Министерство науки и высшего образования РФ  
ФГАОУ ВПО

Национальный исследовательский технологический университет «МИСиС»

Институт Информационных технологий и компьютерных наук (ИТКН)

Кафедра Инфокоммуникационных технологий (ИКТ)

**Отчет по курсовой работе**

по дисциплине «Разработка клиент-серверных приложений»

Выполнил:  
студент группы БПИ20-2

Бобров М.А.

Проверил:   
Стучилин В.В.

Москва, 2022

**Задание**

Разработать скрипт, который считывает текстовый файл с сервера и выводит содержимое этого файла на экран. При этом, если файл содержит код PHP, то синтаксис языка должен подсвечиваться аналогично подсветке редактора Notepad++.

**Блок-схемы**

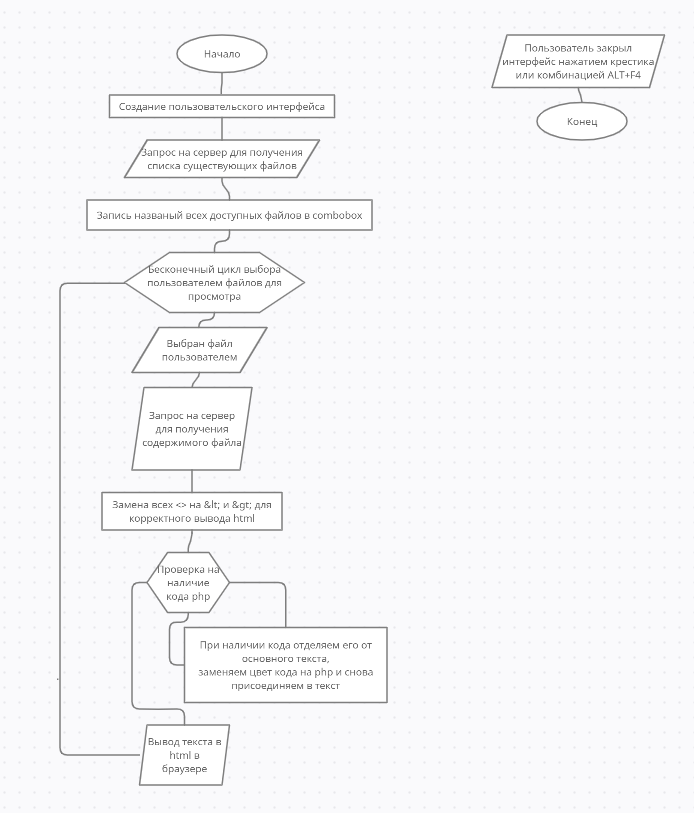


Рис. 1 – блок схема работы программы.

**Примеры выполнения**



Рис. 2 – пользовательский интерфейс.

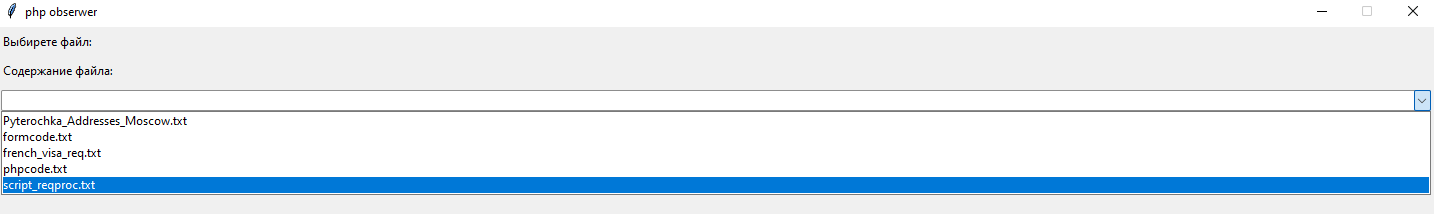


Рис. 3 – выбор файла в пользовательском интерфейсе.

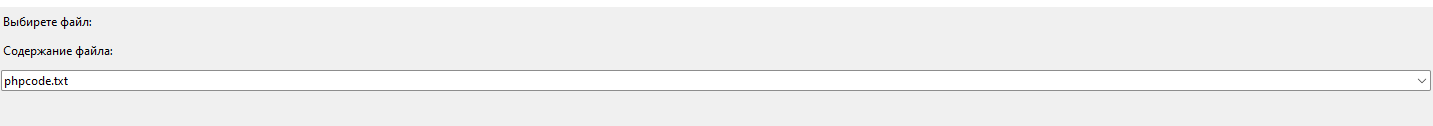


Рис. 4 – выбран файл в пользовательском интерфейсе.

Изображение выглядит как текст

Автоматически созданное описание

Рис. 5 – вывод файла в браузер с php.

Изображение выглядит как текст

Автоматически созданное описание

Рис. 6 – вывод файла в браузер с php.

Изображение выглядит как текст

Автоматически созданное описание

Рис. 7 – вывод файла в браузер без php.

**Листинг**

**kursprogramm.py:**

import os

import json

from tkinter import ttk

from tkinter import \*

import tkinter

import requests

def color\_qmarks(textin):

    if isinstance(textin, str):

        search\_start = 0

        qmi = -2

        qmit = -2

        put\_before = "<font color = \"red\">"

        put\_after = "</font>"

        while qmi != -1 and qmit != -1:

            qmi = textin.find("\"", search\_start)

            if qmi != -1:

                qmit = textin.find("\"", qmi + 1)

                if qmit != -1:

                    search\_start = len(textin[:qmit+1]) + len(put\_before) + len(put\_after)

                    textin = textin[:qmi] + put\_before + textin[qmi:qmit+1] + put\_after + textin[qmit+1:]

    return textin

def color\_amarks(textin):

    if isinstance(textin, str):

        search\_start = 0

        ami = -2

        amit = -2

        put\_before = "<font color = \"red\">"

        put\_after = "</font>"

        while ami != -1 and amit != -1:

            ami = textin.find("\'", search\_start)

            if ami != -1:

                amit = textin.find("\'", ami + 1)

                if amit != -1:

                    search\_start = len(textin[:amit+1]) + len(put\_before) + len(put\_after)

                    textin = textin[:ami] + put\_before + textin[ami:amit+1] + put\_after + textin[amit+1:]

    return textin

functions\_list = ("echo", "isset", "glob", "hihglight\_file", "fopen", "fread", "fclose")

def color\_functions(textin):

    if isinstance(textin, str):

        for function in functions\_list:

            len\_func = len(function)

            last\_index = 0

            index = -1

            index = textin.find(function, last\_index)

            put\_before = "<font color = \"green\">"

            put\_after = "</font>"

            while index != -1:

                if index != 1:

                    last\_index = index + len\_func + len(put\_before) + len(put\_after)

                    textin = textin[:index] + put\_before + textin[index:index+len\_func] + put\_after + textin[index+len\_func:]

                index = textin.find(function, last\_index)

    return textin

loops\_and\_conditions\_list = ("foreach", "elseif", "if", "else", "while", "for")

def color\_loopncondit(textin):

    if isinstance(textin, str):

        for lnc in loops\_and\_conditions\_list:

            len\_func = len(lnc)

            last\_index = 0

            index = -1

            index = textin.find(lnc, last\_index)

            put\_before = "<font color = \"orange\">"

            put\_after = "</font>"

            while index != -1:

                if index != 1:

                    last\_index = index + len\_func + len(put\_before) + len(put\_after)

                    textin = textin[:index] + put\_before + textin[index:index+len\_func] + put\_after + textin[index+len\_func:]

                index = textin.find(lnc, last\_index)

    return textin

def process\_php\_text(textin):

    if isinstance(textin, str):

        textin = color\_qmarks(textin)

        textin = color\_amarks(textin)

        textin = color\_functions(textin)

        textin = color\_loopncondit(textin)

    return textin

def replace\_tribrack(textin):

    if isinstance(textin, str):

        lti = -2

        rti = -2

        while lti != -1 and rti != -1:

            lti = textin.find("<", 0)

            if lti != -1:

                textin = textin[:lti] + "&lt;" + textin[lti+1:]

            rti = textin.find(">", 0)

            if rti != -1:

                textin = textin[:rti] + "&gt;" + textin[rti+1:]

    return textin

def replace\_nl(textin):

    if isinstance(textin, str):

        bri = -2

        while bri != -1:

            bri = textin.find("\n")

            if bri != -1:

                textin = textin[:bri] + "<br>" + textin[bri+1:]

    return textin

ui = Tk()

ui.geometry('1440x720')

ui.title("php obserwer")

ui.resizable(False, False)

label1 = ttk.Label(text="Выбирете файл:")

selectchoice = StringVar()

choicebox = ttk.Combobox(ui, textvariable=selectchoice)

label2 = ttk.Label(text="Содержание файла:")

getfiles = requests.post("http://localhost/spoilfiles.php")

files = json.loads(getfiles.text)

fileslist = []

for string in files:

    fileslist.append(string[8:])

choicebox['values'] = fileslist

choicebox['state'] = 'readonly'

label1.pack(fill=tkinter.X, padx=5, pady=5)

label2.pack(fill=tkinter.X, padx=5, pady=5)

choicebox.pack(fill=tkinter.X, padx=5, pady=5)

def choicechanged(event):

    getfilecontents = requests.post("http://localhost/reqproc.php", {'':selectchoice.get()})

    text = getfilecontents.text

    text = replace\_tribrack(text)

    phpopen = -2

    phpclose = -2

    find\_start = 0

    while phpopen != -1 and phpclose != -1:

        phpopen = text.find("&lt;?php", find\_start)

        if phpopen != -1:

            phpclose = text.find("?&gt;", phpopen+8)

            if phpclose != -1:

                texttophp = text[phpopen:phpclose+5]

                texttophp = process\_php\_text(texttophp)

                find\_start = len(text[:phpopen]) + len(texttophp) + 1

                text = text[:phpopen] + texttophp + text[phpclose+5:]

    text = replace\_nl(text)

    outfile = open("theoutput.html", "w")

    outfile.write(text)

    outfile.close()

    os.system("start theoutput.html")

choicebox.bind('<<ComboboxSelected>>', choicechanged)

ui.mainloop()

**xampp/htdocs/reqproc.php:**

<?php

    $data = file\_get\_contents('php://input');

    $data = "./files/" . substr($data, 1);

    $filecontent = file\_get\_contents($data);

    echo $filecontent;

?>

**xampp/htdocs/reqproc.php:**

<?php

    $arr = glob("./files/\*.txt");

    $encarr = json\_encode($arr);

    echo $encarr;

?>

**xampp/htdocs/files/formcode.txt:**

<h2> Форма для таблицы </h2>

<form action = "Z3Form.php" method=POST target="\_blank">

Строки <br><input type=text name="Rows"><br>

Столбцы <br><input type=text name="Columns"><br>

<p><input type=submit value="Отправить">

<input type=reset value="Отменить"></p>

</form>

**xampp/htdocs/files/phpcode.txt:**

<?php

$Rows = $\_REQUEST["Rows"];

$Columns = $\_REQUEST["Columns"];

echo '<html><body>';

echo '<table border="1">';

for($i = 1; $i <= $Rows; $i++)

{

echo '<tr>';

for($j = 1; $j <= $Columns; $j++)

{

echo '<td style="background-color:RGB(255,255,255);">'.$i.','.$j.'</td>';

}

echo '</tr>';

}

echo '</table>';

echo '</body></html>';

?>

**xampp/htdocs/files/script\_reqproc.txt:**

<?php

if(isset($\_POST["readonly"]))

{

foreach(glob("files/\*.txt") as $filename)

{

echo $filename . "<br>";

}

}

else

{

$path = $\_POST["path"];

if(!highlight\_file($path))

{

$file = fopen($path, "r");

$data = fread($file, 4);

fclose($file);

echo $data;

}

}

?>

**xampp/htdocs/files/french\_visa\_req.txt:**

The supporting documents to be provided are as below.

Pre-requisites:

-A travel document, issued less than 10 years ago,

containing at least two blank pages, with a period of

validity at least 3 months longer than the date on which

you intend to leave the Schengen Area or, in the case of a

long stay, at least three months longer than the expiry date

of the visa requested. Be sure to transmit (scan) ALL PAGES

of your travel document containing visas, entry and exit stamps

or any other inscription.

-2 ID photograph.

Civil status:

-Proof of family relationship (civil registry certificates).

**xampp/htdocs/files/Pyterochka\_Addresses\_Moscow.txt:**

Московское ш., д. 84

Дурыманова, д.2

Интернациональная, 11а

Парковая, 10 кор. 2

ул. Вокзальная, д. 44

**Выводы**

Я применил знания, полученные во время прохождения дисциплины «Разработка клиент-серверных приложений». В итоге я создал программу для пользователя и скрипты для сервера, полностью удовлетворяющий условию задания.

**Список используемой литературы**

1. [PHP: glob - Manual](https://www.php.net/manual/en/function.glob.php)
2. [PHP: php:// - Manual](https://www.php.net/manual/en/wrappers.php.php#:~:text=php%3A%2F%2Finput%20is%20a%20read-only%20stream%20that%20allows%20you,allows%20direct%20access%20to%20the%20given%20file%20descriptor.)
3. [tkinter — Python interface to Tcl/Tk — Python 3.10.4 documentation](https://docs.python.org/3/library/tkinter.html)