

TP N°3 en PHP Objet Gestion des fiches de frais des visiteurs : Menu de navigation – UML et Modèle objet - Héritage - Développement des nouvelles classes – Modification CSS

Ce programme va reprendre les connaissances acquises au premier TP.

Nous allons ajouter une barre de navigation qui sera présente sur l'ensemble des pages, y compris sur la page d'authentification. Elle sera rajoutée collée au bas de l'image présente dans le header de chaque page, ce qui nécessitera quelque ajustement de notre feuille de style.

Nous réfléchirons ensemble aux nouvelles classes à mettre en place à l'aide d'un diagramme de classe. Les nouvelles classes seront en partie fournies par le professeur, ceci dans le but d'accélérer la cadence. Certaines classes ou partie de classes seront développées par vos soins, mais rassurez-vous une correction vous sera donnée avant de passer à la question suivante.

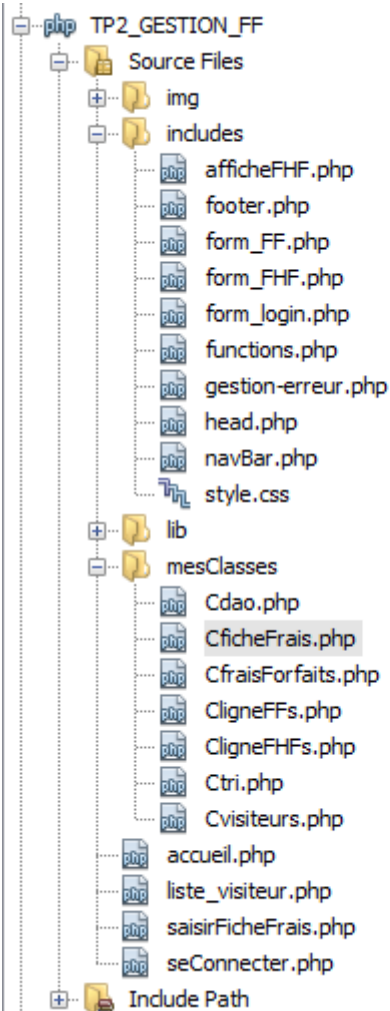
Table des matières

Présentation de l'architecture logicielle du programme	3
Tableau présentant l'architecture	3
Mettons en place la première partie de la page "saisirFicheFrais.php".	3
Voici à quoi va ressembler notre première partie de page.	4
Observons ensemble le fichier "saisirFicheFrais.php".	5
Travail à faire :	5
Mettons en place la partie gestion des frais hors forfait pour la page "saisirFicheFrais.php"	6
Le rendu à l'écran de ces deux fichiers est le suivant :	6
Le fichier "formFHF.php" :	7
Le fichier "afficheFHF.php" en deux parties pour des raisons de commodité	8
• La partie PHP	8
• La partie HTML	8
Travail à faire :	8

Mettons en place la partie gestion des frais forfaitaires pour la page "saisirFicheFrais.php"	9
Aperçu de la page "form_FF.php" qui est incluse dans la page "saisrFicheFrais.php"	9
Travail à faire :	9
Modification du modèle objet (ajout de nouvelles classes) afin de faire fonctionner la partie frontend.	10
Modélisation objet avec UML :	10
Diagramme de classe	10
Travail à faire :	11
Changeons un peu l'aspect des choses grâce aux CSS	14
Changeons l'aspect de la page de connexion.....	14
• Le nouveau rendu graphique.....	14
• Le code HTML de la page.....	14
• Le CSS de la DIV	15
• Le CSS de zone de texte H1	15
• Le CSS du formulaire.....	15

Présentation de l'architecture logicielle du programme

Tableau présentant l'architecture

 <p>Arborescence de fichiers du projet TP2_GESTION_FF :</p> <ul style="list-style-type: none">Source Files<ul style="list-style-type: none">imgincludes<ul style="list-style-type: none">afficheFHF.phpfooter.phpform_FF.phpform_FHF.phpform_login.phpfunctions.phpgestion-erreur.phphead.phpnavBar.phpstyle.csslibmesClasses<ul style="list-style-type: none">Cdao.phpCficheFrais.phpCfraisForfaits.phpCligneFFs.phpCligneFHF.phpCtri.phpCvisiteurs.phpaccueil.phpliste_visiteur.phpsaisirFicheFrais.phpseConnecter.phpInclude Path	<p>L'architecture logicielle de notre application reste la même mais de nouvelles pages ainsi que nouvelles classes apparaissent.</p> <p>En effet, le but de cette application est de gérer la saisie des fiches de frais des visiteurs médicaux de l'entreprise GSB, afin que ceux-ci puissent se faire rembourser les frais inhérents à leur activité professionnelle.</p> <p>Cette gestion des fiches de frais, va impliquer des changements tant au niveau de l'interface (front-end) qu'au niveau du modèle objet (back-end). Ainsi au niveau front-end, nous voyons apparaître la page "saisirFicherais.php" dans laquelle seront incluses (require_once) les pages afficheFHF.php, form_FF ou encore form_FHF.</p> <p>Par ailleurs, comme le programme gère deux types de frais dans une fiche de frais, les frais forfaitaires (FF) et les frais hors forfait (FHF), nous voyons apparaître de nouvelles classes métier, "CficheFrais" , CligneFF etc... Nous y réfléchirons un peu plus loin dans l'exercice.</p>
---	--

Mettons en place la première partie de la page "saisirFicheFrais.php".

Sur cette première partie de page, une barre de menu apparaît dans la page juste en dessous de l'image des tubes à essai. La navBar est accolée à l'image ce qui va nécessiter des ajustements au niveau de la feuille de style nommée "style.css".

Juste en dessous apparait la partie concernant les frais forfaitaires, mais pour des raisons de simplification nous traiterons cette partie plus tard dans l'exercice.

Voici à quoi va ressembler notre première partie de page.



Saisie des frais forfaitaires

Forfait étape:	<input type="text" value="0"/>	<input type="text" value="110.00"/>
Frais kilométrique:	<input type="text" value="0"/>	<input type="text" value="0.62"/>
Nuitée hôtel:	<input type="text" value="0"/>	<input type="text" value="80.00"/>
Repas restaurant:	<input type="text" value="0"/>	<input type="text" value="25.00"/>

Si nous jetons un œil dans la description de notre architecture logicielle dans la première partie de ce document nous avons deux fichiers à créer "saisirFicheFrais.php" et "navbar.php".

Observons ensemble le fichier "saisirFicheFrais.php".

```
<!DOCTYPE html>
<?php session_start(); ?>
<html>
<?php
    require_once 'includes/head.php';
    require_once 'mesClasses/Cvisiteurs.php';
    require_once 'mesClasses/CligneFFHf.php';
    require_once 'mesClasses/CligneFFh.php';
    require_once 'mesClasses/CficheFrais.php';
    require_once 'mesClasses/CfraisForfaits.php';

    // récupération de la référence de l'objet visiteur authentifié si la clef existe dans le tableau associatif $_SESSION
    $ovisiteur = null;
    if (key_exists('visiteur', $_SESSION)) {
        $ovisiteur = unserialize($_SESSION['visiteur']);
    }
    if ($ovisiteur == null) {
        header('location:seConnecter.php');
    }
    $oficheFrais = new CficheFrais();

    $oficheFrais->verifFicheFrais($ovisiteur->id); //verification de l'existence de la fiche de frais
    // récupération en GET de idLFHF ou btnFHF
    if (isset($_GET['idLFHF']) || isset($_POST['btnFHF'])) {
        $_SESSION['successMSG_FF'] = NULL;
    }

?>
<body>
<div class='container'>
<header title='saisirFF'></header>
<?php
    require_once 'includes/navBar.php';

?>

<?php
$formAction = $_SERVER['PHP_SELF'];
require_once 'includes/form_FF.php';
?>
<br>
<?php
require 'includes/gestion-erreur.php';

require_once 'includes/form_FHF.php';
require_once 'includes/afficheFHF.php';
?>

<br>
<br>
<?php
require 'includes/gestion-erreur.php';
?>
</div>
</body>
</html>
```

Ce fichier sera fourni par le professeur. Votre travail pour celui-ci sera de mettre en commentaire les lignes qui posons problèmes du fait de l'absence des fichiers correspondants.

Travail à faire :

1. Après avoir récupéré le fichier "saisirFicheFrais.php" le mettre en place dans le projet. Pour cela vous reprendrez le projet du TP N°1.
2. Après avoir écouté les consignes et les explications du professeur au vidéoprojecteur, mettre en commentaire les lignes du fichier pouvant poser problème du fait de l'absence du fichier ou de la classe correspondante.
3. Mettre en place la partie de la feuille de style permettant de gérer l'image des tubes à essai

```

56 header[title="saisirFF"], header[title="home"]{
57
58     height: 200px;
59     margin-top: 30px;
60     margin-bottom: 0px;
61     background-image: url("../img/med2.jpg");
62     /* positionne et montre le bas de l'image */
63     background-origin: content-box;
64     background-position: bottom;
65     background-repeat: no-repeat;
66     background-size: auto auto;
67
68 }

```

Mettre en commentaire la ligne 60.

4. Mettre en place le fichier « navbar.php » en vous inspirant de la page d'aide suivante https://www.w3schools.com/bootstrap/bootstrap_navbar.asp . Puis décommenter la ligne relative à ce fichier dans le fichier "saisrFicheFrais.php". Mais avant tout examinons les choses ensemble au vidéoprojecteur.
5. Modifiez la page "form_login.php" de manière à ce qu'en cas d'authentification réussie, l'utilisateur soit dirigé vers la page "saisirFicheFrais.php".
6. Observez le résultat. Faire le nécessaire afin que la barre de navigation soit collée à l'image du header.

Mettons en place la partie gestion des frais hors forfait pour la page "saisirFicheFrais.php"

Cette partie va nécessiter votre intervention au niveau frontend et au niveau backend pour développer les classes correspondantes.

La partie frontend doit être réalisée dans deux fichiers "formFHF.php" et "afficheFHF.php" qui sont ou seront inclus dans "saisirFicheFrais.php".

Le rendu à l'écran de ces deux fichiers est le suivant :

Saisie des frais hors forfait ✎

Libellé:

Montant:

Récapitulatif des frais hors forfait du mois : Août ☰

Libellé	Date	Montant	Action
Pas de frais hors forfait enregistrés			

Sur la capture ci-dessus nous pouvons remarquer deux parties principales :

- La saisie des frais hors forfait dans un formulaire (form_FHF.php)
- L'affichage des frais dans un tableau HTML (afficheFHF.php) avec un message d'avertissement quand il n'y a pas de frais à afficher – ce qui est le cas ici.

Ces deux parties vont impliquer des modifications dans la partie frontend (formFHF.php et afficheFHF.php) mais aussi backend avec l'ajout des classes relatives à la gestion des frais hors forfait, CligneFHF et CligneFHF.s respectivement la classe métier et la classe de contrôle des objets de type CligneFHF.

Occupons-nous du frontend dans un premier temps.

Le fichier "formFHF.php" :

```
<?php
2 $errorMsg = NULL;
3 $successMsg = NULL;
4 if(isset($_POST['btnFHF']))
5 {
6     if(isset($_POST['libelle']) && isset($_POST['montant']))
7     {
8         $oLigneFHF = new CligneFHF;
9         try {
10             $oLigneFHF->insertFHF($_POST['libelle'], $_POST['montant']);
11         } catch (Exception $ex) {
12             $errorMsg = "Erreur lors de l'insertion dans la base.".$ex->getMessage()." Prévenir l'administrateur.";
13         }
14     }
15 }
16 ?>
17 <div class="container">
18     <h3><p class="text-primary page-header">Saisie des frais hors forfait <span class="text-primary glyphicon glyphicon-pencil"></p></h3>
19     <br>
20     <form id="formFHF" class="form-horizontal" role="form" method="post" action="<?=$formAction?>">
21         <div class="form-group">
22             <label class="control-label col-sm-2" for="libelle">Libellé:</label>
23             <div class="col-sm-10">
24                 <textarea type="text" class="form-control" name="libelle" placeholder="Entrer libelle frais" required autofocus></textarea>
25             </div>
26         </div>
27         <div class="form-group">
28             <label class="control-label col-sm-2" for="montant">Montant:</label>
29             <div class="col-sm-10">
30                 <input class="form-control" name="montant" placeholder="Entrer montant frais" required="required" type="number" min="0" step="0.01">
31             </div>
32         </div>
33         <div class="form-group">
34             <div class="col-sm-offset-2 col-sm-10">
35                 <button type="submit" name="btnFHF" class="btn btn-default">Enregistrer</button>
36             </div>
37         </div>
38     </form>
39 </div>
```

Dans cette page deux parties principales que le professeur va vous expliquer au vidéoprojecteur :

- La partie HTML qui est un formulaire géré avec des CSS Bootstrap
- La partie PHP de traitement des données envoyées par le formulaire

Le fichier "afficheFHF.php" en deux parties pour des raisons de commodité :

- La partie PHP

```
1 <?php
2
3 $oLigneFHF = new CligneFraisFHF();
4
5 if(isset($_GET['idLFHF']))
6 {
7     try {
8
9         $oLigneFHF->deleteFHF($_GET['idLFHF']);
10        /* Adapte l'url en fonction de vos besoins.
11        */
12        echo '<meta http-equiv="refresh" content="0"
13            .':http://qsbci4.local/saisirFicheFrais.php">'; // 0 est le temps de rafraichissement
14        //header('location:saisirFicheFrais.php');
15    } catch (Exception $ex) {
16
17        $errorMsg = "La ligne n° " . $_GET['idLFHF'] . " n'a pas été correctement supprimée.";
18    }
19 }
20
21
22
23
24
25 ?>
```

- La partie HTML

```
26 <div class="container">
27 <h1><p class="text-primary">Récapitulatif des frais hors forfait du mois :<?=' ' . moisEnFrancais(date('F')) . ' ' .><span class="glyphicon glyphicon-align-justify"></p></h1>
28
29 <table class="table table-hover">
30 <thead>
31 <tr class="bg-info">
32 <th>Libellé</th>
33 <th>Date</th>
34 <th>Montant</th>
35 <th>Action</th> <!-- supprimer ligne -->
36 </tr>
37 </thead>
38 <tbody>
39 <?php
40
41 $mois = getAnneeMois();
42 $visiteur = unserialize($_SESSION['visiteur']);
43 $oCollLigneFHFByVisiteur = $oLigneFHF->getFHFByIdVisiteurMois($visiteur->id, $mois);
44 if($oCollLigneFHFByVisiteur != null)
45 {
46     foreach($oCollLigneFHFByVisiteur as $LigneFHF)
47     {
48         <tr>
49             <td><?=$LigneFHF->libelle?></td>
50             <td><?=$LigneFHF->date?></td>
51             <td><?=$LigneFHF->montant >= 100?>class="text-danger":'";'><?=$LigneFHF->montant?></td>
52             <td><a href="saisirFicheFrais.php?idLFHF='<?=$LigneFHF->id?>'> class="btn btn-danger" id="btnSuppLigneFHF" role="button">Supprimer</a></td>
53         </tr>
54     }
55 }
56 else
57 {
58     if(!isset($errorMsg)) // sinon d'autre message prioritaire
59     {
60         $errorMsg = "Pas de frais hors forfait enregistrés ";
61     }
62 }
63
64 ?>
65
66 </tbody>
67 </table>
```

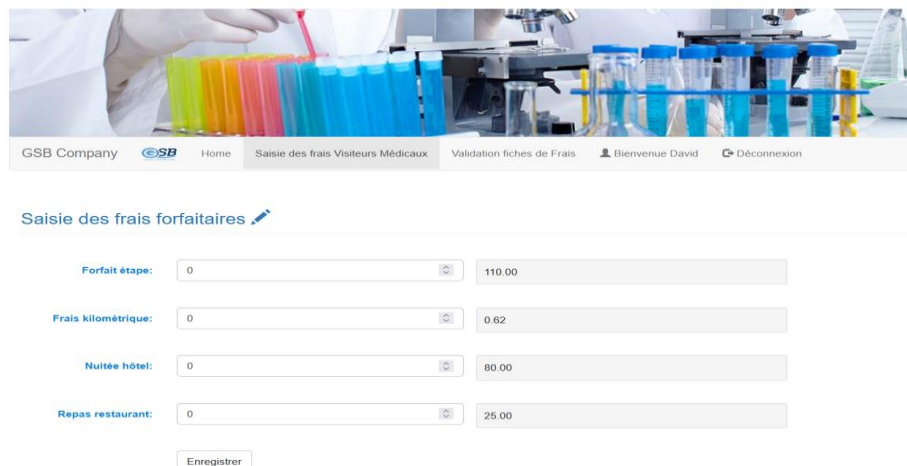
Travail à faire :

1. Implémentez ces deux fichiers à l'endroit indiqué dans l'architecture fournie au début de ce document.

Attention : Ces deux fichiers ne peuvent pas pour l'instant fonctionner correctement car nous n'avons pas encore mis en place les classes relatives aux lignes de frais, c'est ce que je vous propose de faire au prochain paragraphe.

Mettons en place la partie gestion des frais forfaitaires pour la page "saisirFicheFrais.php"

Aperçu de la page "form_FF.php" qui est incluse dans la page "saisrFicheFrais.php"



The screenshot shows a web application interface for 'GSB Company'. The header includes a logo, 'Home' link, and navigation tabs: 'Saisie des frais Visiteurs Médicaux', 'Validation fiches de Frais', 'Bienvenue David', and 'Déconnexion'. The main section is titled 'Saisie des frais forfaitaires' with a pencil icon. It contains four rows of input fields, each with a label, a numeric input, and a read-only display field:

Label	Input Value	Display Value
Forfait étape:	0	110.00
Frais kilométrique:	0	0.62
Nuitée hôtel:	0	80.00
Repas restaurant:	0	25.00

At the bottom of the form is an 'Enregistrer' button.

Il s'agit de la partie située juste en dessous du header et de la navBar que nous avons déjà mis en place plus avant de ce document.

La logique est la même que pour les frais hors forfait. Nous avons dans cette page deux parties :

- Une partie HTML qui est composée de balise « label » et « input » lesquelles sont encadrées par des balises « form » permettant la mise en place d'un formulaire pour l'envoi des données au traitement
- Une partie PHP qui va traiter les données envoyées par le formulaire.

Travail à faire :

1. En vous inspirant de ce qui a été fait pour les frais hors forfait (et des explications fournies à ce sujet) et en vous aidant de la capture ci-dessus – mettre en place la page "form_FF.php".

Modification du modèle objet (ajout de nouvelles classes) afin de faire fonctionner la partie frontend.

Dans ce paragraphe, il s'agit de réfléchir à notre modèle objet, c'est-à-dire à l'ensemble des classes qui vont permettre de nous aider dans la gestion des lignes de frais hors forfait et forfaitaire composant notre fiche de frais. Commençons d'abord par les lignes de frais hors forfait.

Modélisation objet avec UML :

Diagramme de classe

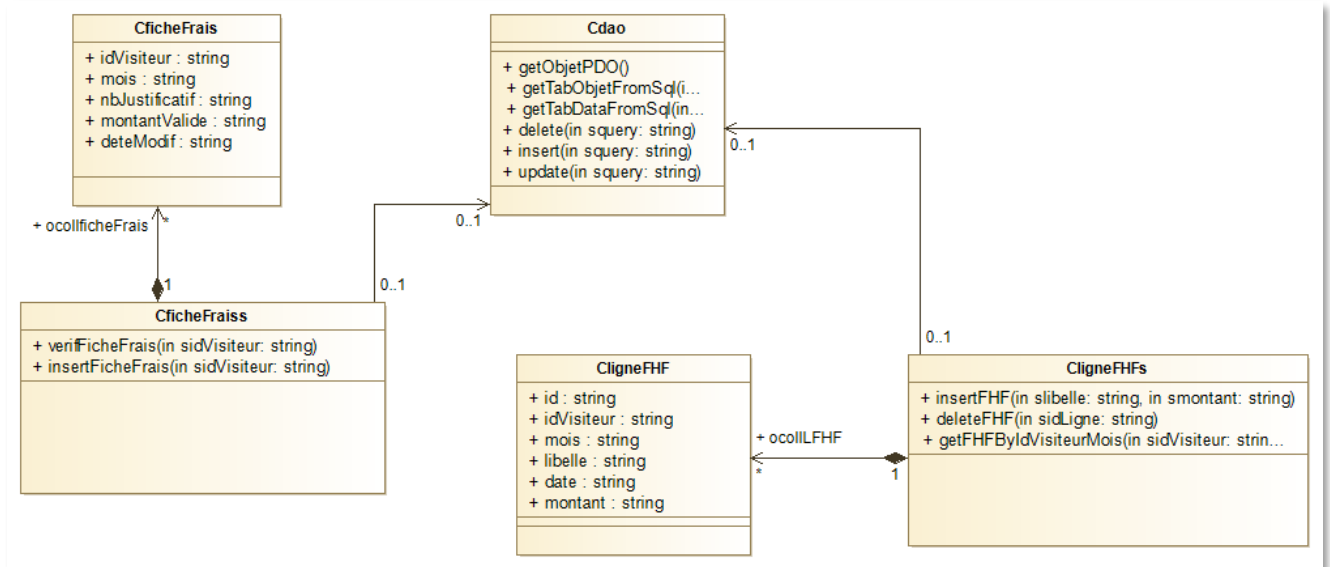
A partir des diagrammes préalablement expliqué je vais construire la réflexion qui me permettra de développer mes classes.

- *Convention de nommage :*
 - Le nom des classes commence par "C"
 - Les classes de contrôle, autrement dit celles qui contrôlent les objets métier, prennent un "s" à la fin de leur nom, cela pour préciser **la pluralité** dans la gestion des objets.
- *Départ de la réflexion concernant la modélisation :*
 - Que gère-t-on ici ? Des fiches de frais composées de ligne de frais hors forfait. Comme pour le premier TP qui gérait l'authentification des visiteurs j'ai besoin à chaque fois d'une classe métier et de la classe de contrôle correspondante. J'ai besoin des classes suivantes :
 - CficheFrais et CficheFrais
 - CligneFHF et CligneFHF
 - Quels seront les attributs de mes classes ? Pour les classes métier je mettrai les attributs en fonction des champs des tables correspondantes de la base de données. Pour les classes de contrôle ce sera en fonction des besoins.
 - Quelles seront les méthodes de mes classes ? Sauf exception les classes métier n'auront pas de méthode. Les classes de contrôles auront des méthodes en fonction des besoins.

Si nous regardons attentivement le code de la page "saisirFicheFrais.php" nous nous apercevons qu'il y a une méthode qui vérifie l'existence de la fiche de frais pour le mois en cours, si celle-ci n'existe pas il faudra la créer. De cela découle qu'il faudra développer deux méthodes une méthode permettant la vérification et une autre qui permettra la création de la fiche si celle-ci n'existe pas. Nous les nommerons respectivement "verifFicheFrais()" et l'autre "insertFicheFrais()". Nous nous occuperons

plus tard de leur code, pour l'instant occupons nous de représenter ces classes dans un diagramme de classe.

- *Représentation du diagramme de classe*



Travail à faire :

Les fichiers demandés dans les questions suivantes sont à mettre dans les répertoires indiqués au début de ce document.

1. Implémenter les classes relatives aux fiches de frais de manière à faire fonctionner la page "saisirFicheFrais.php" au niveau de l'utilisation de la méthode "verifFicheFrais()" qui va vérifier l'existence de la fiche pour le mois et la créer si elle n'existe pas. Pour cette question je vous fournis le code que nous allons étudier ensemble au vidéoprojecteur avant que vous ne l'implémentiez vous-même.

Attention : cela ne doit pas être une simple recopie ni une course contre la montre – Il faut comprendre au moment de l'explication donnée par le professeur mais aussi et surtout au moment où vous notez le code, il faut se poser des questions.

Le fichier `CficheFrais.php` qui contient l'ensemble des classes relatives aux fiches de frais.

- La classe métier "CficheFrais"

```

1 <?php
2 require_once 'Cdao.php';
3
4 class CficheFrais
5 {
6     public $idVisiteur;
7     public $mois;
8     public $nbJustificatifs;
9     public $montantValide;
10    public $dateModif;
11    public $idEtat;
12 }
  
```

Remarque sur la capture ci-dessus :

- Je vais avoir besoin de la classe "Cdao" et d'une nouvelle méthode (getTabObjetFromSql) qui va me renvoyer un tableau d'objets de type "CficheFrais".

Pour l'instant ne nous en préoccupons, c'est l'objet de la prochaine question.

- On peut remarquer aussi que **je n'utilise pas de constructeur** dans la classe "CficheFrais", cela est dû au fait que la méthode que je vais implémenter dans "Cdao" à la prochaine question ("getTabObjetFromSql") utilise l'instruction " \$result = \$sth->fetchAll(PDO::FETCH_CLASS, \$stype); "qui n'accepte pas de constructeur pour le type qui lui est transmis (ici dans la variable chaîne \$stype).

- La classe de contrôle des objets métier en deux parties par commodité

```
13
14 class CficheFrais
15 {
16     public $ocollFicheFrais;
17
18     public function __construct()
19     {
20         $odao = new Cdao();
21
22         $query = 'SELECT * FROM fichefrais';
23
24         $stabObjetFF = $odao->getTabObjetFromSql($query, 'CficheFrais');
25
26         $this->ocollFicheFrais = array();
27
28         foreach($stabObjetFF as $oficheFrais)
29         {
30             $this->ocollFicheFrais[] = $oficheFrais;
31         }
32         unset($odao);
33     }
34 }
```

```
35 public function verifFicheFrais($sidVisiteur)
36 {
37     $oFicheFraisByIdVisiteur = null;
38     foreach($this->ocollFicheFrais as $oficheFrais)
39     {
40         if($oficheFrais->idVisiteur == $sidVisiteur && $oficheFrais->mois == getAnneeMois())
41         {
42             $oFicheFraisByIdVisiteur = $oficheFrais;
43         }
44     }
45     if($oFicheFraisByIdVisiteur == null)
46     {
47         $this->insertFicheFrais($sidVisiteur);
48     }
49 }
50
51 public function insertFicheFrais($sidVisiteur)
52 {
53     $odao = new Cdao();
54     $mois = getAnneeMois();
55     $query = "INSERT INTO fichefrais (idVisiteur, mois, idEtat) VALUES ('" . $sidVisiteur . "', '" . $mois . "', 'CR')";
56
57     $odao->insertion($query);
58     unset($odao);
59 }
60
61 }
```

2. Implémenter la méthode "getTabObjetFromSql" dans la classe "Cdao". Il s'agit d'une méthode à laquelle je fournis deux paramètres (requête SQL et type des objets) et qui est capable de retourner un tableau d'objet du type indiqué, ici "CficheFrais".

```

28     }
29
30     public function getTabObjetFromSql($query, $type) // $type sera le type d'objet que l'on désire
31     {
32         $lepdo = $this->getPDO();
33
34         $sth = $lepdo->prepare($query); //prepare echappe entre autre proprement la requete
35         $sth->execute();
36
37         $result = $sth->fetchAll(PDO::FETCH_CLASS, $type); // fetchall met toutes les données dans un tableau d'objets du type en paramètre
38         unset($lepdo);
39         return $result; // $result est un tableau d'objet du type en parametre comportant tous les objets du type (All de fetchAll)
40
41     }
42

```

- Implémentons ensemble les classes relatives aux lignes de frais hors forfait, "CligneFHF" et "CligneFHF" en nous inspirant du code donné ci-avant pour les classes relatives aux fiches de frais et du diagramme de classe.

Ces classes permettront de faire fonctionner les pages frontend form_FHF.php et affiche_FHF.php que nous avons déjà mis en place.

- Implémentons ensemble les classes relatives aux lignes de frais forfaitaire, "CligneFF" et "CligneFF" en nous inspirant du code donné ci-avant pour les classes relatives aux fiches de frais. Modifier ensuite le diagramme de classe afin d'ajouter ces classes.

Ces classes permettront de faire fonctionner la page frontend form_FF.php.

- Nous voudrions maintenant prévoir que d'autres employés que les visiteurs médicaux utilisent l'application. Pour ce faire nous allons créer une classe "Cemploye" dont héritera la classe "Cvisiteur". Le code est le suivant :

```

8  class Cemploye{
9
10     public $id;
11     public $login;
12     public $mdp;
13     public $nom;
14     public $prenom;
15     public $connecte;
16
17     function __construct($sid,$slogin,$smdp,$snom,$sprenom){
18         $this->id = $sid;
19         $this->login = $slogin;
20         $this->mdp = $smdp;
21         $this->nom = $snom;
22         $this->prenom = $sprenom;
23         $this->connecte = false; // le visiteur est par défaut non connecté*/
24     }
25 }
26
27
28 class Cvisiteur extends Cemploye
29 {
30
31     // champ qui n'est pas dans la base mais uniquement au niveau objet
32
33
34     // Penser à commenter le constructeur pour le cas n°3 du constructeur de Cvisiteurs
35     function __construct($sid,$slogin,$smdp,$snom,$sprenom)
36     {
37         parent::__construct($sid,$slogin,$smdp,$snom,$sprenom);
38
39         /*$this->id = $sid;
40         $this->login = $slogin;
41         $this->mdp = $smdp;
42         $this->nom = $snom;
43         $this->prenom = $sprenom;
44         $this->connecte = false; // le visiteur est par défaut non connecté*/
45     }
46 }
47

```

- Observons et comprenons ensemble ce code – prenez les notes nécessaires. Implémentez-le.
- Que signifie la remarque dans la classe "Cvisiteur" concernant le cas n°3 ?
- Modifiez le diagramme de classe afin de mettre en œuvre cet héritage en UML.

Changeons un peu l'aspect des choses grâce aux CSS

Changeons l'aspect de la page de connexion

- Le nouveau rendu graphique



Cette page de connexion comporte une DIV avec une image de fond. Dans cette div sont incorporés un titre H1 avec une police de caractère "Google font" et un formulaire.

Le secret ? un positionnement relatif (relative) pour la DIV englobante (class « bg-img ») et un positionnement absolu (absolute) pour le H1 (title « cnx ») et la DIV du Formulaire (class « containerForm »). Cela produit une sorte de changement d'échelle, la DIV englobante devient alors le nouveau point de référence du H1 et de la DIV contenant le formulaire, il suffit alors de les placer à l'intérieur avec les attributs « left » ou « right ».

- Le code HTML de la page

Je ne donne pas le code PHP se trouvant dans la même page puisqu'il n'intervient pas au niveau des CSS.

```

25
26 <div class="bg-img">
27
28
29     <h1 title="cnx">Connexion lab GSB</h1>
30
31     <div class="containerForm">
32         <form action="" method="post">
33
34             <label for="username"><b>Login</b></label>
35             <input type="text" id="username" placeholder="Entrez un login" name="username" required="">
36
37             <label for="pwd"><b>Password</b></label>
38             <input type="password" id="pwd" placeholder="Entrez un mot de passe" name="pwd" required="">
39             <input type="submit" class="btn"/>
40         </form>
41     </div>
42
43 </div>
44

```

- Le CSS de la DIV

```

24 .bg-img { /* class de la DIV englobant la DIV avant le formulaire */
25     /* lien vers l'image */
26     background-image: url("../img/med2.jpg");
27     margin-top: 30px; /**** ajouté pour login avec formulaire dans image */
28     min-height: 380px;
29     /*width: 100%*/
30     /* Positionnement et echelle */
31     background-position: center;
32     background-repeat: no-repeat;
33     background-size: cover; /* contain fait apparaître toute l'image mais moins grande que la div */
34     position: relative; /*relative, absolute, fixed */ /*Relative permet de changer l'origine de ce qui sera en absolute à l'intérieur */
35     padding: 10px;
36 }

```

- Le CSS de zone de texte H1

```

49 h1[title="cnx"]{
50     color: #ee718b; /*Couleur Rosée pour rappel tube à essai*/
51     font-family: 'Dancing Script', cursive; /* google font */
52     font-size: 4em ;
53     position: absolute;
54     left: 5px;
55 }

```

- Le CSS du formulaire

```

39 .containerForm {
40     position: absolute; /* top bottom left right se font par rapport à l'élément englobant qui lui est forcément relative*/
41     right: 0; /* Positionnement à droite essayer avec left */
42     margin: 20px; /* marge de la DIV containerForm par rapport à son parent DIV bg-img*/
43     height: 90%;
44     max-width: 300px;
45     padding: 35px; /* marge pour les éléments internes cad les zones du formulaire */
46     background-color: white;
47 }

```