

Московский государственный технический  
университет им. Н.Э. Баумана

Кафедра СМ10 «Колесные машины»

Отчет по лабораторным работам №1-4.

«Разработка WEB приложений»

Студент: Швецов А.С.  
Группа: СМ 10-61Б  
Преподаватель: Гапанюк Ю.Е.

Москва, 2023 г.

## Оглавление

Лабораторная работа №1 .....	3
Лабораторная работа №2 .....	4
Лабораторная работа №3 .....	6
Лабораторная работа №4 .....	8

## 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 предложено написать программу для нахождения корней биквадратного уравнения. Листинг этой программы представлен далее:

```
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")
```

## Лабораторная работа №2

В ходе второй лабораторной работы были изучены основы использования фреймворка Flask в языке программирования Python. 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 Flaskfrom flask import request  
  
app = Flask(__name__)  
@app.route("/")def hello_world():  
    return "<p>Hello, World!</p>"  
@app.route("/test")def test():  
    return "<p>Test</p>"  
@app.route("/test/aboba")def test_aboba():  
    return "<p>Test_ABOBUS</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)
return "<p>Сумма, {}</p>".format(a+b)
@app.route("/calc/scale/<a>/<b>")def calc_scale(a,b):
    a = int(a)
    b = int(b)
    return "<p>Умножение, {}</p>".format(a*b)
@app.route("/calc/div/<a>/<b>")def calc_div(a,b):
    a = int(a)
    b = int(b)
    return "<p>Деление, {}</p>".format(a/b)
@app.route("/calc/sub/")def calc_sub():
    args_dict = request.args
    a = float(args_dict["a"])
    b = float(args_dict["b"])
    return "<p>Вычитание, {}</p>".format(a-b)
@app.route("/calc/food")def food():
    args_dict = request.args
    p = args_dict["первое"]
    f = 0
    if p == "суп":
        f = 100
    elif p == 'бо':
        f = 350
    elif p == 'щи':
        f = 55
    b = args_dict["второе"]
    v = 0
    if b == "котлета":
        v = 100
    elif b == 'бризоль':
        v = 150
    elif b == 'гуляш':
        v = 195
    v = int(v)
    f = int(f)
    return "Сумма за {} и {} = {}".format(p,b,f+v)
@app.route("/pic/<name>")def pic(name):
    return '<img src ="http://127.0.0.1:5000/static/{}.jpg">'.format(name)

```

### Лабораторная работа №3

Третья лабораторная работа по Python была посвящена освоению работы с веб-страницами. Были изучены основные принципы разметки HTML, стилизации элементов страницы с помощью CSS и создания интерактивных элементов с помощью JavaScript. Применены полученные знания при разработке простой веб-страницы, а также написаны и использованы файлы index.html.txt, script.js и style.css для продвинутой работы с web-страницами.

Листинг кода из файла index.html.txt:

```
<!doctype html>
<html>
  <head>
    <link rel="stylesheet" href="style.css">
    <script src="script.js"></script>
    <title>Лабораторная 3 Компы</title>
  </head>
  <body>
    <div class = "myhead">
      <h1>Компьютер</h1>
    </div>
    <div class = "myhead">
      <h2>Леха</h2>
    </div>
    <div id="otv_face">
      Дуиай:
      
    </div>
    <div>
      Операционные системы:
      <table id="mytable">
        <tr>
          <th>Название</th>
          <th>Лого</th>
          <th>Год выпуска</th>
          <th>Цена</th>
        </tr>
        <tr>
          <td> Windows10</td>
          <td> </td>
```

```

                <td> 2012 </td>
                <td> 50 </td>
            </tr>
            <tr>
                <td> Linux</td>
                <td> </td>
                <td> 1998 </td>
                <td> 0 </td>
            </tr>
            <tr>
                <td> macOS</td>
                <td> </td>
                <td> 2005 </td>
                <td> 100 </td>
            </tr>
        </table>
    </div>
</div>

Регистрация:
<br>
<label for "name">
Имя
</label>
<input id="in_name" type="text" name="name" value=""><br>
<label for "age">
Возраст
</label>
<input id="in_age" type="number" name="age" value=""><br>
<label for "name">
Пол
</label>
<input id="in_sex_m" type="radio" name="sex" value="male" />
М
<input id="in_sex_f" type="radio" name="sex" value="female" />
ж
<br>
<label for "name">
Важный
</label>
<input type="checkbox" name="otv" value="Важный?" /><br>
<button type="button" onclick="myfunc()">
Зарегистрироваться</button><br>
<a href="#otv_face">Наверх</a><br>
<a href="https://www.chess.com/home">Ну давай</a>
</div>
</body>
</html>

```

## 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 OS():
    def __init__(self, name, year, price, image):
        self.name = name
        self.year = year
        self.price = price
        self.image = image

    def __str__(self):
        return ("Название: ", name,
                " Год выхода:", year,
                " Цена:", price,
                " Логотип:", image,)

class Comp():
    def __init__(self, name, os, country):
        self.name = name
        self.os = os
        self.country = country

    def __str__(self):
        return ("Название: ", name,
                " Операционные системы:", os,
                " Страна:", country)
```



```

windows = OS("Windows 10",2012,50,"windows.png")
linux = OS("Linux",1995,40,"Linux.png")
macOS = OS("macOS",2000,60,"macOS.jpg")

oss = [windows,linux,macOS]

comp_sm = Comp("ЗИЛ-130",oss,"Россия")
@app.route("/")def hello_world():
    return render_template('index.html',comp = comp_sm)
@app.route("/new_os")def adding():
    name = request.args.get('name')
    year = request.args.get('year')
    price = request.args.get('price')
    new_dep = OS(name,year,price,"EZY.jpg")
    comp_sm.os.append(new_dep)
    return "Добавил"
@app.route("/delete")def deleting():
    name = request.args.get('name')
    c = 0
    for d in comp_sm.os:
        if d.name == name:
            del comp_sm.os[c]
            return "Удалил " + d.name
        else:
            c = c + 1
    return "Не нашёл"
@app.route("/change")def changing():
    name = request.args.get('name')
    price = request.args.get('price')
    c = 0
    for d in comp_sm.os:
        if d.name == name:
            comp_sm.os[c].price = price
            comp_sm.os[c].name = "Changed"
            return "Поменял цену у " + d.name
        else:
            c = c + 1
    return "Не нашёл"

```