TP 6 : Site monopage avec modèle

Objectifs du TP:

 Un site Web mono page avec un formulaire. Programmation d'un modèle à partir de la modélisation UML.

1. Conversion de devises

On souhaite réaliser un mini site Web de conversion de devises l'aide d'une seule page en PHP.

Une devise est une unité monétaire d'un pays ou d'une zone économique. Dans cet exercice une devise est identifiée par 3 lettres majuscules comme 'EUR' pour les euros. Le taux de conversion entre deux devises est le coefficient multiplicateur qu'il faut appliquer au montant dans la devise source pour obtenir le montant dans la devise cible.

- EUR : Euro
- Soit les devises suivantes :

- USD : Dollar des Etats-Unis GBP : Livre britannique JPY : Yen japonais · CAD: Dollar canadien
- avec des taux de conversion entre ces devises . Exemple (avec notation anglophone des nombres) :

Taux De Vers

```
EUR
      USD
            1.11069
      GBP
            0.860733
EUR
GBP
      CAD
            1.69413
            0.0120627
JPY
      CAD
```

Vous avez un squelette de site avec les fichiers et les répertoires suivants : data : les données de devises et du taux de change dans des fichiers.

helper : des fonctions générales pour aider au developpement (pour la version 2 uniquement). index.php : la version de base du site.

- index2.php: la version 2.

• model : le code php (sans HTML) du modèle de l'application.

- test: scripts et données pour réaliser des tests hors serveur.
- A faire:

Outil de convertion de devises

Boursomachin met à votre disposition ce convertisseur de devises, qui vous permet de convertir des monnaies instantanément et gratuitement. Vous pouvez convertir entre elles les devises les plus populaires comme, Euro

```
Usage : vous entrez dans le convertisseur le montant que vous souhaitez convertir, indiquez la devise d'origine et
 la devise qui vous intéresse. Et vous obtenez instantanément le montant dans la devise souhaitée, avec le taux de
 change entre les 2 monnaies.
 Convertisseur
                                                                                Convertir
                              EUR (Euro)
                                                              USD (Dollar)
         Montant
Le bouton 'Convertir' lance la conversion et affiche le résultat, sous le formulaire :
```

la devise qui vous intéresse. Et vous obtenez instantanément le montant dans la devise souhaitée, avec le taux de change entre les 2 monnaies.

Boursomachin met à votre disposition ce convertisseur de devises, qui vous permet de convertir des monnaies instantanément et gratuitement. Vous pouvez convertir entre elles les devises les plus populaires comme, Euro

Usage : vous entrez dans le convertisseur le montant que vous souhaitez convertir, indiquez la devise d'origine et

5.55 USD (1.11069)

EUR, Dollar US USD, Yen japonais JPY, Livre Sterling GBP.

Outil de convertion de devises

Convertisseur Convertir EUR (Euro) USD (Dollar)

5

index.php). Les données de conversion sont dans le fichier data/exchangeRate.csv. Ce fichier est au format CSV. Ce format texte contient une première ligne qui indique le noms des champs, puis chaque ligne contient une donnée structurée correspondant aux champs. Le caractère séparateur est par exemple la virgule, Par exemple le fichier data/exchangeRate.csv contient :

EUR, USD, 1.11069 EUR, GBP, 0.860733

leut clée.

A faire:

\$s = 'EUR';

\$c = 'USD';

// Calcul de la clée

\$taux[\$clee] = 1.11069;

["EUR USD"]=> float(1.11069)

\$clee = \$s.' '.\$c;

var_dump(\$taux);

array(1) {

\$s = 'EUR';

c = 'USD';

1.2 Analyse Important : dans cette version nous n'utilisons que le fichier data/exchangeRate.csv, donc pas le fichier data/devise.csv qui sera utilisé dans la version 2.

Nous proposons de faire une analyse en Orienté Objet de cet exercice. Ce qui importe est la notion de change d'une monnaie à une autre. Cette notion représente en fait toute l'application. Il y a

aussi la notion de Devise, de Taux, mais nous ne les modéliserons pas par des objets dans cette version. Nous réalisons volontairement une analyse objet incomplète pour limiter le codage. La

function change(string \$from, string \$to, float \$amount) : float

Pour réaliser le calcul du change, la méthode change() a besoin de connaître le taux de change entre deux devises. Nous proposons donc la méthode :

Change

+load()

// Charge la liste des Taux et des idf de devises private function load(string \$filename)

Le taux de change est dans un fichier, donc il faut prévoir une méthode privée, qui charges ces informations dans l'objet change. Ce chargement sera réalisé à la construction de l'objet.

```
Nous devons afficher dans l'interface la liste des devises disponibles. Dans cette version, nous n'afficherons que l'identifiant de 3 lettres. Ces identifiants apparaissent dans le fichier
data/exchangeRate.csv. Il faut donc que la fonction load() conserve dans un attribut de l'objets, tous les identifiants de devise qu'il rencontre.
Un taux de change dans notre modèle, n'est qu'un nombre flottant. Ce nombre est associé à une clée unique qui est composée des 3 lettres de la devise source et des trois lettres de la devise
cible. Nous proposons d'ajouter un espace entre les deux pour faciliter le debug. La fonction load() est donc chargée de parcourir le fichier en entier, et de conserver tous les taux associées à
```

Une modelisation ULM est proposée ci-dessous. Attention les taux de conversions et les devises apparaissent comme des classes, mais ne doivent pas être traduites en classes dans le code de cette première version. La seconde version (en bonus), vous propose de créer la classe des devises.

+getRate() +getDevises() +src () +dest +change()

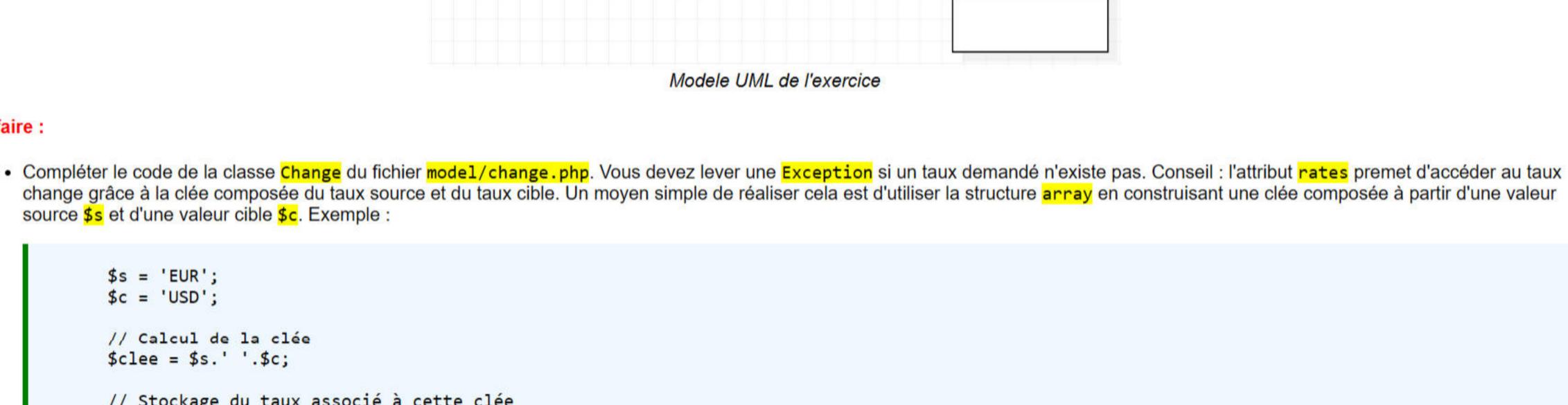
Rate

Devise

+value

+idf

+name



\$s = 'EUR'; \$c = 'USD';// Stockage du taux associé ces deux valeurs

La récupération des données se fait par un accès direct de la même manière :

print("Le taux de \$s vers \$c est de : ",\$taux[\$s.' '.\$c);

```
Pour savoir si le taux existe on peut utiliser isset(). L'espace entre la cible et la source n'est pas nécéssaire, mais permet de vérifier plus facilement le contenu avec var_dump(). Une
autre solution consiste à faire un tableau de tableau :
         taux[s][c] = 1.11069;
         var_dump($taux);
         array(1) {
           ["EUR"]=> array(1) {
                ["USD"]=> float(1.11069)
```

["GBP CAD"]=> string(7) "1.69413" ["JPY CAD"]=>

```
[0]=>
                string(3) "EUR"
                 [1]=>
                 string(3) "USD"
                 [2]=>
                 string(3) "GBP"
                 [3]=>
                string(3) "JPY"
                 [4]=>
                string(3) "CAD"
            OK: taux de change EUR -> EUR = 1.000000
            OK: taux de change EUR -> GBP = 0.860733
            OK: taux de change GBP -> EUR = 1.161800
            OK: change inconnu bien détecté. Message : ERREUR : taux de EUR vers XXX inconnu
            Le change de 150.00 EUR en USD fait 166.60
   • Compléter le fichier index.php qui contient l'affichage (la vue) en PHP. Ce fichier doit utiliser le modèle. Trouver une solution simple pour que lorsque qu'une valeur de change est calculée,
    l'affichage du formulaire soit conservée. Pour cela regarder <u>l'attribut selected de la balise option.</u>
1.3 Poursuite de l'analyse (version 2) Bonus
Dans cette partie nous poursuivons l'analyse objet, et décidont de modéliser également les devises. Une devise possède un identifiant de 3 lettres et un nom plus long. C'est ce nom qui doit être
affiché à l'utilisateur. Le fichier data/devise.csv contient toutes les devises dont on connaît les taux de change.
On remarque que ce fichier est également au format CSV. Il serait donc utile de factoriser le code de lecture de n'importe quel fichier CSV dans une fonction à part à titre d'exercice, et pour éviter
de rallonger le code par du "couper/coller" qu'il faut toujours éviter.

    Compléter dans le fichier helper/readCSV.php le code de la fonction :

                 function readCSV(string $filename, string $delimiter=',') : array
```

```
OK: fichier avec une ligne trop courte bien détecté. Message : Error line 4 has only 2 fields, 3 expected in 'd,e'
#Warning line 5 has 4 fields, only 3 expected, extra is ignored in 'f,g,h,i'
array(5) {
  [0]=>
 array(3) {
```

```
string(1) "5"
  ["champ3"]=>
 string(1) "6"
[3]=>
array(3) {
  ["champs1"]=>
  string(1) "f"
  ["champs2"]=>
```

```
string(1) "k"
    ["champ3"]=>
    string(1) "l"
array(4) {
  [0]=>
  array(2) {
    ["idf"]=>
    string(3) "EUR"
    ["name"]=>
    string(4) "Euro"
```

```
["name"]=>
  string(20) "GBP (Livre Sterling)"
[3]=>
array(2) {
  ["idf"]=>
  string(3) "JPY"
```

 Compléter le fichier model/changeV2.php. Il est très proche du fichier model/change.php. Les différences sont les suivantes : Il faut lire 2 fichiers CSV lors de la construction, et utiliser la fonction readCSV pour cela. Il faut sauvegarder non pas uniquement les identifiants des devises, mais les objets devises, avec un acces par leur clée qui est l'IDF.

Sújet de TP Programmation vv⊏B (PHF).

Sujet de TP Programmation WEB (PHP)

Exemples de conversions : la conversion de 23,00€ (EUR) en USD s'obtient avec 23,00€ * 1,11069 = 25,55 USD arrondi à deux chiffres. la conversion de 30,00 Livres britannique (GBP) en Euro (EUR) s'obtient avec 30,00£ / 0,860733 = 34,85€.

design : le styles et images du site.

Récupérer les données dans : <u>data/devise.zip</u>

1.1 Spécification Le site est un formulaire contenant une zone d'entrée pour indiquer le montant à convertir, deux sélecteurs qui indiquent la devise source et la devise cible.

EUR, Dollar US USD, Yen japonais JPY, Livre Sterling GBP.

Le site Web n'est composé que d'une seule page (

from, to, rate Cela signifie que pour convertir des Euros en US Dollars il faut appliquer le taux de 1.11069.

version 2 décrite plus bas, ira un peu plus loin dans l'analyse objet. Une devise n'est donc pas modélisée et est représentée simplement par une chaine de 3 caractères qui est son identifiant. De même, le taux est simplement un flottant. Nous proposons donc de modéliser cette notion de change par une classe sigleton appelée Change. C'est à dire qu'elle ne va produire d'une unique instance d'objet que l'on appèle simplement change. Ce que cet objet change doit savoir faire c'est changer un montant d'une devise en une autre. Nous le modélisations par la méthode :

// Calcul du taux entre deux IDF de devises function getRate(string \$from, string \$to) : float

On remarque que le fichier ne contient pas tous les taux. Il n'y a pas le taux inverse, ni le taux de 1.0 quand les devises sont les mêmes. Le taux inverse se calcule par rapport au taux donné dans le fichier. Par exemple si le taux de "EUR USD" est 1.11069, alors le taux pour "USD EUR" est 1/1.11069. Ce calcul peut être fait au chargement ou (de préférence) à la demande dans la méthode getRate().

source \$s et d'une valeur cible \$c. Exemple : // Stockage du taux associé à cette clée

• Tester ce code en ligne de commande avec test/test_change.php. Vous devez obtenir le résultat suivant : object(Change)#1 (2) { ["rates": "Change": private] => array(10) { ["EUR USD"]=> string(7) "1.11069" ["EUR GBP"]=> string(8) "0.860733" ["EUR JPY"]=>

string(7) "120.884"

A faire:

["EUR CAD"]=> string(7) "1.45792" ["USD GBP"]=> string(8) "0.775347" ["USD JPY"]=> string(7) "108.878" ["USD CAD"]=> string(7) "1.31327" ["GBP JPY"]=> string(7) "140.442" string(9) "0.0120627" ["devises": "Change": private]=> array(5) {

Cette fonction ouvre un fichier CSV et le lit intégralement. Elle produit en sortie un tableau. Chaque élément de ce tableau est lui même un tableau qui contient les valeurs de la ligne indicées par le nom du champs (voir exemple ci-dessous). Cette fonction doit envoyer une exception (différente) dans les situations d'erreur suivantes : Le fichier ne peut pas être ouvert. Le fichier est vide. Le fichier ne contient qu'une seule ligne. o Le fichier possède une ligne avec moins de valeurs qu'attendu dans la liste des champs de la première ligne.

OK: fichier à une ligne bien détecté. Message : Error file 'oneLine.csv' has only one line 'champs1, champs2, champ3'

["champs1"]=> string(1) "1" ["champs2"]=> string(1) "2" ["champ3"]=>

string(1) "3"

["champs1"]=>

string(1) "a"

["champs2"]=>

string(1) "b"

["champ3"]=>

string(1) "g"

["champ3"]=>

[1]=>

array(2) {

["name"]=>

string(9) "JPY (Yen)"

[1]=>

array(3) {

Par contre, cette fonction ne met qu'un message d'alerte si une ligne possède plus de valeurs qu'attendu.

OK: fichier vide bien détecté. Message : Error file 'empty.csv' is empty

OK: fichier inconnu bien détecté. Message : Error file 'doNotExists.csv' not found

Tester cette fonction avec le fichier test/test readCSV.php. Vous devez avoir la sortie suivante :

string(1) "c" [2]=> array(3) { ["champs1"]=> string(1) "4" ["champs2"]=>

string(1) "h" [4]=> array(3) { ["champs1"]=> string(1) "j" ["champs2"]=>

["idf"]=> string(3) "USD" ["name"]=> string(12) "USD (Dollar)" [2]=> array(2) { ["idf"]=> string(3) "GBP"

• Compléter le fichier indexV2.php. Il est très proche du fichier index.php, la seule différence est qu'il affiche le nom de la devise à la place de son identifiant, dans les listes déroulantes.