @param $lazyload bool Activer le chargement fainéant ?
* @return Traiter_Sur
*/
public static function load (PDO $pdo,
                             $num_contr_c,
                             $id_periode,
                             $lazyload = true) {}

/**
* Charger tous/toutes les traiter_surs
* @param $pdo PDO
* @param $lazyload bool Activer le chargement fainéant ?
* @return Traiter_Sur[] Tableau de traiter_surs
*/
public static function loadAll (PDO $pdo,
                                $lazyload = true) {}

/**
* Sélectionner tous/toutes les traiter_surs
* @param $pdo PDO
* @return PDOStatement
*/
public static function selectAll (PDO $pdo) {}

/**
* Récupérer le/la traiter_sur suivant(e) d'un jeu de résultats
* @param $pdo PDO

```

```

* @param $pdoStatement PDOStatement
* @param $lazyload bool Activer le chargement fainéant ?
* @return Traiter_Sur
*/
public static function fetch (PDO $pdo,
                             PDOStatement $pdoStatement,
                             $lazyload = true) {}

/**
 * Récupérer tous/toutes les traiter_surs d'un jeu de résultats
 * @param $pdo PDO
 * @param $pdoStatement PDOStatement
 * @param $lazyload bool Activer le chargement fainéant ?
 * @return Traiter_Sur[] Tableau de traiter_surs
 */
public static function fetchAll (PDO $pdo,
                                 PDOStatement $pdoStatement,
                                 $lazyload = true) {}

/**
 * Test d'égalité
 * @param $traiter_sur Traiter_Sur
 * @return bool Les objets sont-ils égaux ?
 */
public function equals ($traiter_sur) {}

/**
 * Vérifier que le/la traiter_sur existe en base de données
 * @return bool Le/La traiter_sur existe en base de données ?
 */
public function exists () {}

/**
 * Supprimer le/la traiter_sur
 * @return bool Opération réussie ?
 */
public function delete () {}

/**
 * Récupérer le/la num_contr_c
 * @return int
 */
public function getNum_contr_c () {}

/**
 * Récupérer le/la id_periode
 * @return int
 */
public function getId_periode () {}

/**
 * ToString
 * @return string Représentation de traiter_sur sous la forme d'un string

```

```

*/
public function __toString () {}

/**
 * Sérialiser
 * @param $serialize bool Activer la sérialisation ?
 * @return string Sérialisation du/de la traiter_sur
 */
public function serialize ($serialize = true) {}

/**
 * Désérialiser
 * @param $pdo PDO
 * @param $string string Sérialisation du/de la traiter_sur
 * @param $lazyload bool Activer le chargement fainéant ?
 * @return Traiter_Sur
 */
public static function unserialize (PDO $pdo,
                                     $string,
                                     $lazyload = true) {}

}

```

```
+ www > model > classes > Traiter_Sur.php
```

```
/**  
 * @name Traiter_Sur  
 * @version 03/05/2016 (dd/mm/yyyy)  
 */  
class Traiter_Sur extends Base_Traiter_Sur {  
  
    const TABLENAME = 'TRAITER_SUR';  
  
    const FIELDNAME_NUM_CONTR_C = 'num_contr_c';  
  
    const FIELDNAME_ID_PERIODE = 'id_periode';  
  
}
```

```
+ www > model > classes > TypeContrat.php
```

```
/**
 * @name Model_Base_Type_Contrat
 * @version 02/05/2016 (dd/mm/yyyy)
 * @author Axel Jalabert
 */
abstract class Model_Base_Type_Contrat {

    const TABLENAME = 'TYPE_CONTRAT';

    const FIELDNAME_LIB_TCONT = 'lib_tcont';

    /** @var PDO */
    protected $_pdo;

    /** @var array tableau pour le chargement fainéant */
    protected static $_lazyload;

    /** @var string */
    protected $_lib_tcont;

    /**
     * Construire un(e) type_contrat
     * @param $pdo PDO
     * @param $lib_tcont string
     * @param $lazyload bool Activer le chargement fainéant ?
     */
    protected function __construct (PDO $pdo,
                                     $lib_tcont,
                                     $lazyload = true) {}

    /**
     * Créer un(e) type_contrat
     * @param $pdo PDO
     * @param $lib_tcont string
     * @param $lazyload bool Activer le chargement fainéant ?
     * @return Model_Type_Contrat
     */
    public static function create (PDO $pdo,
                                   $lib_tcont,
                                   $lazyload = true) {}

    /**
     * Compter les type_contrats
     * @param $pdo PDO
     * @return int Nombre de type_contrats
     */
    public static function count (PDO $pdo) {}

    /**
     * Requête de sélection
     * @param $pdo PDO
     */
}
```

```

* @param $where string|array
* @param $orderby string|array
* @param $limit string|array
* @param $from string|array
* @return PDOStatement
*/
protected static function _select (PDO $pdo,
                                     $where = null,
                                     $orderby = null,
                                     $limit = null,
                                     $from = null) {}

/**
 * Charger un(e) type_contrat
 * @param $pdo PDO
 * @param $lib_tcont string
 * @param $lazyload bool Activer le chargement fainéant ?
 * @return Model_Type_Contrat
 */
public static function load (PDO $pdo,
                             $lib_tcont,
                             $lazyload = true) {}

/**
 * Charger tous/toutes les type_contrats
 * @param $pdo PDO
 * @param $lazyload bool Activer le chargement fainéant ?
 * @return Model_Type_Contrat[] Tableau de type_contrats
 */
public static function loadAll (PDO $pdo,
                                $lazyload = true) {}

/**
 * Sélectionner tous/toutes les type_contrats
 * @param $pdo PDO
 * @return PDOStatement
 */
public static function selectAll (PDO $pdo) {}

/**
 * Récupérer le/la type_contrat suivant(e) d'un jeu de résultats
 * @param $pdo PDO
 * @param $pdoStatement PDOStatement
 * @param $lazyload bool Activer le chargement fainéant ?
 * @return Model_Type_Contrat
 */
public static function fetch (PDO $pdo,
                              PDOStatement $pdoStatement,
                              $lazyload = true) {}

/**
 * Récupérer tous/toutes les type_contrats d'un jeu de résultats
 * @param $pdo PDO

```

```

* @param $pdoStatement PDOStatement
* @param $lazyload bool Activer le chargement fainéant ?
* @return Model_Type_Contrat[] Tableau de type_contrats
*/
public static function fetchAll (PDO $pdo,
                                PDOStatement $pdoStatement,
                                $lazyload = true) {}

/**
* Test d'égalité
* @param $type_contrat Model_Type_Contrat
* @return bool Les objets sont-ils égaux ?
*/
public function equals ($type_contrat) {}

/**
* Vérifier que le/la type_contrat existe en base de données
* @return bool Le/La type_contrat existe en base de données ?
*/
public function exists () {}

/**
* Supprimer le/la type_contrat
* @return bool Opération réussie ?
*/
public function delete () {}

/**
* Récupérer le/la lib_tcont
* @return string
*/
public function getLib_tcont () {}

/**
* ToString
* @return string Représentation de type_contrat sous la forme d'un string
*/
public function __toString () {}

/**
* Sérialiser
* @param $serialize bool Activer la sérialisation ?
* @return string Sérialisation du/de la type_contrat
*/
public function serialize ($serialize = true) {}

/**
* Désérialiser
* @param $pdo PDO
* @param $string string Sérialisation du/de la type_contrat
* @param $lazyload bool Activer le chargement fainéant ?
* @return Model_Type_Contrat
*/

```

```
public static function unserialize (PDO $pdo,  
    $string,  
    $lazyload = true) {}  
  
}
```

```
+ www > model > classes > TypeContrat.php
```

```
/**  
 * @name Model_Type_Contrat  
 * @version 02/05/2016 (dd/mm/yyyy)  
 */  
class Model_Type_Contrat extends Model_Base_Type_Contrat {  
  
    const TABLENAME = 'TYPE_CONTRAT';  
  
    const FIELDNAME_LIB_TCONT = 'lib_tcont';  
  
}
```

```
+ www > model > classes > TypeCulture.php
```

```
class TypeCulture {  
  
    private $libelle_typeculture;  
  
    public function setLibelleTypeCulture ($libelle_typeculture) {}  
  
    public function getLibelleTypeCulture () {}  
  
}
```

```
+ www > model > classes > Type_Periode.php
```

```
/**
 * @name Base_Type_Periode
 * @version 03/05/2016 (dd/mm/yyyy)
 * @author Axel Jalabert
 */
abstract class Base_Type_Periode {

    const TABLENAME = 'TYPE_PERIODE';

    const FIELDNAME_ID_TPER = 'id_tper';

    const FIELDNAME_LIB_TPER = 'lib_tper';

    /** @var PDO */
    protected $_pdo;

    /** @var array tableau pour le chargement fainéant */
    protected static $_lazyload;

    /** @var int */
    protected $_id_tper;

    /** @var string */
    protected $_lib_tper;

    /**
     * Construire un(e) type_periode
     * @param $pdo PDO
     * @param $id_tper int
     * @param $lib_tper string
     * @param $lazyload bool Activer le chargement fainéant ?
     */
    protected function __construct (PDO $pdo,
                                     $id_tper,
                                     $lib_tper,
                                     $lazyload = true) {}

    /**
     * Créer un(e) type_periode
     * @param $pdo PDO
     * @param $id_tper int
     * @param $lib_tper string
     * @param $lazyload bool Activer le chargement fainéant ?
     * @return Type_Periode
     */
    public static function create (PDO $pdo,
                                   $id_tper,
                                   $lib_tper,
                                   $lazyload = true) {}

    /**
```

```

* Compter les type_periodes
* @param $pdo PDO
* @return int Nombre de type_periodes
*/
public static function count (PDO $pdo) {}

/**
* Requête de sélection
* @param $pdo PDO
* @param $where string|array
* @param $orderby string|array
* @param $limit string|array
* @param $from string|array
* @return PDOStatement
*/
protected static function _select (PDO $pdo,
                                     $where = null,
                                     $orderby = null,
                                     $limit = null,
                                     $from = null) {}

/**
* Charger un(e) type_periode
* @param $pdo PDO
* @param $id_tper int
* @param $lazyload bool Activer le chargement fainéant ?
* @return Type_Periode
*/
public static function load (PDO $pdo,
                             $id_tper,
                             $lazyload = true) {}

/**
* Recharger les données depuis la base de données
*/
public function reload () {}

/**
* Charger tous/toutes les type_periodes
* @param $pdo PDO
* @param $lazyload bool Activer le chargement fainéant ?
* @return Type_Periode[] Tableau de type_periodes
*/
public static function loadAll (PDO $pdo,
                                $lazyload = true) {}

/**
* Sélectionner tous/toutes les type_periodes
* @param $pdo PDO
* @return PDOStatement
*/
public static function selectAll (PDO $pdo) {}

```

```

/**
 * Récupérer le/la type_période suivant(e) d'un jeu de résultats
 * @param $pdo PDO
 * @param $pdoStatement PDOStatement
 * @param $lazyload bool Activer le chargement fainéant ?
 * @return Type_Période
 */
public static function fetch (PDO $pdo,
                             PDOStatement $pdoStatement,
                             $lazyload = true) {}

/**
 * Récupérer tous/toutes les type_périodes d'un jeu de résultats
 * @param $pdo PDO
 * @param $pdoStatement PDOStatement
 * @param $lazyload bool Activer le chargement fainéant ?
 * @return Type_Période[] Tableau de type_périodes
 */
public static function fetchAll (PDO $pdo,
                                 PDOStatement $pdoStatement,
                                 $lazyload = true) {}

/**
 * Test d'égalité
 * @param $type_période Type_Période
 * @return bool Les objets sont-ils égaux ?
 */
public function equals ($type_période) {}

/**
 * Vérifier que le/la type_période existe en base de données
 * @return bool Le/La type_période existe en base de données ?
 */
public function exists () {}

/**
 * Supprimer le/la type_période
 * @return bool Opération réussie ?
 */
public function delete () {}

/**
 * Mettre à jour un champ dans la base de données
 * @param $fields array
 * @param $values array
 * @return bool Opération réussie ?
 */
protected function _set ($fields,
                         $values) {}

/**
 * Mettre à jour tous les champs dans la base de données
 * @return bool Opération réussie ?
 */

```

```

*/
public function update () {}

/**
 * Récupérer le/la id_tper
 * @return int
 */
public function getId_tper () {}

/**
 * Sélectionner les périodes
 * @return PDOStatement
 */
public function selectPeriodes () {}

/**
 * Récupérer le/la lib_tper
 * @return string
 */
public function getLib_tper () {}

/**
 * Définir le/la lib_tper
 * @param $lib_tper string
 * @param $execute bool Exécuter la requête update ?
 * @return bool Opération réussie ?
 */
public function setLib_tper ($lib_tper,
                           $execute = true) {}

/**
 * ToString
 * @return string Représentation de type_période sous la forme d'un string
 */
public function __toString () {}

/**
 * Sérialiser
 * @param $serialize bool Activer la sérialisation ?
 * @return string Sérialisation du/de la type_période
 */
public function serialize ($serialize = true) {}

/**
 * Désérialiser
 * @param $pdo PDO
 * @param $string string Sérialisation du/de la type_période
 * @param $lazyload bool Activer le chargement fainéant ?
 * @return Type_Période
 */
public static function unserialize (PDO $pdo,
                                    $string,
                                    $lazyload = true) {}

```

}

```
+ www > model > classes > Type_Periode.php

/**
 * @name Type_Periode
 * @version 03/05/2016 (dd/mm/yyyy)
 */
class Type_Periode extends Base_Type_Periode {

    const TABLENAME = 'TYPE_PERIODE';

    const FIELDNAME_ID_TPER = 'id_tper';

    const FIELDNAME_LIB_TPER = 'lib_tper';

}
```

```
+ www > model > classes > Variete.php
```

```
/**
 * @name Base_Variete
 * @version 30/05/2016 (dd/mm/yyyy)
 * @author GROS FDP
 */
abstract class Base_Variete {

    const TABLENAME = 'VARIETE';

    const FIELDNAME_CODE_VAR = 'code_var';

    const FIELDNAME_LIB_VAR = 'lib_var';

    /** @var PDO */
    protected $_pdo;

    /** @var array tableau pour le chargement fainéant */
    protected static $_lazyload;

    /** @var int */
    protected $_code_var;

    /** @var string */
    protected $_lib_var;

    /**
     * Construire un(e) variete
     * @param $pdo PDO
     * @param $code_var int
     * @param $lib_var string
     * @param $lazyload bool Activer le chargement fainéant ?
     */
    protected function __construct (PDO $pdo,
                                    $code_var,
                                    $lib_var,
                                    $lazyload = true) {}

    /**
     * Créer un(e) variete
     * @param $pdo PDO
     * @param $code_var int
     * @param $lib_var string
     * @param $lazyload bool Activer le chargement fainéant ?
     * @return Variete
     */
    public static function create (PDO $pdo,
                                   $code_var,
                                   $lib_var,
                                   $lazyload = true) {}

    /**
```

```

* Compter les varietes
* @param $pdo PDO
* @return int Nombre de varietes
*/
public static function count (PDO $pdo) {}

/**
* Requête de sélection
* @param $pdo PDO
* @param $where string|array
* @param $orderby string|array
* @param $limit string|array
* @param $from string|array
* @return PDOStatement
*/
protected static function _select (PDO $pdo,
                                     $where = null,
                                     $orderby = null,
                                     $limit = null,
                                     $from = null) {}

/**
* Charger un(e) variete
* @param $pdo PDO
* @param $code_var int
* @param $lazyload bool Activer le chargement fainéant ?
* @return Variete
*/
public static function load (PDO $pdo,
                             $code_var,
                             $lazyload = true) {}

/**
* Recharger les données depuis la base de données
*/
public function reload () {}

/**
* Charger tous/toutes les varietes
* @param $pdo PDO
* @param $lazyload bool Activer le chargement fainéant ?
* @return Variete[] Tableau de varietes
*/
public static function loadAll (PDO $pdo,
                                $lazyload = true) {}

/**
* Sélectionner tous/toutes les varietes
* @param $pdo PDO
* @return PDOStatement
*/
public static function selectAll (PDO $pdo) {}

```

```

/**
 * Récupérer le/la variété suivant(e) d'un jeu de résultats
 * @param $pdo PDO
 * @param $pdoStatement PDOStatement
 * @param $lazyload bool Activer le chargement fainéant ?
 * @return Variete
 */
public static function fetch (PDO $pdo,
                             PDOStatement $pdoStatement,
                             $lazyload = true) {}

/**
 * Récupérer tous/toutes les variétés d'un jeu de résultats
 * @param $pdo PDO
 * @param $pdoStatement PDOStatement
 * @param $lazyload bool Activer le chargement fainéant ?
 * @return Variete[] Tableau de variétés
 */
public static function fetchAll (PDO $pdo,
                                PDOStatement $pdoStatement,
                                $lazyload = true) {}

/**
 * Test d'égalité
 * @param $variete Variete
 * @return bool Les objets sont-ils égaux ?
 */
public function equals ($variete) {}

/**
 * Vérifier que le/la variété existe en base de données
 * @return bool Le/La variété existe en base de données ?
 */
public function exists () {}

/**
 * Supprimer le/la variété
 * @return bool Opération réussie ?
 */
public function delete () {}

/**
 * Mettre à jour un champ dans la base de données
 * @param $fields array
 * @param $values array
 * @return bool Opération réussie ?
 */
protected function _set ($fields,
                        $values) {}

/**
 * Mettre à jour tous les champs dans la base de données
 * @return bool Opération réussie ?

```

```

*/
public function update () {}

/**
 * Récupérer le/la code_var
 * @return int
 */
public function getCode_var () {}

/**
 * Récupérer le/la lib_var
 * @return string
 */
public function getLib_var () {}

/**
 * Définir le/la lib_var
 * @param $lib_var string
 * @param $execute bool Exécuter la requête update ?
 * @return bool Opération réussie ?
 */
public function setLib_var ($lib_var,
                           $execute = true) {}

/**
 * ToString
 * @return string Représentation de variete sous la forme d'un string
 */
public function __toString () {}

/**
 * Sérialiser
 * @param $serialize bool Activer la sérialisation ?
 * @return string Sérialisation du/de la variete
 */
public function serialize ($serialize = true) {}

/**
 * Désérialiser
 * @param $pdo PDO
 * @param $string string Sérialisation du/de la variete
 * @param $lazyload bool Activer le chargement fainéant ?
 * @return Variete
 */
public static function unserialize (PDO $pdo,
                                    $string,
                                    $lazyload = true) {}
}

```

```
+ www > model > classes > Variete.php
```

```
/**
 * @name Variete
 * @version 30/05/2016 (dd/mm/yyyy)
 */
class Variete extends Base_Variete {

    const TABLENAME = 'VARIETE';

    const FIELDNAME_CODE_VAR = 'code_var';

    const FIELDNAME_LIB_VAR = 'lib_var';

    public static function infoVariete (PDO $pdo,
                                         $codevar) {}

    public static function listVarieteAll (PDO $pdo) {}

}
```

```
+ www > model > classes > gestion_admin_ajax_image.php
```

```
class gestion_admin_ajax_image {

    private $extension;

    private $maxsize;

    private $maxwidth;

    private $maxheight;

    private $flash;

    private $name;

    private function setImageError1 () {}

    private function setImageError2 () {}

    private function setImageError3 () {}

    private function setImageError4 () {}

    private function setImageError5 () {}

    private function setImageError6 () {}

    private function setImageError7 () {}

    public function getName () {}

    public function getFlash () {}

    public function setFlash ($flash) {}

    public function getMaxWidth () {}

    public function setMaxWidth ($maxwidth) {}

    public function getMaxHeight () {}

    public function setMaxHeight ($maxheight) {}

    public function __construct () {}

    private function myalnum ($str,
                             $rep = '',
                             $lwr = false) {}

    public function uploadImage ($name,
                                $dir) {}
}
```

}

```
+ www > model > classes > index_content.php
```

```
/**
 * les Heredoc ne doivent pas être indentés
 * @version 06/04/20168 (dd/mm/yyyy)
 */
class index_content {

    /**
     * Class Database
     * @var $db Database
     */
    private $db;

    /**
     * chemin des images
     * @var string
     */
    public $img;

    /**
     * @param $db Database
     */
    public function __construct (Database $db) {}

    /**
     * <$name class="array$class">$content</$name>
     * @param $name string
     * @param $content string
     * @param $class array
     * @return string
     */
    public function create_element ($name,
                                    $content,
                                    $class) {}

    /**
     * echo htmlentities
     * @param $i string
     * @return string
     */
    public function h (string $i) {}

    /**
     * Retourne les ANCREs de la section HEADER
     * _key -> titre
     * _value -> ancre
     * @return object
     */
    public function get_header_ancore () {}

    public function get_admin_header_ancore () {}
}
```

```

/**
 * Retourne le LIEN vers l'application de la section HEADER
 * _key -> titre
 * _link -> href
 * @return object
 */
public function get_header_login () {}

/**
 * Retourne le LOGO de l'application de la section HEADER
 * _key -> alt
 * _link -> src
 * @return object
 */
public function get_header_logo () {}

/**
 * Retourne le LIEN vers l'application de la section HEADER
 * _key -> slogan
 * @return object
 */
public function get_header_slogan () {}

/**
 * Retourne le HEADER de la section RECAP
 * _key -> titre
 * _value -> description
 * @return object
 */
public function get_recap_header () {}

/**
 * Retourne les CELLULES et le nombre de CELLULES de la section RECAP
 * _key -> titre
 * _value -> description
 * _link -> image
 * @return array(object, int)
 */
public function get_recap_cell () {}

public function get_admin_recap_cell () {}

/**
 * Retourne le HEADER de la section ABOUT
 * _key -> titre
 * _value -> description
 * @return object
 */
public function get_about_header () {}

/**
 * Retourne le LOGO et le LIEN de la section ABOUT
 * _key -> alt

```

```

* _value -> href
* _link -> src
* @return object
*/
public function get_about_logo () {}

/**
* Retourne le TITRE de la section PARTNER
* _key -> titre
* @return object
*/
public function get_partner_title () {}

/**
* Retourne le HEADER de la section PARTNER
* _key -> description
* @return object
*/
public function get_partner_header () {}

/**
* Retourne les CELLULES et le nombre de CELLULES de la section PARTNER
* _value -> href
* _link -> src
* @return object
*/
public function get_partner_cell () {}

public function get_admin_partner_cell () {}

/**
* Retourne le COPYRIGHT de la section FOOTER
* _key -> copyright
* _value -> name
* _link -> href
* @return object
*/
public function get_footer_copyright () {}

/**
* Retourne les MEDIAS de la section FOOTER
* _key -> alt
* _value -> href
* _link -> src
* @return object
*/
public function get_footer_media () {}

public function get_admin_footer_media () {}

}

```