



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 лабораторной работе

По курсу «Разработка WEB-приложений»

Студент группы СМ10-61Б

(Подпись, дата)

Н. С. Мартыненко

2023 г.

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

```
import math
print("Напишите, пожалуйста, первый коэффициент")
a = int(input())
print("Напишите, пожалуйста, второй коэффициент")
b = int(input())
print("Напишите, пожалуйста, третий коэффициент")
c = int(input())
print("Вы ввели ",a,b,c)
D = b*b - 4*a*c
if D >= 0:
    x1 = (-b + math.sqrt(D))/(2 * a)
    x2 = (-b - math.sqrt(D))/(2 * a)
    if x1 < 0 and x2 < 0:
        print ("Нет корней")
    elif x1 < 0 and x2 > 0:
        y1 = - math.sqrt(x2)
        y2 = math.sqrt(x2)
        print("Уравнение имеет 2 корня:",y1,y2)
    elif x1 > 0 and x2 < 0:
        y1 = - math.sqrt(x1)
        y2 = math.sqrt(x1)
```

```

    print("y1 =", y1, " y2 =", y2)
else:
    y1 = -math.sqrt(x1)
    y2 = math.sqrt(x1)
    y3 = -math.sqrt(x2)
    y4 = math.sqrt(x2)
    print("y1 =",y1," y2 =",y2," y3 =",y3," y4 =",y4)
elif D == 0:
    x1 = -b/(2*a)
    if x1 > 0:
        y1 = -math.sqrt(x2)
        y2 = math.sqrt(x2)
    elif x1 < 0:
        print("Нет корней")
    else:
        ('Один корень: x1 = 0')
    print("y1 =",y1,"y2 =",y2)
else:
    print("Нет корней")

```

2. 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 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>Nikita Martynenko<p>"
@app.route("/test/test2")
def test2():
    return "<p>Sleva<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)
```

3. Лабораторная работа №3

Третья лабораторная работа 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>

        <title> Это заголовок </title>

    </head>

    <body>

        <p> Это содержимое
```

```
        <strong> стронг </strong>
    <p>
        вот такое вот </p>
    </body>
</html>
```

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"
}
```

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;
}
```

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__(self):

        return("Название: ",self.name,

               " Кафедры:",self.departments,

               " Декан:",self.decan)


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

```
deps = [dep_sm1,dep_sm9,dep_sm10]
```

```
faculty_sm = Faculty("СМ",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"
```