Programmation PHP

le chargement automatique de classes

Un programme utilisant des classes

```php
require_once 'src/personapp/personne/Personne.php';
require_once 'src/personapp/personne/Enseignant.php';
require_once 'src/personapp/personne/Etudiant.php';
require_once 'src/personapp/afficheur/AfficheurDePersonne.php';
require_once 'src/personapp/afficheur/AfficheurDEnseignant.php';
require_once 'src/personapp/afficheur/AfficheurDEtudiant.php';

use \personapp\personne\Etudiant as Etudiant;



$p1 = new Etudiant('Jagger');


$p1->prenom='Mick'; $p1->age=23; $p1->noEtudiant = 435667;
$p1->refFormation='XDMA234'; $p1->groupe='2E';

$p2 = new \personapp\personne\Enseignant('Richards');
$p2->prenom='Keith'; $p2->age=22;
$p2->codeDiscipline = 27; $p2->composante='IUT-NC'; $p2->noBureau='0023';


$a = new \personapp\afficheur\AfficheurDEnseignant($p2);

$a->afficher('long');
```

Que faire si on a 1
gros projet
avec 200 classes et
10 librairies
de chacune 50 classes ???

réponse de PHP : le chargement automatique de classes

- Principe : lors de l'utilisation d'une classe (new, appel statique), si la classe n'est pas connue, l'interprète appelle une fonction de chargement de la classe.
- Cette fonction reçoit le nom (complet) de la classe en paramètre.
- Cette fonction peut être définie par le programmeur.
- Cette fonction est chargée de faire le "require" pour charger le fichier contenant la définition de la classe

pour que ça marche :

- il faut pouvoir calculer le nom du fichier à inclure à partir du nom de la classe.

- la fonction doit être enregistrée comme "autoloader",

utilisation et programmation d'autoloader

- il est possible d'enregistrer ***plusieurs*** autoloader,
- lord d'un défaut de nom, l'interprète parcours la pile d'autoloader jusqu'à ce que le nom soit défini,
- lorsque tous les autoloaders ont été exécutés, si le nom n'est pas défini : erreur

- **usage** : chaque librairie ou package réutilisable définit et enregistre son propre autoloader

programmation d'un autoloader

- un autoloader est une fonction ou une méthode :
  - reçoit le nom complet de la classe (avec le namespace) en paramètre
  - calcule le chemin vers le fichier correspondant
    - ajouter le répertoire contenant le namespace
    - ajouter le suffixe '.php'
    - change les séparateurs de namespace " \ " en séparateurs de répertoire (DIRECTORY_SEPARATOR)
  - charge le fichier avec require
- ne doit pas provoquer d'erreur pour laisser au suivant dans la pile une chance de réussir le chargement

enregistrement d'un autoloader
spl_autoload_register()

```
// enregistrement de la fonction fnAutoload

function fnAutoload($classname) { … }
spl_autoload_register( 'fnAutoload' ) ;


// enregistrement d'une fonction anonyme

spl_autoload_register( function($classname) {
                            …
                            require( $filename ) ;
                       } ) ;

// enregistrement d'une méthode d'un objet

$loader = new Loader() ;
spl_autoload_register( [ $loader, 'loadClass' ] ) ;
```

génération automatique d'un autoloader

- **composer** : gestionnaire de packages php, génération d'autoloader
- https://getcomposer.org/

- principe : télécharge et installe les librairies utiles au projet, génère l'autoloader du projet
- utilise un fichier de configuration nommé `composer.json`

```
{
  "name": "personapp",
  "type" : "project",

  "autoload": {
    "psr-4": {
      "personapp\\": "src/personapp",
      "conf\\" : "conf"
    }
  }
}
```

infos sur le projet

déclaration des autoloaders psr-4
le ns "personapp" se trouve dans le répertoire "src"
le ns "conf" se trouve dans le répertoire courant

- laissons la magie opérer :

```
keith:TD3.3 canals$ composer install
Loading composer repositories with package information
Installing dependencies (including require-dev)
Nothing to install or update
Generating autoload files
keith:TD3.3 canals$
```

- yapuca :

```php
require_once 'vendor/autoload.php';

use \personapp\personne\Etudiant as Etudiant;

$p1 = new Etudiant('Jagger');
```

PSR-0 vs PSR-4 avec composer

```
{
  "name": "personapp",
  "type" : "project",

  "autoload": {
    "psr-4": {
      "personapp\\": "src/personapp",
      "conf\\" : "conf"
    }
  }
}
```

```
{
  "name": "personapp",
  "type" : "project",

  "autoload": {
    "psr-0": {
      "personapp": "src",
      "conf" : "."
    }
  }
}
```