

**Московский государственный технический
университет им. Н.Э. Баумана**

Факультет «Информатика и системы управления»

Кафедра ИУ5 «Системы обработки информации и управления»

Курс «Парадигмы и конструкция языков программирования»

Отчет по лабораторной работе №1-3

«Основные конструкции языка Python»

Выполнил:

студент группы ИУ5-36Б

Каверина С.Г

Подпись и дата:

Проверил:

преподаватель каф. ИУ5

Нардид А. Н.

Подпись и дата:

Цель лабораторной работы:

1. Программа должна быть разработана в виде консольного приложения на языке Python.
2. Программа осуществляет ввод с клавиатуры коэффициентов A, B, C, вычисляет дискриминант и ДЕЙСТВИТЕЛЬНЫЕ корни уравнения (в зависимости от дискриминанта).
3. Коэффициенты A, B, C могут быть заданы в виде параметров командной строки (вариант задания параметров приведен в конце файла с примером кода). Если они не заданы, то вводятся с клавиатуры в соответствии с пунктом 2. Описание работы с параметрами командной строки.
4. Если коэффициент A, B, C 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.
5. Дополнительное задание 1 (*). Разработайте две программы на языке Python - одну с применением процедурной парадигмы, а другую с применением объектно-ориентированной парадигмы.
6. Дополнительное задание 2 (*). Разработайте две программы - одну на языке Python, а другую на любом другом языке программирования (кроме C++).

Текст программы:

```
1  import sys
2  import math
3
4  def get_coef(index, prompt):
5      try:
6          # Попробуем прочесть коэффициент из командной строки
7          coef_str = sys.argv[index]
8      except IndexError:
9          # Если параметр отсутствует в командной строке, вводим с клавиатуры
10         print(prompt)
11         coef_str = input()
12
13     try:
14         # Переводим строку в действительное число
15         coef = float(coef_str)
16     except ValueError:
17         # Обрабатываем ошибку, если строка не является числом
18         print("Ошибка: введено некорректное значение. Попробуйте снова.")
19         sys.exit(1)
20
21     return coef
22
23 def get_roots(a, b, c):
24     result = []
25     if a == 0:
26         if b == 0:
27             if c == 0:
28                 print("Уравнение имеет бесконечно много решений.")
29             else:
30                 print("Уравнение не имеет решений.")
```

```

28         print("Уравнение имеет бесконечно много решений.")
29     else:
30         print("Уравнение не имеет решений.")
31     return result
32 else:
33     # Линейное уравнение  $bx + c = 0$ 
34     root = -c / b
35     result.append(root)
36     return result
37
38 # Квадратное уравнение
39  $D = b^2 - 4 * a * c$ 
40 if D == 0.0:
41     root = -b / (2.0 * a)
42     result.append(root)
43 elif D > 0.0:
44     sqD = math.sqrt(D)
45     root1 = (-b + sqD) / (2.0 * a)
46     root2 = (-b - sqD) / (2.0 * a)
47     result.append(root1)
48     result.append(root2)
49     return result
50
51 def main():
52     a = get_coef(1, 'Введите коэффициент A:')
53     b = get_coef(2, 'Введите коэффициент B:')
54     c = get_coef(3, 'Введите коэффициент C:')
55     # Вычисление корней
56     roots = get_roots(a, b, c)
57     # Вывод корней
58
59     sqD = math.sqrt(D)
60     root1 = (-b + sqD) / (2.0 * a)
61     root2 = (-b - sqD) / (2.0 * a)
62     result.append(root1)
63     result.append(root2)
64     return result
65
66 def main():
67     a = get_coef(1, 'Введите коэффициент A:')
68     b = get_coef(2, 'Введите коэффициент B:')
69     c = get_coef(3, 'Введите коэффициент C:')
70     # Вычисление корней
71     roots = get_roots(a, b, c)
72     # Вывод корней
73     len_roots = len(roots)
74     if len_roots == 0:
75         print('Нет корней')
76     elif len_roots == 1:
77         print('Один корень: {}'.format(roots[0]))
78     elif len_roots == 2:
79         print('Два корня: {} и {}'.format(roots[0], roots[1]))
80
81 # Если сценарий запущен из командной строки
82 if __name__ == "__main__":
83     main()

```

Текст дополнительного задания на языке Java Script:

```
const readline = require('readline');
```

```

function getCoef(index, prompt) {
  if (process.argv.length > index) {
    const coefStr = process.argv[index];
    const coef = parseFloat(coefStr);
    if (isNaN(coef)) {
      console.error("Error: Invalid value entered. Please try again.");
      process.exit(1);
    }
    return coef;
  } else {
    return new Promise((resolve) => {
      const rl = readline.createInterface({
        input: process.stdin,
        output: process.stdout
      });
      rl.question(prompt, (input) => {
        const coef = parseFloat(input);
        if (isNaN(coef)) {
          console.error("Error: Invalid value entered. Please try again.");
          process.exit(1);
        }
        rl.close();
        resolve(coef);
      });
    });
  }
}

```

```

function getRoots(a, b, c) {
  const result = [];
  if (a === 0) {
    if (b === 0) {
      if (c === 0) {
        console.log("The equation has infinitely many solutions.");
      } else {
        console.log("The equation has no solutions.");
      }
    }
    return result;
  }
  // линейное уравнение bx+c=0
  const root = -c / b;
  result.push(root);
  return result;
}

```

```

// Дискриминант
const D = b * b - 4 * a * c;
if (D === 0) {
  const root = -b / (2 * a);
  result.push(root);
} else if (D > 0) {
  const sqD = Math.sqrt(D);
  const root1 = (-b + sqD) / (2 * a);
  const root2 = (-b - sqD) / (2 * a);
  result.push(root1, root2);
}
return result;
}

```

```

async function main() {
  const a = await getCoef(2, "Enter coefficient A: ");

```

```
const b = await getCoef(3, "Enter coefficient B: ");
const c = await getCoef(4, "Enter coefficient C: ");
```

```
// Корни
const roots = getRoots(a, b, c);
```

```
// Корни
const lenRoots = roots.length;
if (lenRoots === 0) {
    console.log("No roots");
} else if (lenRoots === 1) {
    console.log(`One root: ${roots[0].toFixed(2)}`);
} else if (lenRoots === 2) {
    console.log(`Two roots: ${roots[0].toFixed(2)} and ${roots[1].toFixed(2)}`);
}
}
```

```
main();
```

Текст дополнительного задания на языке Golang.

```
package main
```

```
import (
    "fmt"
    "math"
    "os"
    "strconv"
)
```

```
func getCoef(index int, prompt string) float64 {
    if len(os.Args) > index {
        coefStr := os.Args[index]
        coef, err := strconv.ParseFloat(coefStr, 64)
        if err != nil {
            fmt.Println("Ошибка: введено некорректное значение. Попробуйте снова.")
            os.Exit(1)
        }
        return coef
    } else {
        fmt.Print(prompt)
        var coefStr string
        fmt.Scanln(&coefStr)
        coef, err := strconv.ParseFloat(coefStr, 64)
        if err != nil {
            fmt.Println("Ошибка: введено некорректное значение. Попробуйте снова.")
            os.Exit(1)
        }
        return coef
    }
}
```

```
func getRoots(a, b, c float64) []float64 {
    var result []float64
    if a == 0 {
        if b == 0 {
            if c == 0 {
                fmt.Println("Уравнение имеет бесконечно много решений.")
            } else {
                fmt.Println("Уравнение не имеет решений.")
            }
        }
        return result
    }
}
```

```

    }
    // Линейное уравнение  $bx + c = 0$ 
    root := -c / b
    result = append(result, root)
    return result
}

```

```

// Квадратное уравнение
D := b*b - 4*a*c
if D == 0 {
    root := -b / (2 * a)
    result = append(result, root)
} else if D > 0 {
    sqD := math.Sqrt(D)
    root1 := (-b + sqD) / (2 * a)
    root2 := (-b - sqD) / (2 * a)
    result = append(result, root1, root2)
}
return result
}

```

```

func main() {
    a := getCoef(1, "Введите коэффициент A:")
    b := getCoef(2, "Введите коэффициент B:")
    c := getCoef(3, "Введите коэффициент C:")
}

```

```

// Вычисление корней
roots := getRoots(a, b, c)

```

```

// Вывод корней
lenRoots := len(roots)
if lenRoots == 0 {
    fmt.Println("Нет корней")
} else if lenRoots == 1 {
    fmt.Printf("Один корень: %.2f\n", roots[0])
} else if lenRoots == 2 {
    fmt.Printf("Два корня: %.2f и %.2f\n", roots[0], roots[1])
}
}

```

Вывод: Я изучила основные конструкции языка python.