**Московский государственный технический**

**университет им. Н.Э. Баумана**

Факультет «Информатика и системы управления»

Кафедра ИУ5 «Системы обработки информации и управления»

Курс «Парадигмы и конструкции языков программирования»

Отчет по лабораторной работе №1

«Основные конструкции языка Python.»

|  |  |  |
| --- | --- | --- |
| Выполнил: |  | Проверил: |
| студент группы ИУ5-32Б |  | преподаватель каф. ИУ5 |
| Фролов Илья |  |  |
| Подпись и дата: |  | Подпись и дата: |

Москва, 2024 г.

### Задание:

Разработать программу для решения [квадратного уравнения.](https://ru.wikipedia.org/wiki/%D0%9A%D0%B2%D0%B0%D0%B4%D1%80%D0%B0%D1%82%D0%BD%D0%BE%D0%B5_%D1%83%D1%80%D0%B0%D0%B2%D0%BD%D0%B5%D0%BD%D0%B8%D0%B5)

1. Программа должна быть разработана в виде консольного приложения на языке Python.
2. Программа осуществляет ввод с клавиатуры коэффициентов А, В, С, вычисляет дискриминант и ДЕЙСТВИТЕЛЬНЫЕ корни уравнения (в зависимости от дискриминанта).
3. Коэффициенты А, В, С могут быть заданы в виде параметров командной строки ( [вариант задания параметров приведен в конце файла с примером кода](https://github.com/ugapanyuk/BKIT_2022/blob/main/code/lab1_code) ). Если они не заданы, то вводятся с клавиатуры в соответствии с пунктом 2. [Описание работы с параметрами командной строки.](https://realpython.com/python-command-line-arguments/#the-command-line-interface)
4. Если коэффициент А, В, С введен или задан в командной строке некорректно, то необходимо проигнорировать некорректное значение и вводить коэффициент повторно пока коэффициент не будет введен корректно. Корректно заданный коэффициент - это коэффициент, значение которого может быть без ошибок преобразовано в действительное число.

Для реализации данной задачи будем использовать WEB сайт с Клиент-Серверной архитектурой. FrontEnd будем писать на VueJS, BackEnd – на NodeJS + ExpressJS.

### Текст программы:

Клиент:  
 HomePage.vue:  
<template>  
<div class="main">  
 <h2>Решение Квадратного Уравнения</h2>  
 <form class="mainForm">  
 <input type="text" class="Koef" id="KoefA" placeholder="A">  
 <label>X^2 + </label>  
 <input type="text" class="Koef" id="KoefB" placeholder="B">  
 <label>X + </label>  
 <input type="text" class="Koef" id="KoefC" placeholder="C">  
 <label>= 0</label>  
 </form>  
 <h1></h1>  
 <button @click="() => {calculate()}"><h3>Расчитать</h3></button>  
 <h1></h1>  
 <h1 id="EnterEq"></h1>  
 <h1></h1>  
 <h2 id="Disc"></h2>  
 <h1></h1>  
 <h2 id="countRoots"></h2>  
 <h1></h1>  
 <h2 id="Roots"></h2>  
 <h1></h1>  
 </div>  
</template>  
  
<script>  
import axios from 'axios'  
  
export default {  
 name: 'HomePage',  
 methods: {  
 calculate() {  
 let A = document.getElementById("KoefA").value;  
 let B = document.getElementById("KoefB").value;  
 let C = document.getElementById("KoefC").value;  
 const data = {A: A, B: B, C: C};  
 axios.post("http://localhost:3001/calculate/", data).then((response) => {  
 const data = response.data;  
  
 const EnterEq = document.getElementById("EnterEq");  
 let enterText = "Вы ввели: " + data['A'] + "X^2 ";  
 if(data['B'] < 0) {  
 enterText += data['B'] + "X ";  
 } else {  
 enterText += " +" + data['B'] + "X ";  
 }  
  
 if(data['C'] < 0) {  
 enterText += data['C'];  
 } else {  
 enterText += " +" + data['C'];  
 }  
  
 enterText += " = 0";  
  
 EnterEq.textContent = enterText;  
  
 const Disc = document.getElementById("Disc");  
 if(data['D']) {  
 Disc.textContent = "Квадратное уравнение, его Дискриминант равен: " + data['D'];  
 } else {  
 Disc.textContent = "";  
 }  
  
 const countRoots = document.getElementById("countRoots");  
 const Roots = document.getElementById("Roots");  
  
 if(data['ManyRoot']) {  
 countRoots.textContent = "Уравнение имеет бесконечное количество корней";  
 Roots.textContent = "X∈R";  
 } else {  
 if(data['ZeroRoot']) {  
 countRoots.textContent = "Уравнение не имеет корней";  
 Roots.textContent = "X∈Ø";  
 } else {  
 if(data['OneRoot']) {  
 countRoots.textContent = "Уравнение имеет единственный корень";  
 Roots.textContent = "X = " + data['X'];  
 } else {  
 if(data['TwoRoot']) {  
 countRoots.textContent = "Уравнение имеет два корня";  
 Roots.textContent = "X1 = " + data['X1'] + "; X2 = " + data['X2'];  
 } else {  
 countRoots.textContent = "";  
 Roots.textContent = "";  
 }  
 }  
 }  
 }  
 });  
 }  
 }  
};  
</script>  
  
<style>  
div.main {  
 font-family: system-ui;  
 color: white;  
 margin: left;  
 text-align: center;  
}  
body {  
 background: #f06d06;  
}  
  
.Koef {  
 width: 23px;  
 text-align: center;  
}  
</style>

Сервер:

index.js:

const express = require("express");  
const app = express();  
const port = 3001;  
const cors = require("cors");  
  
app.use(express.json());  
app.use(cors());  
  
const calcRouter = require("./routes/calculate.js");  
app.use("/calculate", calcRouter);  
  
app.listen(port, () => {  
 console.log("Server is running on port " + port);  
});

routes/calculate.js

const express = require("express");  
const router = express.Router();  
  
router.post("/", async (req, res) => {  
 let A = req.body['A'];  
 let B = req.body['B'];  
 let C = req.body['C'];  
 if(A == null || A == NaN || A == "" || A == " ") {  
 A = 1;  
 }  
 if(B == null || B == NaN || B == "" || B == " ") {  
 B = 1;  
 }  
 if(C == null || C == NaN || C == "" || C == " ") {  
 C = 0;  
 }  
 if(A == 0) {  
 if(B == 0) {  
 if(C == 0) {  
 res.json({A: A, B: B, C: C, ManyRoot: true});  
 } else {  
 res.json({A: A, B: B, C: C, ZeroRoot: true});  
 }  
 } else {  
 res.json({A: A, B: B, C: C, X: -C/B, OneRoot: true});  
 }  
 } else {  
 const D = B \* B - 4 \* A \* C;  
  
 if(D < 0) {  
 res.json({A: A, B: B, C: C, D: D, ZeroRoot: true});  
 }  
 if(D == 0) {  
 Root = B/(-2.0\*A);  
 res.json({A: A, B: B, C: C, D: 0, OneRoot: true, X: Root});  
 }  
 if(D > 0) {  
 let Root1 = (B - Math.sqrt(D))/(-2.0\*A);  
 let Root2 = (B + Math.sqrt(D))/(-2.0\*A);  
 res.json({A: A, B: B, C: C, D: D, TwoRoot: true, X1: Math.min(Root1, Root2), X2: Math.max(Root1, Root2)});  
 }  
 }  
 return res;  
});  
  
module.exports = router;

### Результаты исполнения:



