تكاليف:

الخليل عبدالفتاح سالم حسن

قسم IT/E

**تكليف: الثوابت و الأعضاء الثابته**

**الثوابت (const) :**

* **تعريف**: تُستخدم const لتحديد ثوابت (constants) داخل الفئة. هذه الثوابت لا يمكن تغيير قيمتها بعد تعريفها.
* **الوصول**: يمكن الوصول إلى الثوابت مباشرة من الفئة دون الحاجة إلى إنشاء كائن (object).
* **القيمة**: يجب أن تكون قيمة const ثابتة ولا يمكن تعيينها بقيمة متغيرة أو نتيجة دالة.
* **التخزين**: يتم تخزين الثوابت في ذاكرة البرنامج ولا تتغير طوال فترة تنفيذ البرنامج.

class MyClass {

const MY\_CONST = "Hello, World!";

public function showConstant() {

echo self::MY\_CONST;

}

}

echo MyClass::MY\_CONST ;

$obj = new MyClass() ;

$obj->showConstant ;

**الأعضاء الثابته (( static**

**تعريف**: تُستخدم static لتحديد أعضاء (properties أو methods) تنتمي إلى الفئة وليس إلى الكائن. يمكن تغيير قيمتها أثناء تنفيذ البرنامج.

* **الوصول**: يمكن الوصول إلى الأعضاء الثابتة مباشرة من الفئة أو من الكائن.
* **القيمة**: يمكن تعيين قيمة متغيرة أو نتيجة دالة للأعضاء الثابتة.
* **التخزين**: يتم تخزين الأعضاء الثابتة في ذاكرة البرنامج، ولكن يمكن تغيير قيمتها.

class MyClass {

public static $myStaticProperty = "I am static!";

public static function myStaticMethod() {

echo "This is a static method.";

}

public function showStaticProperty() {

echo self::$myStaticProperty;

}

}

echo MyClass::$myStaticProperty;

MyClass::$myStaticProperty = "New value” ;

echo MyClass::$myStaticProperty:

MyClass::myStaticMethod;

$obj = new MyClass();

$obj->showStaticProperty:

|  |  |  |
| --- | --- | --- |
| **الميزة** | **const** | **static** |
| **التعريف** | ثابت (constant) | خاصية أو دالة ثابتة (static) |
| **القيمة** | يجب أن تكون ثابتة | يمكن تغييرها |
| **الوصول** | ClassName::CONST\_NAME | ClassName::$property أو ClassName::method() |
| **التعديل** | لا يمكن تعديل القيمة | يمكن تعديل القيمة |
| **التخزين** | في ذاكرة البرنامج | في ذاكرة البرنامج |
| **الاستخدام** | للقيم الثابتة التي لا تتغير | للقيم التي يمكن أن تتغير |

الفرق بين ال const وال static :

تكليف : بحث على الدوال السحرية.

|  |  |
| --- | --- |
| **الدالة السحرية** | **الوصف** |
| \_\_construct() | يتم استدعاؤها عند إنشاء كائن جديد (المنشئ). |
| \_\_destruct() | يتم استدعاؤها عند تدمير الكائن. |
| \_\_call() | يتم استدعاؤها عند محاولة استدعاء دالة غير موجودة في الكائن. |
| \_\_callStatic() | يتم استدعاؤها عند محاولة استدعاء دالة ثابتة غير موجودة في الكائن. |
| \_\_get() | يتم استدعاؤها عند محاولة الوصول إلى خاصية غير موجودة أو غير عامة. |
| \_\_set() | يتم استدعاؤها عند محاولة تعيين قيمة لخاصية غير موجودة أو غير عامة. |
| \_\_isset() | يتم استدعاؤها عند استخدام isset() أو empty() على خاصية غير موجودة. |
| \_\_unset() | يتم استدعاؤها عند استخدام unset() على خاصية غير موجودة. |
| \_\_toString() | يتم استدعاؤها عند محاولة تحويل الكائن إلى سلسلة نصية. |
| \_\_invoke() | يتم استدعاؤها عند محاولة استدعاء الكائن كدالة. |
| \_\_sleep() | يتم استدعاؤها قبل تسلسل الكائن (serialization). |
| \_\_wakeup() | يتم استدعاؤها بعد إعادة تكوين الكائن (unserialization). |
| \_\_clone() | يتم استدعاؤها عند استنساخ الكائن باستخدام clone. |

تكليف : التعامل مع الملفات في ال pi6.

* **فتح الملفات (Opening Files)**

**تُستخدم الدالة  fopen()  لفتح ملفات في PHP. تأخذ هذه الدالة مسار الملف ووضع الفتح (mode) كمعاملات.**

**$file = fopen('example.txt', 'r');**

**أوضاع فتح الملفات (Modes):**

* 'r' : للقراءة فقط (افتراضي).
* 'w' : للكتابة (يُحذف محتوى الملف إذا كان موجودًا أو يُنشئ ملفًا جديدًا).
* 'a' : للإضافة إلى نهاية الملف (بدون حذف المحتوى الحالي).
* 'x' : لإنشاء ملف جديد، ولكن يُعيد خطأ إذا كان الملف موجودًا بالفعل.
* 'b' : لفتح الملف في الوضع الثنائي (binary mode)، مثل 'rb' أو 'wb' .
* '+' : لفتح الملف للقراءة والكتابة معًا، مثل 'r+' أو 'w+' .
* **قراءة الملفات (Reading Files)**

بعد فتح الملف، يمكنك قراءة محتواه باستخدام عدة طرق:

**أ. قراءة الملف كاملًا:**

**$content = fread($file, filesize('example.txt'));**

**echo $content;**

**ب. قراءة الملف سطرًا سطرًا:**

**while (!feof($file)) {**

**$line = fgets($file);**

**echo $line . "<br>";**

**}**

**ج. قراءة سطر واحد:**

**$line = fgets($file);**

**echo $line;**

**د. قراءة جميع الأسطر كمجموعة:**

**$lines = file('example.txt');**

**foreach ($lines as $line) {**

**echo $line . "<br>";**

**}**

* **كتابة الملفات (Writing Files)**

للكتابة في ملف، افتح الملف بوضع 'w' أو 'a'، ثم استخدم الدالة fwrite() .

**أ. الكتابة في ملف:**

**$file = fopen('example.txt', 'w');**

**fwrite($file, 'Hello, World!');**

**fclose($file);**

**ب. الإضافة إلى ملف:**

**$file = fopen('example.txt', 'a');**

**fwrite($file, '\nThis is a new line.');**

**fclose($file);**

* **إغلاق الملفات (Closing Files)**

يجب دائمًا إغلاق الملف بعد الانتهاء من استخدامه باستخدام الدالة fclose() .

**fclose($file);**

* **التعامل مع الملفات الثنائية (Binary Files)**

لقراءة أو كتابة ملفات ثنائية (مثل الصور أو الفيديوهات)، استخدم الوضع 'b' .

**قراءة ملف ثنائي:**

**$file = fopen('image.jpg', 'rb');**

**$binary\_data = fread($file, filesize('image.jpg'));**

**fclose($file);**

**كتابة ملف ثنائي:**

**$file = fopen('copy.jpg', 'wb');**

**fwrite($file, $binary\_data);**

**fclose($file);**

* **إدارة الملفات والمجلدات**

**(File and Directory Management)**

**توفر PHP العديد من الدوال لإدارة الملفات والمجلدات.**

**أ. إنشاء مجلد:**

**mkdir('new\_folder');**

**ب. حذف ملف:**

**unlink('example.txt');**

**ج. نسخ ملف:**

**copy('source.txt', 'destination.txt');**

**د. التحقق من وجود ملف:**

**if (file\_exists('example.txt')) {**

**echo 'File exists!';**

**}**

* **معالجة الأخطاء (Error Handling)**

**عند التعامل مع الملفات، قد تواجه أخطاء مثل عدم وجود الملف. يمكن استخدام try-catch للتعامل مع هذه الأخطاء.**

**try {**

**$file = fopen('non\_existent\_file.txt', 'r');**

**if (!$file) {**

**throw new Exception('File not found!');**

**}**

**} catch (Exception $e) {**

**echo 'Error: ' . $e->getMessage();**

**}**