

Лабораторная работа № 6

Принципы Объектно-ориентированного программирования

Основы работы с DOM

Задание

- В Вашей папке находящейся в корневой директории web-сервера создайте документ с названием l6.php.
- Задайте ему кодировку UTF-8.
- Также создайте дополнительную директорию users, в которой создайте 3 файла:
 - 1.ini
 - 2.ini
 - 3.ini

Каждый из этих файлов должен иметь следующую структуру:

```
FIO =  
Address =  
Phone =
```

Причём значения должны `FIO`, `Address` и `Phone` быть в каждом файле разные.

Пример:

```
FIO = Ivanov Ivan  
Address = Riga, Brivibas 117-97  
Phone = 12345678
```

Все следующие операции производятся в l5.php.

Создать класс содержащий информацию о пользователе и методы её получения из ini файла.

Требования к классу:

- Класс должен иметь три приватные (private) внутренние переменные, содержащие ФИО, адрес и телефон пользователя.
- Класс должен иметь метод вывода всей информации сохранённой в текущих значениях полей. (примерно следующее: `public function showMyData(){echo $this->fio.»
»...}`)
- Класс должен иметь private метод загрузки информации из ini файла (`get_data_from_ini(String iniFilePath)`). Для этого можно использовать встроенную функцию `bool parse_ini_file(string filename)` описание которой находится в разделе справочной информации данной лабораторной работы. Также должна производиться проверка на существование запрашиваемого файла, и если его нет, то функция должна возвращать **False**. Для проверки существования файла существует функция `bool file_exists (string filename)`, описание которой также находится в информационной части данной лабораторной работы.
- Класс должен иметь метод получения информации о пользователе без указания того, что данные хранятся в ini файлах. Необходимо замаскировать вызов функции `get_data_from_ini()` создав функцию `get_user_by_ID(Int uID)`. Эта функция должна заполнять внутренние переменные только на основе номера требуемого

пользователя, вызывая в своём теле `parse_ini_file()`, в таком случае полученный параметр `uID` является именем `ini` файла с информацией о нужном пользователе.

В случае, если информация о пользователе с таким `ID` недоступна, должно выдаваться сообщение и внутренние поля заполняются значениями `Unknown`.

- Класс должен иметь конструктор в котором производится заполнение внутренних переменных с помощью вызова метода `get_user_by_ID`, где `ID` требуемого пользователя передаётся в обязательном параметре конструктора.

Примечание: для подавления сообщений об ошибках, при вызове функции применяется символ `@`. К примеру, если в конструкторе класса ожидается параметр, то при вызове конструктора без параметров будет сгенерировано предупреждение `Warning: Missing argument 1 for robot::__construct()`.

Пример: `$robot = @new robot();`

Основы работы с DOM

- Создать простую форму содержащую следующие элементы ввода:
 - Текстовое поле с текстом “Текст по умолчанию” и именем `txt1`;
 - Кнопку с текстом “Изменить текст в текстовом поле”, при нажатии на эту кнопку, текст в текстовом поле `txt1` должен меняться на “Текст обновлён”.
 - Кнопку с текстом “Изменить самое себя”, при нажатии на эту кнопку, текст на этой кнопке должен меняться на “Мой новый текст”, при чём в функцию должен поступать объект `this` этой кнопки и манипуляции должны производиться с ним.
 - Вне формы создать дополнительно две кнопки, нажатие на которые соответственно делает все элементы доступными/недоступными (`disabled=true/false`).

Дополнительная информация

ООП

Создание класса с методами и свойствами

```
class A
{
    public    $pubVar = "Public";
    private  $prVar  = "Private var";

    public function fooPub(){
        echo $this->prVar;
    }
    private function fooPriv(){
        echo "Hello from class method!";
    }

    public function A($initStr){
        $this->prVar = $initStr;
    }
}

$ff = new A("Initializing private vars from konstruktor<br />");
$ff->fooPub();
echo $ff->pubVar;
```

Информация о функции *parse_ini_file*

Функция обрабатывает конфигурационный файл. Прототип выглядит так:

`array parse_ini_file (string filename [, bool process_sections])`

В итоге возвращается ассоциативный массив с ключами => значениями соответствующими структуре анализируемого файла.

Первый параметр имя ini файла, второй, необязательный, позволяет получить не только названия отдельных колонок, но и секции. По умолчанию второй параметр имеет значение FALSE.

Замечание: Существует зарезервированные слова, которые вы не можете использовать в качестве ключей в ini-файлах. Такими словами являются следующие: null, yes, no, true и false.

Замечание: Ключи и имена секций, состоящие из цифр, будут обработаны как целые числа в PHP, поэтому числа, начинающиеся с 0 будут считаться восьмеричными, а начинающиеся с 0x - шестнадцатеричными.

Пример: Имеется 1.ini файл следующего содержания:

FIO = Ivanov Ivan
Address = Riga, Brivibas 117-97
Phone = 12345678

[Another data]

FIO = Petrov Petr
Address = Olaine, Maizes 17-9
Phone = 987654321

Вызов функции `parse_ini_file` с различными параметрами приведёт к различным результатам:

```
$arr = parse_ini_file("users/1.ini");  
print_r($arr);
```

Выведет: Array ([FIO] => Ivanov Ivan [Address] => Riga, Brivibas 117-97 [Phone] => 12345678)

```
$arr = parse_ini_file("users/1.ini",true);  
print_r($arr);
```

Выведет: Array ([FIO] => Ivanov Ivan [Address] => Riga, Brivibas 117-97 [Phone] => 12345678 [Another data] => Array ([FIO] => Ivanov Ivan [Address] => Riga, Brivibas 117-97 [Phone] => 12345678))

Функция `file_exists()`

Функция проверяет существования файла, адрес которого получен в параметре. Если файл существует, то возвращается TRUE, в противном случае соответственно FALSE.

Пример:

```
if (file_exists("users/2.ini")) {  
    echo "The file users/2.ini exists";  
    return true;  
} else {  
    echo "The file users/2.ini NOT exists";  
    return false;  
}
```

Методы работы с DOM

Доступ к элементу формы по его имени

```
var lngfrom = document.forms.form1_lng_name.titleText;
```

Доступ к элементу формы по его ID

```
var lngfrom = document.getElementById("someElementID");
```

Изменение свойства (содержащийся в текстовом поле текст) элемента

```
lngfrom.value= "Текущий язык: Русский";
```

Дополнительный текст к тегу `<div id='div1'>`

```
var lngfrom = document.getElementById("div1");  
lngfrom.innerHTML = "Текущий язык: Русский";
```

Получение и использование указателя на самого себя (this).

```
<script language="JavaScript" type="text/javascript">  
    function chngThisValue(inElement, inTxt){  
        inElement.value=inTxt;  
    }  
  
</script>
```

```
<input type="button" value="this.Example" onClick="javascript:chgThisValue(this,'New  
text for this button');">
```

Функция делает доступными/недоступными все элементы формы:

```
function disableAll(formName,bFlag){  
    var frmObj    =    document.getElementById(formName);  
    for(i=0;i< frmObj.elements.length;i++)  
        frmObj.elements[i].disabled=bFlag;  
}
```