

Московский государственный технический
университет им. Н.Э. Баумана

Кафедра СМ10 «Колесные машины»

Отчет по лабораторным работам №1-4.

«Разработка WEB приложений»

Студент: Белокопытов П. А.

Группа: СМ 10-61Б

Преподаватель: Гапанюк Ю.Е.

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("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")

```

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>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)

```

3 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>

    <title> Это заголовок </title>

  </head>

  <body>

    <p> Это содержимое

      <strong> стронг </strong>

    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 Student(object):
```

```
    def __init__(self, id, fio, average_mark, image):
```

```
        self.id = id
```

```
        self.fio = fio
```

```
        self.average_mark = average_mark
```

```
        self.image = image
```

```
    def __str__(self, id, fio, average_mark, image):
```

```
        return("Имя:",fio,
```

```
               "ID:",id,
```

```
               "Средний бал:",average_mark,
```

```
"Фотография:",image)
```

```
class Group(object):
```

```
    def __init__(self, id, number, students):
```

```
        self.id = id
```

```
        self.number = number
```

```
        self.students = students
```

```
    def __str__(self, id, number):
```

```
        return("ID:",id,
```

```
               "Номер группы:",number)
```

```
st1 = Student(1,"Зинитулен Г.О",5.0,"oGVnnPC5RaU.jpg")
```

```
st2 = Student(2,"Сафулин К.Ф",3.0,"person1941267519.jpg")
```

```
st3 = Student(3,"Татарский Т.Т",4.0,"w2QcTRfLzy8.jpg")
```

```
sts = [st1,st2,st3]
```

```
gr_sm = Group("1","СМ10_61Б",sts)
```

```
@app.route("/")
```

```
def hello_world():
```

```
    return render_template('index.html',group = gr_sm)
```

```
@app.route("/new_st")
```

```
def adding():
```

```
    id = request.args.get('id')
```

```
    fio = request.args.get('fio')
```

```
    average_mark = request.args.get('average_mark')
```

```
    new_st = Student(id,fio,average_mark,"w2QcTRfLzy8.jpg")
```

```
gr_sm.students.append(new_st)
return "Добавил"
```

```
<html>
<head>
  <title>Группы МГТУ</title>
</head>
<body>
  <h1>Самая полная база данных студентов и групп в МГТУ им.
Баумана</h1>
  <div>Номер интересующей вас группы: {{group.number}}</div>
  <div>ID группы: {{group.id}}</div>
  <div>Студенты группы: </div>
  {% for d in group.students: %}
  <div>{{d.id}} </div>
  <div>{{d.fio}} </div>
  <div>{{d.average_mark}} </div>
  
  {% endfor %}
  <br>
</body>
</html>
```