

CONTRÔLE DES CONNAISSANCES N° 3
DURÉE : 1h

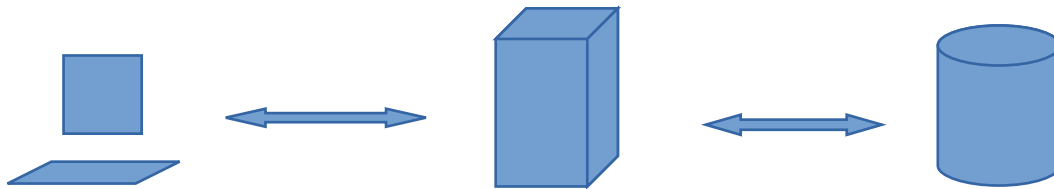
Vendredi 12 novembre 2021

NOM/Prénom : SOW Awa / Enzo Carpentier

PARTIE 1 - MVC Web - 40 points

Q1. Architecture - (4 points)

Q1 A partir de l'architecture 3 tiers Web, déterminer quels termes techniques proposés dans la liste ci-dessous font référence à une technologie active sur plus d'un tiers. Justifier votre réponse. (1 point si non argumenté)



Client Web Navigateur

Http, JavaScript

serveur HTTP

Http, JavaScript (NodeJS)

SGBDR

Liste de termes : JavaScript, HTML, HTTP, SQL, PHP, Flash, Twig, Symfony

- **http** : il est actif sur le navigateur qui interagit avec le serveur (lance des requêtes HTTP)
- **javaScript** : il est actif sur le navigateur mais peut l'être sur le serveur (nodejs)

Q2 Voici un code extrait d'un projet VETUX-LINE de l'équipe *kiproko*.

// dans un contrôleur

```
public function read1()
{
    $fichier = 'uploads/french-data.csv';
    $file = fopen($fichier, 'r');
    while($tab1[] = fgetcsv($file, 2000, ",")) ;
    fclose($file);
    return $tab1;
}

public function read2()
{
    $fichier2 = 'uploads/german-data.csv';
    $file2 = fopen($fichier2, 'r');
    while($tab2[] = fgetcsv($file2, 2000, ","));
    fclose($file2);
    return $tab2;
}
```

Q2-a Identifier 3 caractéristiques de ce code susceptibles d'être améliorées. (3 points)

- on peut améliorer le nommage car il est très pauvre
- la boucle « while » n'a pas de cœur (la visibilité)
- Français ou Anglais faut choisir « de préférence en anglais »
- la redondance de code
- la variable \$tab[] n'est pas déclaré avant

Q2-b Ces deux fonctions se ressemblent trop. Proposer une solution qui éviterait la redondance de code. (4 points)

```
// API
public function fromCSVfileToArray($nomFichier) {
    $fichier = 'uploads' + $nomFichier;
    $file = fopen($fichier, 'r');
    while($tab[] = fgetcsv($file, 2000, ","));
    fclose($file);
    return $tab;
}

// Function insérer fichier csv
public function read1{ return fromCSVfileToArray('french-data.csv') }
public function read2{ return fromCSVfileToArray('german-data.csv') }

// Utilisation
$Tab1 = read1();
$Tab2 = read2();
```

Q3 Compréhension de code. Toujours dans le même projet du team *kiproko*, voici un contrôleur responsable de la fusion séquentielle :

```
/**
 * @Route ("/fusionseq", name="fusionseq")
 */
public function fusionseq()
{
    $frenchtab = $this->read1();
    $germantab = $this->read2();
    $filer = 'download/vide.csv';
    $filef = fopen($filer, 'w');
    foreach ($frenchtab as $values) {
        if ($values) {
            fputcsv($filef, $values, ",");
        }
        foreach ($germantab as $values) {
            if ($values) {
                fputcsv($filef, $values, ",");
            }
        }
    }
    fclose($filef);
    return $this->redirectToRoute('fusion');
}
```

Le corps de boucle est appelé 3 fois

La boucle est appelée 3 fois aussi

Sachant que les tableaux \$frenchtab et \$germantab sont :

```
$frenchtab : [ ["nom", "prenom"], ["a","b"], ["c","d"] ]
$germantab : [ ["nom", "prenom"], ["d","e"], ["f","g"] ]
```

Ce type de tableau est un tableau de ligne (csv)

Déterminer le contenu du fichier de destination \$filef (indiquer sur votre copie le résultat obtenu lors d'une impression ou de l'affichage du contenu de ce fichier sur un terminal) - **6 points**

1. Nom, prenom → Premier élément du \$frenchtab

Nom, prenom
d, e
f, g

Tous les éléments du \$germantab

2. a, b → Deuxième élément du \$frenchtab

nom, prenom
d, e
f, g

Tous les éléments du \$germantab

3. c, d → Troisième élément du \$frenchtab

nom, prenom
d, e
f, g

Tous les éléments du \$germantab

Q4

```
<?php namespace jeu;
```

```
class Personnage {
    private $_force;
    private $_nom;
    private $_experience = 0;
    private $_degats = 0;

    // Déclarations des constantes en rapport avec la force.
    const FORCE_NULLE = 0;
    const FORCE_PETITE = 20;
    const FORCE_MOYENNE = 50;
    const FORCE_Grande = 80;

    public function __construct($_nom, $forceInitiale){
        $this->nom = nom;
        // assignation de sa force uniquement depuis son setter
        $this->setForce($forceInitiale);
    }

    public function setForce($force) {
        if (in_array($force,
            [self::FORCE_PETITE,
            self::FORCE_MOYENNE,
            self::FORCE_Grande
            ])
        )
        {
            $this->_force = $force;
        } else {
            $this->_force = self::FORCE_NULLE;
        }
    }

    public function getForce() : int {
        return $this->_force;
    }
    // + getters et setters pour les autres attributs privés
}
```

```
// programme de test
```

```
1 : $p1 = new Personnage("j1", 80);
2 : $p2 = new Personnage("j2", 25);
3 : echo "p1 : " . $p1->getForce(); // p1 a une force de 80
(FORCE_Grande)
4 : echo "p2 : " . $p2->getForce(); // p2 a une force de 0
(FORCE_NULLE)
5 : $p2->setForce($p1->getForce()); p2 prend la force de p1
6 : echo "p1 : " . $p1->getForce(); // p1 a une force de 80
(FORCE_Grande)
7 : echo "p2 : " . $p2->getForce(); // p2 a une force de 80
(FORCE_Grande)
8 : $p2 = $p1 la variable référence le même objet p2 référence le même
objet de p1
9 : $p1->setForce(50); p1 prend un force de 50
10: echo "p1 : " . $p1->getForce(); // p1 a une force de 50
(FORCE_MOYENNE)
11: echo "p2 : " . $p2->getForce(); // p2 a une force de 50
(FORCE_MOYENNE)
```

Votre réponse à Q4 ici en
face des lignes



Antérieur à php8.0.0, leur ordre était indéfini donc, avec une version de php supérieure ou égale à 8, le nom du premier joueur sera A3 car il est le premier à être ajouté dans le tableau (il a la plus grande force "80").

ANNEXE A

Exemple d'utilisation de la fonction *usort* et d'une fonction de comparaison *cmp* pour trier un tableau.

Exemple #1 Exemple avec usort()

```
<?php
function cmp($a, $b)
{
    if ($a == $b) {
        return 0;
    }
    return ($a < $b) ? -1 : 1;
}

$a = array(3, 2, 5, 6, 1);

usort($a, "cmp");

foreach ($a as $key => $value) {
    echo "$key: $value\n";
}
?>
```

L'exemple ci-dessus va afficher :

```
0: 1
1: 2
2: 3
3: 5
4: 6
```