



Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет
имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ «Специальное машиностроение»

КАФЕДРА «Колесные машины»

Отчет по лабораторным работам

По курсу «разработка Веб-приложений»

Студент группы СМ10-61Б

_____ М.А. Пёрушкин

Преподаватель

_____ (Подпись, дата)

Ю.Е. Гапанюк

1 Лабораторная работа №1

В ходе этой лабораторной работы были изучены основы языка программирования Python, в том числе типы данных, операторы, условные конструкции, циклы, функции.

По итогам лабораторной работы было выдано задание написать программу для нахождения корней биквадратного уравнения. Листинг этой программы представлен ниже:

```
import math

print("Введите первый коэффициент")
a= int(input())

print("Введите второй коэффициент")
b= int(input())

print("Введите третий коэффициент")
c= int(input())

print("Находим дискриминант... ")
D=b**2-(4*a*c)

print("Дискриминант равен = ",D)
if D > 0 :
    x1 = (-b+math.sqrt(D))/(2*a)
    x2 = (-b-math.sqrt(D))/(2*a)
    if (x1 > 0) and (x2 > 0) :
        print("Есть 4 корня")
        x1_1=math.sqrt(x1)
        x1_2=(-x1_1)
        print("Первый корень = ",x1_1,"Второй корень = ",x1_2)
        x2_1=math.sqrt(x2)
        x2_2=(-x2_1)
        print("Третий корень = ",x2_1,"Четвёртый корень = ",x2_2)
```

```
elif x1 < 0 :  
    print("Есть 2 корня")  
    x2_1=math.sqrt(x2)  
    x2_2=(-x2_1)  
    print("Первый корень =",x2_1,"Второй корень =",x2_2)  
elif x2 < 0:  
    print("Есть два корня")  
    x1_1=math.sqrt(x1)  
    x1_2=(-x1_1)  
    print("Первый корень =",x1_1,"Второй корень =",x1_2)  
elif (x1 < 0) and (x2 < 0) :  
    print("Нет корней")  
elif D==0:  
    print("Есть 1 корень")  
    x1 = -b/(2*a)  
    print("Он равен =",x1)  
else:  
    print("Нет корней")
```

Фотография с лабораторной работы с рабочим кодом представлена
ниже:

The screenshot shows a Notepad++ window with the file `lab0.py` open. The code implements a quadratic equation solver using the `math` module. It prompts the user for coefficients a , b , and c , calculates the discriminant $d = b^2 - 4ac$, and then determines the number and nature of solutions based on the value of d . The code handles all possible cases: two positive real roots ($d > 0$), two negative real roots ($d < 0$), one real root ($d == 0$), and no real roots ($d < 0$).

```
1 import math
2
3 print("Напишите первый коэф")
4 a = int(input())
5 print("Напишите второй коэф")
6 b = int(input())
7 print("Напишите третий коэф")
8 c = int(input())
9 print("Вы ввели ", a,b,c)
10
11 d = b*b - 4*a*c
12 print("Д:", d)
13
14 if d>0:
15     t1 = (-b + math.sqrt(d))/(2*a)
16     t2 = (-b - math.sqrt(d))/(2*a)
17     if t1>0 and t2>0:
18         x1 = math.sqrt(t1)
19         x2 = -math.sqrt(t1)
20         x3 = math.sqrt(t2)
21         x4 = -math.sqrt(t2)
22         print ("Четыре корня:", x1, x2, x3, x4)
23
24 elif t1<0 and t2>0:
25     x1 = math.sqrt(t2)
26     x2 = -math.sqrt(t2)
27     print ("Два корня:", x1, x2)
28
29 elif t1>0 and t2<0:
30     x1 = math.sqrt(t1)
31     x2 = -math.sqrt(t1)
32     print ("Два корня:", x1, x2)
33
34 else:
35     print ("Нет корней")
36
37 elif d==0:
38     t = (-b)/(2*a)
39     if t>0:
40         x1 = math.sqrt(t)
41         x2 = -math.sqrt(t)
42         print("Два корня:", x1, x2)
43
44 elif t<0:
45     print ("Нет корней")
46
47 else:
48     print("Нет корней")|
```

2 Лабораторная работа №2

В ходе второй лабораторной работы были изучены основы использования фреймворка Flask в языке программирования Python, создание своего локального сайта, работа с функциями. Разработано простое веб-приложение, демонстрирующее принцип работы Flask и возможность создания веб-серверов на основе данного фреймворка.

Листинг программного кода, который был написан на лабораторной работе представлен далее:

```
from flask import Flask
from flask import request
app = Flask(__name__)
@app.route("/")
def hello_world():
    return "<p>Hello, World!<p>"
@app.route("/test")
def test():
    return "<p>Denis Kholod<p>"
@app.route("/test/test2")
def test2():
    return "<p>Sleva ot meny<p>"
@app.route("/name/<user>")
def name(user):
    return "<p>Привет, {} <p>".format(user)
@app.route("/calc/sum/<a>/<b>")
def calc_sum(a,b):
    a = int(a)
    b = int(b)
    c = a+b
```

```

    return "<p>Сумма: { } <p>".format(c)

@app.route("/calc/del/<e>/<s>")
def calc_del(e,s):
    e = int(e)
    s = int(s)
    x = e/s

    return "<p>Деление: { } <p>".format(x)

@app.route("/calc/umn/<k>/<d>")
def calc_umn(k,d):
    k = int(k)
    d = int(d)
    f = k*d

    return "<p>Умножение: { } <p>".format(f)

@app.route("/calc-minus/")
def calc_minus():

    args_dict = request.args
    print(args_dict)
    a = float (args_dict["a"])
    b = float (args_dict["b"])
    c = a - b

    return "<p>Разность: { } <p>".format(c)

@app.route("/food")
def food():

    args_dict = request.args
    print(args_dict)
    a = args_dict["первое"]
    b = args_dict["второе"]
    c = 0
    d = 0

    if a == "суп":

```

```
c = 100
if a == "солянка":
    c = 50
if a == "окрошка":
    c = 75
if b == "пюре":
    d = 60
    t = c+d
return "<p>Стоимость за {} и {}: {} </p>".format(a,b,t)
@app.route("/mywaifu/<name>")
def mywaifu_pic(name):
    return ''.format(name)
```

Важно, что в последней строчке необходимо правильно указывать формат изображения. В противном случае картинка не откроется, и вместо нее будет символ ошибки загрузки изображения.

Фотографии с лабораторной работы с рабочим кодом представлены ниже:

The screenshot shows a Notepad++ window with the file `lab22.py` open. The code implements a Flask application with various routes and functions:

```
1  from flask import Flask
2  from flask import request
3
4  app = Flask(__name__)
5
6  @app.route("/")
7  def hello_world():
8      return "<p>Hello World!</p>"
9
10 @app.route("/test")
11 def test():
12     return "<p>Test</p>"
13
14 @app.route("/test/aboba")
15 def test_aboba():
16     return "<p>Test_aboba</p>"
17
18 @app.route("/name/<user>")
19 def name(user):
20     return "<p>Привет, {}</p>".format(user)
21
22 @app.route("/calc/sum/<a>/<b>")
23 def calc_sum(a,b):
24     a = int(a)
25     b = int(b)
26     c = a + b
27     return "<p>Сумма: {}</p>".format(c)
28
29 @app.route("/calc/mult/<d>/<e>")
30 def calc_mul(d,e):
31     d = int(d)
32     e = int(e)
33     f = d * e
34     return "<p>Произведение: {}</p>".format(f)
35
36 @app.route("/calc/dev/<a>/<b>")
37 def calc_dev(a,b):
38     a = float(a)
39     b = float(b)
40     c = a / b
41     return "<p>Частное: {}</p>".format(c)
42
43 @app.route("/calc/sub/")
44 def calc_sub():
45     args_dict = request.args
46     print(args_dict)
47     a = float(args_dict["a"])
48     b = float(args_dict["b"])
49     c = a - b
50     return "<p>Вычитание: {}</p>".format(c)
51
52 @app.route("/food")
53 def food():
```

The screenshot shows a Notepad++ window displaying a Python script named `lab22.py`. The code defines three routes: `/calc/sub/`, `/food`, and `/voyk/<name>`. The `/calc/sub/` route performs subtraction. The `/food` route calculates the total cost of two items based on their names. The `/voyk/<name>` route returns an image from a local static directory.

```
43     @app.route("/calc/sub/")
44     def calc_sub():
45         args_dict = request.args
46         print(args_dict)
47         a = float(args_dict["a"])
48         b = float(args_dict["b"])
49         c = a - b
50         return "<p>Вычитание: {}</p>".format(c)
51
52     @app.route("/food")
53     def food():
54         args_dict = request.args
55         p = args_dict["первое"]
56         r = args_dict["второе"]
57         c = 0
58         d = 0
59         t = 0
60         if p == "суп":
61             c = 100
62         if p == "борщ":
63             c = 80
64         if p == "шкі":
65             c = 50
66         if r == "шициль":
67             d = 120
68         if r == "котлета":
69             d = 75
70         if r == "курица":
71             d = 95
72         t=c + d
73         return "Стоймость за {} и {}: {}".format(p,r,t)
74
75
76     @app.route("/voyk/<name>")
77     def voyk_pic(name):
78         return ''.format(name)
79
```

```
C:\Windows\System32\cmd.exe - flask run
Defaulting to user installation because normal site-packages is not writeable
ERROR: Could not find a version that satisfies the requirement flask (from versions: none)
ERROR: No matching distribution found for flask
WARNING: You are using pip version 22.0.4; however, version 23.0.1 is available.
You should consider upgrading via the 'C:\Program Files (x86)\Python37-32\python.exe -m pip install -
mand.

C:\Users\Lab5UIVC505-506\Desktop\Пёрушкин>set FLASK_APP=lab22.py
C:\Users\Lab5UIVC505-506\Desktop\Пёрушкин>flask run
"flask" не является внутренней или внешней
командой, исполняемой программой или пакетным файлом.

C:\Users\Lab5UIVC505-506\Desktop\Пёрушкин>python -m venv .
C:\Users\Lab5UIVC505-506\Desktop\Пёрушкин>Scripts\activate
"Scripts" не является внутренней или внешней
командой, исполняемой программой или пакетным файлом.

C:\Users\Lab5UIVC505-506\Desktop\Пёрушкин>cd Scripts
C:\Users\Lab5UIVC505-506\Desktop\Пёрушкин\Scripts>activate
(Пёрушкин) C:\Users\Lab5UIVC505-506\Desktop\Пёрушкин\Scripts>cd ..
(Пёрушкин) C:\Users\Lab5UIVC505-506\Desktop\Пёрушкин>flask
"flask" не является внутренней или внешней
командой, исполняемой программой или пакетным файлом.

(Пёрушкин) C:\Users\Lab5UIVC505-506\Desktop\Пёрушкин>pip intsall flask
ERROR: unknown command "intsall" - maybe you meant "install"

(Пёрушкин) C:\Users\Lab5UIVC505-506\Desktop\Пёрушкин>pip install flask
Collecting flask
  Using cached https://files.pythonhosted.org/packages/95/9c/a3542594ce4973786236a1b7b702b8ca81dbf40ea27
348/Flask-2.2.3-py3-none-any.whl
Collecting importlib-metadata>=3.6.0; python_version < "3.10" (from flask)
  Using cached https://files.pythonhosted.org/packages/f8/7d/e3adad613703c86d62aa991b45d6f990cf59975078a
948/importlib_metadata-6.1.0-py3-none-any.whl
Collecting itsdangerous>=2.0 (from flask)
  Using cached https://files.pythonhosted.org/packages/68/5f/447e04e828f47465eeab35b5d408b7ebaaaee207f48
0ae/itsdangerous-2.1.2-py3-none-any.whl
Collecting Jinja2>=3.0 (from flask)
  Using cached https://files.pythonhosted.org/packages/bc/c3/f068337a370801f372f2f8f6bad74a5c140f6fdad3d9c
c65/Jinja2-3.1.2-py3-none-any.whl
Collecting Werkzeug>=2.2.2 (from flask)
  Using cached https://files.pythonhosted.org/packages/f6/f8/9da63c1617ae2a1dec2fbf6412f3a0cfe9d4ce029ec0
45f/Werkzeug-2.2.3-py3-none-any.whl
Collecting click>=8.0 (from flask)
  Using cached https://files.pythonhosted.org/packages/c2/f1/df59e28c642d583f7dacffb1e0965d0e00b218e0186d
840/click-8.1.3-py3-none-any.whl
Collecting typing-extensions>=3.6.4; python_version < "3.8" (from importlib-metadata>=3.6.0; python_versi
flask)
  Using cached https://files.pythonhosted.org/packages/31/25/5abcd82372d3d4a3932e1fa8c3dbf9efac10cc7c0d16
404/typing_extensions-4.5.0-py3-none-any.whl
Collecting zipp>=0.5 (from importlib-metadata>=3.6.0; python_version < "3.10"->flask)
  Using cached https://files.pythonhosted.org/packages/5b/fa/c9e82bbe1af6266adf08afb563905eb87cab83fde00a
c5-803a061577b
```

3 Лабораторная работа №3

Третья лабораторная работа по Python была посвящена освоению работы с веб-страницами. Было произведена работа с файлами форматов .html, .css и .js. Получены знания по стилевому оформлению сайта, вставки картинок нужного разрешения и размера. Также настроена простейшая навигация и авторизация пользователя на главной странице.

3.1 Листинг кода из файла index.html.txt:

```
<!doctype html>
<html>
  <head>
    <title> Это заголовок </title>
  </head>
  <body>
    <p> Это содержимое
      <strong> стронг </strong>
      вот такое вот </p>
    </body>
</html>
```

3.2 Листинг кода из файла script.js:

```
a=5
b=10
console.log("Привет",a*b)
function myfunction()
{
  n = document.getElementById("in_name").value;
  a = document.getElementById("in_age").value;
  response = "Имя: " +" "+ n +" "+ "Возраст: " + a
  alert(response)
```

```
t = document.getElementById("mytable")
var row = t.insertRow(3);
var c_name = row.insertCell(0);
var c_photo = row.insertCell(1);
var c_task = row.insertCell(2);
c_name.innerHTML = n;
c_photo.innerHTML = '';
c_task.innerHTML = "Сотрудник"
}
```

3.3 Листинг кода из файла style.css:

```
body {
    background-color: orange;
    color: black;
}

#otv_face {
    background-color: red;
    color: white;
    width: 150px;
    margin: 20px;
}

#my_photo {
    margin: auto;
}

.myheader {
    margin: 20px;
    font-family: "Courier New";
    color: green;
}
```

Фотографии с лабораторной работы с рабочим кодом представлены ниже:

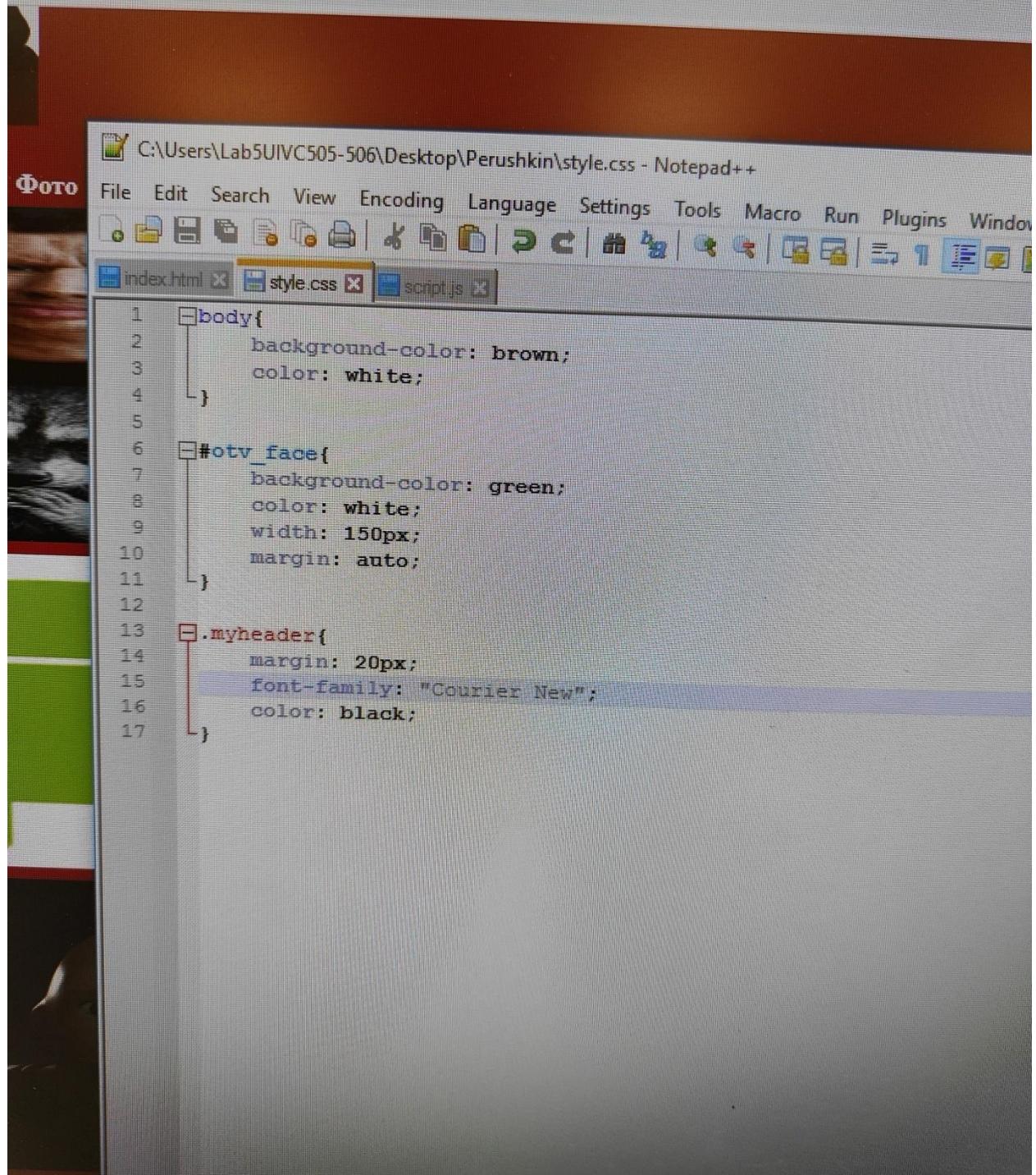
C:\Users\Lab5UIVC505-506\Desktop\Perushkin\script.js - Notepad++

File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?

index.html style.css script.js

```
7 function myfunction() {
8     n = document.getElementById("in_name").value;
9     a = document.getElementById("in_age").value;
10
11    response = "Имя: " + n + " Возраст: " + a
12    alert(response)
13    t = document.getElementById("mytable")
14    var row = t.insertRow(4);
15    var c_name = row.insertCell(0);
16    var c_photo = row.insertCell(1);
17    var c_task = row.insertCell(2);
18    var c_salary = row.insertCell(3);
19    c_name.innerHTML = n;
20    c_photo.innerHTML = '';
21    c_task.innerHTML = "Сотрудник";
22    c_salary.innerHTML = "50000";
23}
```

/Desktop/Perushkin/index.html



The screenshot shows a Notepad++ window with the title bar "C:\Users\Lab5UIVC505-506\Desktop\Perushkin\style.css - Notepad++". The menu bar includes File, Edit, Search, View, Encoding, Language, Settings, Tools, Macro, Run, Plugins, and Windows. Below the menu is a toolbar with various icons. The main editor area contains three tabs: "index.html", "style.css" (which is currently active), and "script.js". The "style.css" tab displays the following CSS code:

```
1 body{
2     background-color: brown;
3     color: white;
4 }
5
6 #otv_face{
7     background-color: green;
8     color: white;
9     width: 150px;
10    margin: auto;
11 }
12
13 .myheader{
14     margin: 20px;
15     font-family: "Courier New";
16     color: black;
17 }
```

Desktop/Perushkin/index.html

Фото

C:\Users\Lab5UV\VC505-506\Desktop\Perushkin\index.html - Notepad++

File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?

index.html style.css script.js

```
1 <!doctype html>
2 <html>
3   <head>
4     <link rel="stylesheet" href = "style.css">
5     <script src = "script.js"> </script>
6     <title>
7       Лабораторная 3 Университет
8     </title>
9   </head>
10  <body>
11    <div class="myheader">
12      <h1>Кафедра СМ-10</h1>
13    </div>
14    <div class="myheader">
15      <h2>
16        Курс Веб-технологии
17      </h2>
18    </div>
19    <div class="myheader">
20      <h3>
21        Лабораторные:
22      </h3>
23    </div>
24    <div>
25      <ul>
26        <li>Лаб1</li>
27        <li>Лаб2</li>
28        <li>Лаб3</li>
29      </ul>
30    </div>
31    <div id = 'otv_face'>
32      Ответственное лицо:
33      
34    </div>
35    <div>
36      Не ответственное лицо:
37      
38    </div>
39    <div>
40      Сотрудники:
41      <table id = "mytable">
```

Hyper Text Markup Language file length : 2 490 lines : 102 Ln : 48 Col : 13 Pos :

esktop/Perushkin/index.html

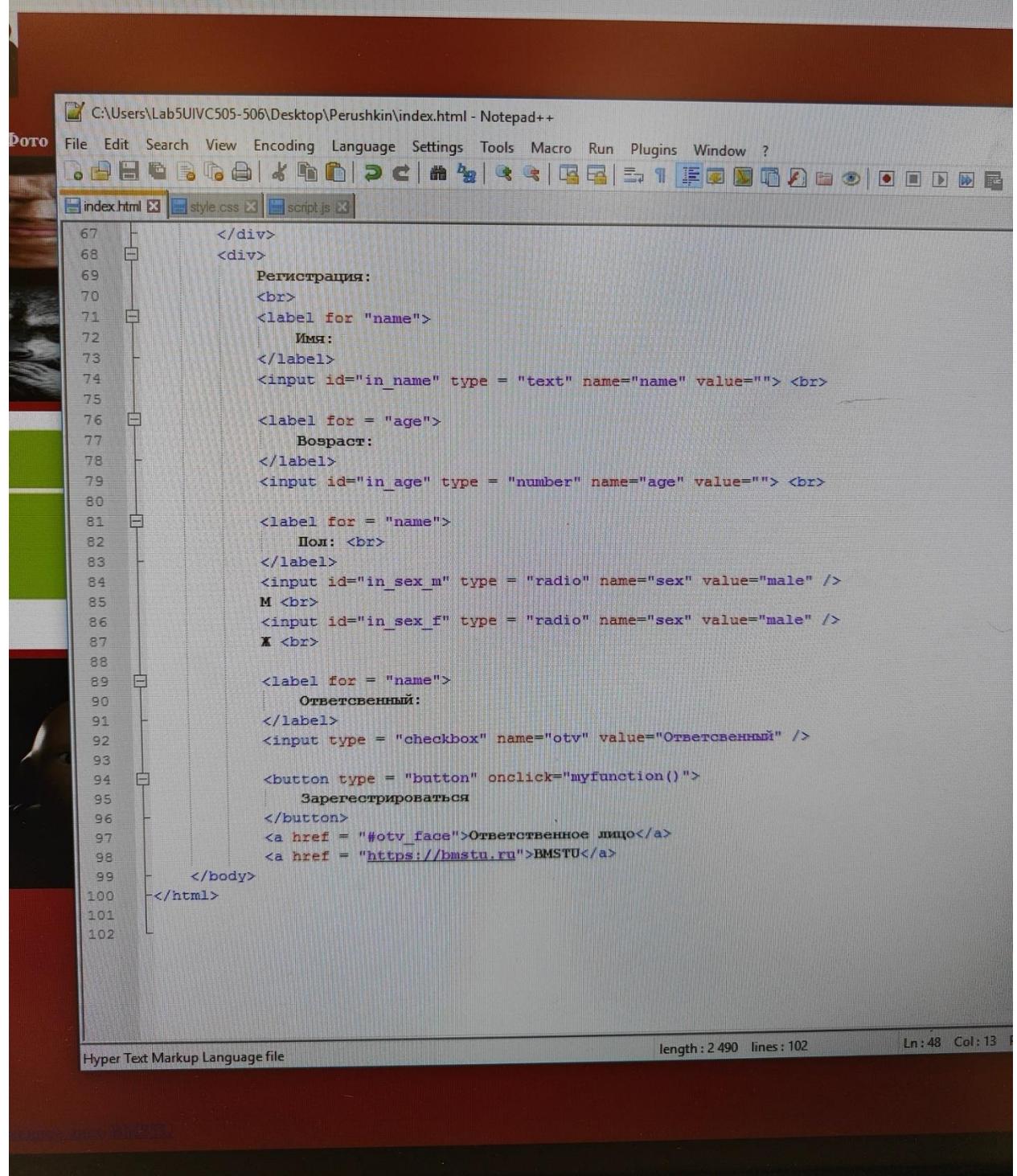
C:\Users\Lab5UIVC505-506\Desktop\Perushkin\index.html - Notepad++

OTO File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?

index.html style.css script.js

```
37 
38 </div>
39 <div>
40 Сотрудники:
41 <table id = "mytable">
42 <tr>
43     <th>Имя</th>
44     <th>Фото</th>
45     <th>Должность</th>
46     <th>Зарплата</th>
47 </tr>
48 <tr>
49     <th>Влад</th>
50     <th>  </th>
51     <th>Повар</th>
52     <th>10000</th>
53 </tr>
54 <tr>
55     <th>Андроид</th>
56     <th></th>
57     <th>Инженер</th>
58     <th>20000</th>
59 </tr>
60 <tr>
61     <th>Бован</th>
62     <th></th>
63     <th>Охранник</th>
64     <th>15000</th>
65 </tr>
66 </table>
67 </div>
68 <div>
69     Регистрация:
70 <br>
71     <label for = "name">
72         Имя:
73     </label>
74     <input id="in_name" type = "text" name="name" value=""> <br>
75
76     <label for = "age">
77         Возраст:
```

Hyper Text Markup Language file length : 2 490 lines : 102 Ln : 48 Col : 13 Pos : 111



The screenshot shows the Notepad++ interface with the file `C:\Users\Lab5UIVC505-506\Desktop\Perushkin\index.html` open. The code is an HTML form for registration. It includes fields for name, age, gender (radio buttons), and a checkbox for ' ответственный'. A button triggers a function, and links are provided for 'Ответственное лицо' and 'BMSTU'.

```
67 </div>
68 <div>
69     Регистрация:
70     <br>
71     <label for="name">
72         Имя:
73     </label>
74     <input id="in_name" type="text" name="name" value=""> <br>
75
76     <label for="age">
77         Возраст:
78     </label>
79     <input id="in_age" type="number" name="age" value=""> <br>
80
81     <label for="name">
82         Пол: <br>
83     </label>
84     <input id="in_sex_m" type="radio" name="sex" value="male" />
85     М <br>
86     <input id="in_sex_f" type="radio" name="sex" value="female" />
87     Ж <br>
88
89     <label for="name">
90         Ответственный:
91     </label>
92     <input type="checkbox" name="otv" value="Ответственный" />
93
94     <button type="button" onclick="myfunction()">
95         Зарегистрироваться
96     </button>
97     <a href="#otv_face">Ответственное лицо</a>
98     <a href="https://bmstu.ru">BMSTU</a>
99
100    </body>
101
102  </html>
```

Length : 2.490 Lines : 102 Ln: 48 Col: 13

4 Лабораторная работа №4

Четвёртая лабораторная работа по Python была посвящена применению ранее полученных знаний и опыта работы с фреймворком Flask, HTML, CSS и JavaScript. Было изучено использование классов и списков в Python, а также применение их при разработке web-приложения с использованием Flask. В рамках лабораторной работы было разработано приложение для создания списка задач с возможностью добавления, редактирования и удаления задач. Получен опыт в создании и использовании классов и списков в Python, а также в интеграции данной функциональности с веб-приложением на Flask.

Листинг программного кода этой лабораторной работы представлен далее:

```
from flask import Flask, render_template,request  
from json import dumps as jsonstring  
  
app = Flask(__name__)  
  
class Department(object):  
    def __init__(self, name, floor, zavkaf, students,image):  
        self.name = name  
        self.floor = floor  
        self.zavkaf = zavkaf  
        self.students = students  
        self.image = image  
  
    def __str__(self):  
        return("Название: ",self.name,  
              " Этаж:",self.floor,  
              " Завкаф:",self.zavkaf,
```

```

    " число студентов:", self.students,
    " картинка:", self.image)

class Faculty(object):

    def __init__(self, name, departments, decan):
        self.name = name
        self.departments = departments
        self.decan = decan

    def __str__():
        return("Название: ",self.name,
              " Кафедры:",self.departments,
              " Декан:",self.decan)

dep_sm10 = Department("CM-10",7,"Котиев Г.О",50,"Leha.png")
dep_sm9 = Department("CM-9",3,"Горелов В.А",40,"rem.png")
dep_sm1 = Department("CM-1",4,"Завкаф И.И",60,"felix.png")

deps = [dep_sm1,dep_sm9,dep_sm10]

faculty_sm = Faculty("CM",deps,"Калугин В.Т.")

@app.route("/")
def hello_world():
    return render_template('index.html',faculty = faculty_sm)

@app.route("/new_dep")
def adding():
    name = request.args.get('name')
    floor = request.args.get('floor')

```

```
students = request.args.get('students')
zavkaf = request.args.get('zavkaf')
new_dep = Department(name,floor,zavkaf,students,"Leha.png")
faculty_sm.departments.append(new_dep)
return "Добавил"
```

```
@app.route("/delete")
def deleting():
    name = request.args.get('name')
    c = 0
    for d in faculty_sm.departments:
        if d.name == name:
            del faculty_sm.departments[c]
            return "Удалил" + d.name
        else:
            c = c + 1
    return "Не нашёл такую кафедру"
```

Фотографии с лабораторной работы с рабочим кодом представлены
ниже:

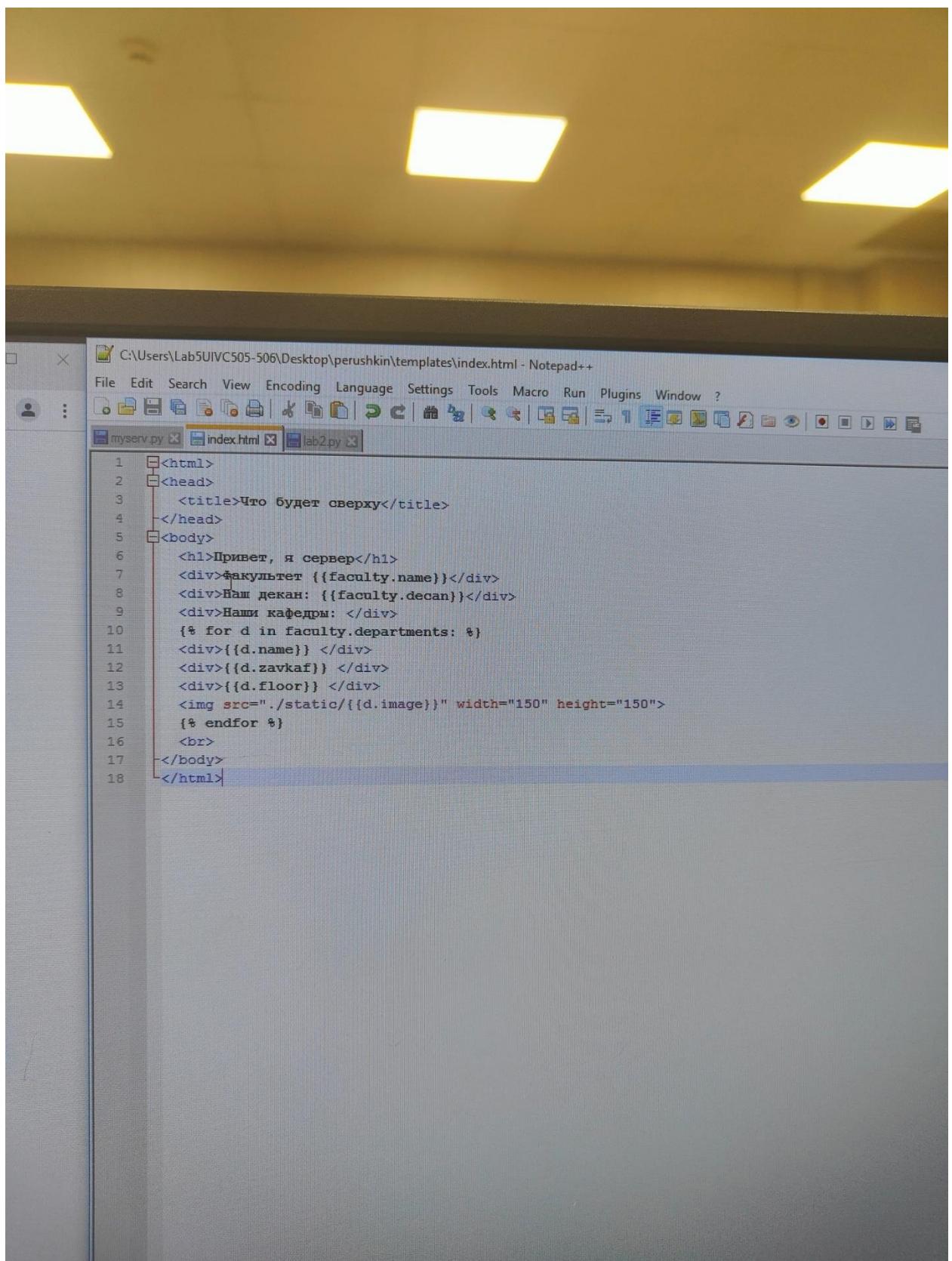
The screenshot shows a Notepad++ window with the following Python code:

```
1  from flask import Flask, render_template, request
2  from json import dumps as jsonstring
3
4
5  app = Flask(__name__)
6
7  class Department(object):
8      def __init__(self, name, floor, zavkaf, students, image):
9          self.name = name
10         self.floor = floor
11         self.zavkaf = zavkaf
12         self.students = students
13         self.image = image
14
15     def __str__(self):
16         return("Название: ",self.name,
17               " Этаж:",self.floor,
18               " Завкаф:",self.zavkaf,
19               " число студентов:", self.students,
20               " картинка:", self.image)
21
22 class Faculty(object):
23     def __init__(self, name, departments, decan):
24         self.name = name
25         self.departments = departments
26         self.decan = decan
27
28     def __str__(self):
29         return("Название: ",self.name,
30               " Кафедры:",self.departments,
31               " Декан:",self.decan)
32
33 dep_sm10 = Department("CM-10",7,"Котиев Г.О",50,"sm10.jpg")
34 dep_sm9 = Department("CM-9",3,"Горелов В.А",40,"sm9.jpg")
35 dep_sm1 = Department("CM-1",4,"Завкаф И.И",60,"sm1.jpg")
36
37 deps = [dep_sm1,dep_sm9,dep_sm10]
38
39 faculty_sm = Faculty("CM",deps,"Калугин В.Т.")
40
41 @app.route("/")
42 def hello_world():
43     return render_template('index.html',faculty = faculty_sm)
44
45 @app.route("/new_dep")
46 def adding():
47     name = request.args.get('name')
48     floor = request.args.get('floor')
49     students = request.args.get('students')
50     zavkaf = request.args.get('zavkaf')
51     new_dep = Department(name,floor,zavkaf,students,"sm9.jpg")
52     faculty_sm.departments.append(new_dep)
```

The code defines a `Department` class with attributes `name`, `floor`, `zavkaf`, `students`, and `image`. It includes a `__init__` method and a `__str__` method for string representation. A `Faculty` class is defined with attributes `name`, `departments`, and `decan`, also with a `__str__` method. Three instances of `Department` are created: `dep_sm10`, `dep_sm9`, and `dep_sm1`, and added to a list `deps`. A `Faculty` object `faculty_sm` is created with `deps` and the name "CM". The code then defines two routes: a default route returning the `index.html` template with `faculty_sm` as a parameter, and a route `/new_dep` that handles POST requests. The `adding()` function retrieves parameters from the URL (`name`, `floor`, `students`, `zavkaf`) and creates a new `Department` object, which is then appended to the `departments` list of the `faculty_sm` object.

The screenshot shows a Notepad++ window with the file "myserv.py" open. The code is a Python script for a web application using Flask. It defines a Faculty class with methods for adding and deleting departments. The script uses Jinja templates for rendering HTML pages.

```
25     self.departments = departments
26     self.decan = decan
27
28     def __str__(self):
29         return("Название: ",self.name,
30               " Кафедры:",self.departments,
31               " Декан:",self.decan)
32
33     dep_sm10 = Department("CM-10",7,"Котиев Г.О",50,"sm10.jpg")
34     dep_sm9 = Department("CM-9",3,"Горелов В.А",40,"sm9.jpg")
35     dep_sm1 = Department("CM-1",4,"Завкаф И.И",60,"sm1.jpg")
36
37     deps = [dep_sm1,dep_sm9,dep_sm10]
38
39     faculty_sm = Faculty("CM",deps,"Калугин В.Т.")
40
41     @app.route("/")
42     def hello_world():
43         return render_template('index.html',faculty = faculty_sm)
44
45     @app.route("/new_dep")
46     def adding():
47         name = request.args.get('name')
48         floor = request.args.get('floor')
49         students = request.args.get('students')
50         zavkaf = request.args.get('zavkaf')
51         new_dep = Department(name,floor,zavkaf,students,"sm9.jpg")
52         faculty_sm.departments.append(new_dep)
53         return "Добавил"
54
55     @app.route("/delete")
56     def deleting():
57         name = request.args.get('name')
58         c = 0
59         for d in faculty_sm.departments:
60             if d.name == name:
61                 del faculty_sm.departments[c]
62                 return "Удалил" + d.name
63             else:
64                 c = c + 1
65         return "Не нашёл такую кафедру"
66
```



The screenshot shows a Notepad++ window displaying an HTML file named index.html. The file content is as follows:

```
1 <html>
2   <head>
3     <title>Что будет сверху</title>
4   </head>
5   <body>
6     <h1>Привет, я сервер</h1>
7     <div>Факультет: {{faculty.name}}</div>
8     <div>Наш декан: {{faculty.decan}}</div>
9     <div>Наши кафедры:</div>
10    {% for d in faculty.departments: %}
11      <div>{{d.name}}</div>
12      <div>{{d.zavkaf}}</div>
13      <div>{{d.floor}}</div>
14      
15    {% endfor %}
16    <br>
17  </body>
18 </html>
```