**ГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ**

**ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ**

**"ДОНЕЦКИЙ НАЦИОНАЛЬНЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ"**

Лабораторная работа № 7

Тема: «Работа с файлами и каталогами»

Проверил: Выполнил:

асс. каф. ПИ ст. гр. ПИ-18б

Артёменко О.Г. Моргунов А.Г.

\_\_\_\_.\_\_\_\_.2021г. \_\_\_\_.\_\_\_\_.2021г.

асс. каф. ПИ

Филипишин Д.А.

\_\_\_\_.\_\_\_\_.2021г.

Донецк – 2021

**Задание к лабораторной работе:**

1. Создать структуру файлов и каталогов для задания, часть из которых удовлетворяет ему, а часть – нет (при проверке могут быть добавлены внутренние каталоги).

2. Продумать, как будут показываться и сравниваться исходные данные и результаты.

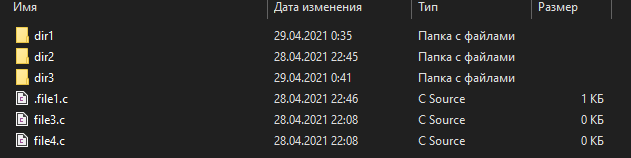
3. Все операции по чтению, сравнению и выводу на экран реализовать внутри класса, который поместить в пространство имён lab7, сам класс подключать средствами автоматической загрузки классов.

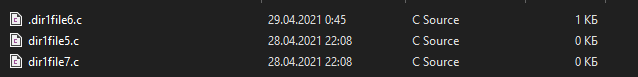
Вариант 11

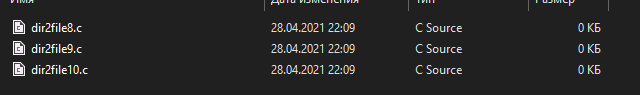
11. Пройти по дереву каталогов, начиная с текущего, и удалить все файлы на языке Си, содержащие внутри максимальное количество вложенных скобок более 2 и имеющие шаблон имени файла: начинаются с точки, потом - две буквы латинского алфавита, а далее любые и младше недели с правами только на чтение.

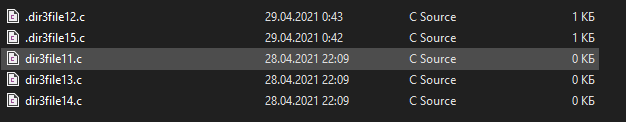
**Скриншоты**

Структура проверяемого каталога

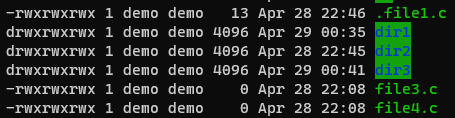






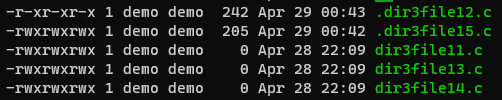


Права доступа для файлов

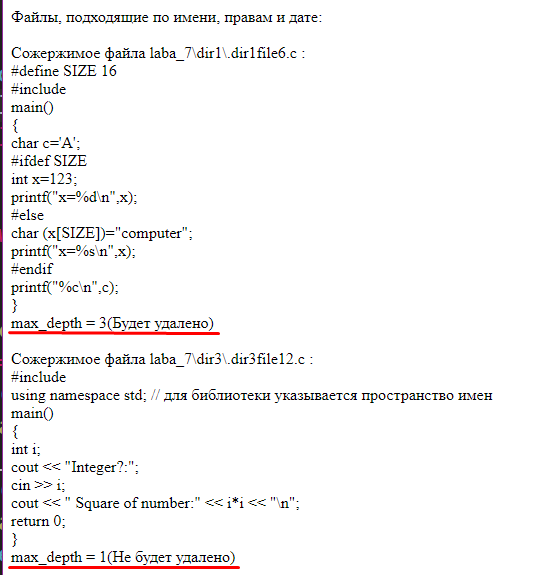








Результат выполнения программы



**Листинг програмы**

**myClass.php**

<?php

namespace lab7;

class myClass{

function getRecursiveCFiles($root) {

$all\_files\_list = new \RecursiveIteratorIterator(new \RecursiveDirectoryIterator($root), \RecursiveIteratorIterator::SELF\_FIRST);

$name\_files\_list = array();

$ended\_files\_list = array();

foreach ($all\_files\_list as $file) {

if (preg\_match('/^\.[A-z]{2}.\*\.c$/', basename($file))) {

$name\_files\_list[] = $file;

}

}

foreach ($name\_files\_list as $file) {

$file\_date = round((time()-date(filemtime($file))) / (60\*60\*24));

$file\_name = basename($file);

if(($file\_date) < 7 && is\_readable($file) && !is\_writable($file) && !is\_executable($file)){

$ended\_files\_list[] = $file;

}

}

return $ended\_files\_list;

}

public function scanDirectory($str){

foreach (glob($str."/\*") as $file) {

//echo basename($file) . " (size: " . filesize($file) . " bytes)" . "<br>";// delete

if (is\_dir($file)){

echo "<br>";// delete

$this->scanDirectory($file);

echo '<br>';// delete

} else if(preg\_match('/\./',basename($file))) {

$this->checkFile($file);

}

}

}

public function checkFile($str){

$max\_depth = 0;

$cur\_depth = 0;

echo '<br>' . "Сожержимое файла " . $str . " : " ;

$file = fopen($str,'r') or die("не удалось открыть файл.");

while (!feof($file)){

$cur\_string = fgets($file);

$string\_length = strlen($cur\_string);

echo '<br>';

for ($i = 0; $i < $string\_length; $i++) {

echo $cur\_string[$i];

switch ($cur\_string[$i]) {

case '(':

case '{':

case '[':

$cur\_depth++;

break;

case ')':

case '}':

case ']':

$cur\_depth--;

break;

}

if ($cur\_depth>$max\_depth){

$max\_depth=$cur\_depth;

}

}

}

fclose($file);

echo "<br>"."max\_depth = ".$max\_depth;

if ($max\_depth>2){

echo '(Будет удалено)<br>';

return true;

}

else{

echo '(Не будет удалено)<br>';

return false;

}

}

}

?>

**TRiSIS\_lab\_5\_main.php**

<?php

spl\_autoload\_register(function ($class\_name) {

$file\_name = preg\_replace('/(\\\)/','/', $class\_name);

include $file\_name . '.php';

});

$class = new lab7\myClass;

print("Файлы, подходящие по имени, правам и дате: <br>");

$files = $class->getRecursiveCFiles('laba\_7');

foreach ($files as $file) {

//print($file);

if($class->checkFile($file)){

unlink($file);

}

}

?>