

Московский государственный технический университет
им. Н.Э. Баумана

Факультет “Информатика и системы управления”
Кафедра “Системы обработки информации и управления”



Дисциплина «Парадигмы и конструкции языков программирования»

Отчет по лабораторной работе №1
«Основные конструкции языка Python»

Выполнил:
студент группы ИУ5-36Б
Бойко Артем Андреевич
Проверил:
Гапанюк Юрий Евгеньевич

Москва, 2025

Задание:

Разработать программу для решения биквадратного уравнения.

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++).

Программа на языке Python с применением процедурной парадигмы **lab_proc.py**

```
1  import sys
2  import math
3
4  def get_valid_coefficient(prompt):
5
6      while True:
7          try:
8              value = input(prompt)
9              return float(value)
10         except ValueError:
11             print("Ошибка! Введите действительное число.")
12
13 def solve_biquadratic(a, b, c):
14
15     print(f"\nРешение уравнения: {a}x⁴ + {b}x² + {c} = 0")
16
17     if a == 0:
18         print("Ошибка: коэффициент A не может быть равен 0 для биквадратного уравнения")
19         return []
20
21     D = b**2 - 4*a*c
22     print(f"Дискриминант D = {b}² - 4*{a}*{c} = {D}")
23
24     real_roots = []
25
26     if D < 0:
27         print("Дискриминант отрицательный. Действительных корней нет.")
28     else:
29
30         t1 = (-b + math.sqrt(D)) / (2*a)
31         t2 = (-b - math.sqrt(D)) / (2*a)
```

```

32
33     print(f"Корни для t = x2: t1 = {t1:.4f}, t2 = {t2:.4f}")
34
35     # Находим действительные корни x
36     if t1 >= 0:
37         root1 = math.sqrt(t1)
38         root2 = -math.sqrt(t1)
39         real_roots.extend([root1, root2])
40         print(f"Из t1 = {t1:.4f} получаем корни: x = ±{root1:.4f}")
41
42     if t2 >= 0 and abs(t2 - t1) > 1e-10: # Чтобы избежать дублирования
43         root3 = math.sqrt(t2)
44         root4 = -math.sqrt(t2)
45         real_roots.extend([root3, root4])
46         print(f"Из t2 = {t2:.4f} получаем корни: x = ±{root3:.4f}")
47
48     if not real_roots:
49         print("Нет действительных корней (t1 и t2 отрицательные)")
50
51     return real_roots
52
53 def main():
54     print("Решение биквадратного уравнения вида: Ax4 + Bx2 + C = 0")
55
56     coefficients = []
57     for i in range(3):
58         if i < len(sys.argv) - 1:
59             try:
60                 coefficients.append(float(sys.argv[i + 1]))
61
62             except (ValueError, IndexError):
63                 coefficients.append(None)
64             else:
65                 coefficients.append(None)
66
67     a = coefficients[0] if coefficients[0] is not None else get_valid_coefficient("Введите коэффициент A: ")
68     b = coefficients[1] if coefficients[1] is not None else get_valid_coefficient("Введите коэффициент B: ")
69     c = coefficients[2] if coefficients[2] is not None else get_valid_coefficient("Введите коэффициент C: ")
70
71     roots = solve_biquadratic(a, b, c)
72
73     if roots:
74         print(f"\nдействительные корни уравнения: {sorted(roots)}")
75     else:
76         print("\nУравнение не имеет действительных корней.")
77
78 if __name__ == "__main__":
79     main()

```

Программа на языке Python с применением объектно-ориентированной парадигмы **lab_oop.py**

```
1 import sys
2 import math
3
4 class BiquadraticEquation:
5
6     def __init__(self, a, b, c):
7         self.a = a
8         self.b = b
9         self.c = c
10        self.roots = []
11        self.discriminant = None
12
13    def validate_coefficients(self):
14        if self.a == 0:
15            raise ValueError("Коэффициент A не может быть равен 0")
16
17    def calculate_discriminant(self):
18        self.discriminant = self.b**2 - 4*self.a*self.c
19        return self.discriminant
20
21    def solve(self):
22        """Решение биквадратного уравнения"""
23        self.validate_coefficients()
24        self.calculate_discriminant()
25
26        if self.discriminant < 0:
27            return []
28
29        t1 = (-self.b + math.sqrt(self.discriminant)) / (2*self.a)
30        t2 = (-self.b - math.sqrt(self.discriminant)) / (2*self.a)
31
32        real_roots = []
33
34        if t1 >= 0:
35            root1 = math.sqrt(t1)
36            root2 = -math.sqrt(t1)
37            real_roots.extend([root1, root2])
38
39        if t2 >= 0 and abs(t2 - t1) > 1e-10:
40            root3 = math.sqrt(t2)
41            root4 = -math.sqrt(t2)
42            real_roots.extend([root3, root4])
43
44        self.roots = real_roots
45        return self.roots
46
47    def display_solution(self):
48        print(f"\nРешение уравнения: {self.a}x⁴ + {self.b}x² + {self.c} = 0")
49        print(f"Дискриминант D = {self.b}² - 4*{self.a}*{self.c} = {self.discriminant}")
50
51        if self.discriminant < 0:
52            print("Дискриминант отрицательный. Действительных корней нет.")
53        elif not self.roots:
54            print("Нет действительных корней.")
55        else:
56            print(f"Действительные корни уравнения: {sorted(self.roots)}")
57
58 class InputHandler:
59
60     @staticmethod
```

```
61     def get_valid_coefficient(prompt):
62         while True:
63             try:
64                 value = input(prompt)
65                 return float(value)
66             except ValueError:
67                 print("Ошибка! Введите действительное число.")
68
69     @staticmethod
70     def parse_command_line_args():
71         coefficients = []
72
73         for i in range(3):
74             if i + 1 < len(sys.argv):
75                 try:
76                     coefficients.append(float(sys.argv[i + 1]))
77                 except ValueError:
78                     coefficients.append(None)
79             else:
80                 coefficients.append(None)
81         return coefficients
82
83     def main_oo():
84         print("Решение биквадратного уравнения вида: Ax4 + Bx2 + C = 0")
85
86         # Обработка параметров командной строки
87         coefficients = InputHandler.parse_command_line_args()
88
89         # Получение коэффициентов
90         a = coefficients[0] if coefficients[0] is not None else InputHandler.get_valid_coefficient("Введите коэффициент A: ")
91         b = coefficients[1] if coefficients[1] is not None else InputHandler.get_valid_coefficient("Введите коэффициент B: ")
92         c = coefficients[2] if coefficients[2] is not None else InputHandler.get_valid_coefficient("Введите коэффициент C: ")
93
94         try:
95             # Создание и решение уравнения
96             equation = BiquadraticEquation(a, b, c)
97             equation.solve()
98             equation.display_solution()
99         except ValueError as e:
100             print(f"Ошибка: {e}")
101
102 if __name__ == "__main__":
103     main_oo()
```

Программа на языке JavaScript lab_java.js

```
1  class BiquadraticSolver {
2      constructor() {
3          this.a = 0;
4          this.b = 0;
5          this.c = 0;
6          this.roots = [];
7          this.discriminant = 0;
8      }
9
10     getValidCoefficient(promptText) {
11         let value;
12         while (true) {
13             value = prompt(promptText);
14             if (value === null) {
15                 console.log("Ввод отменен");
16                 process.exit(0);
17             }
18             const num = parseFloat(value);
19             if (!isNaN(num)) {
20                 return num;
21             }
22             console.log("Ошибка! Введите действительное число.");
23         }
24     }
25
26     solve(a, b, c) {
27         this.a = a;
28         this.b = b;
29         this.c = c;
30         this.roots = [];
31
32         if (this.a === 0) {
33             throw new Error("Коэффициент A не может быть равен 0");
34         }
35
36         this.discriminant = Math.pow(this.b, 2) - 4 * this.a * this.c;
37         console.log(`Дискриминант D = ${this.b}^2 - 4*${this.a}*${this.c} = ${this.discriminant}`);
38
39         if (this.discriminant < 0) {
40             console.log("Дискриминант отрицательный. Действительных корней нет.");
41             return this.roots;
42         }
43
44         const t1 = (-this.b + Math.sqrt(this.discriminant)) / (2 * this.a);
45         const t2 = (-this.b - Math.sqrt(this.discriminant)) / (2 * this.a);
46
47         console.log(`Корни для t = x^2: t1 = ${t1}, t2 = ${t2}`);
48
49         if (t1 >= 0) {
50             const root1 = Math.sqrt(t1);
51             const root2 = -Math.sqrt(t1);
52             this.roots.push(root1, root2);
53             console.log(`Из t1 = ${t1} получаем корни: x = ±${root1}`);
54         }
55
56         if (t2 >= 0 && t2 !== t1) {
57             const root3 = Math.sqrt(t2);
58             const root4 = -Math.sqrt(t2);
59             this.roots.push(root3, root4);
60             console.log(`Из t2 = ${t2} получаем корни: x = ±${root3}`);
61     }
```

```

61      }
62
63      if (this.roots.length === 0) {
64          console.log("Нет действительных корней (t1 и t2 отрицательные)");
65      }
66
67      return this.roots;
68  }
69
70  main() {
71      console.log("Решение биквадратного уравнения вида: Ax4 + Bx2 + C = 0");
72
73      const a = this.getValidCoefficient("Введите коэффициент A: ");
74      const b = this.getValidCoefficient("Введите коэффициент B: ");
75      const c = this.getValidCoefficient("Введите коэффициент C: ");
76
77      try {
78          const roots = this.solve(a, b, c);
79          if (roots.length > 0) {
80              console.log("Действительные корни уравнения: ${roots.sort((x, y) => x - y)}");
81          } else {
82              console.log("Уравнение не имеет действительных корней.");
83          }
84      } catch (error) {
85          console.log(`Ошибка: ${error.message}`);
86      }
87  }
88 }
89
90 if (typeof require !== 'undefined' && require.main === module) {
91     const readline = require('readline');
92
93     const rl = readline.createInterface({
94         input: process.stdin,
95         output: process.stdout
96     });
97
98     function question(prompt) {
99         return new Promise((resolve) => {
100             rl.question(prompt, resolve);
101         });
102     }
103
104     async function mainNode() {
105         console.log("Решение биквадратного уравнения вида: Ax4 + Bx2 + C = 0");
106
107         const solver = new BiquadraticSolver();
108
109         function getCoefficient(prompt) {
110             return new Promise(async (resolve) => {
111                 while (true) {
112                     const input = await question(prompt);
113                     const num = parseFloat(input);
114                     if (!isNaN(num)) {
115                         resolve(num);
116                         break;
117                     }
118                     console.log("Ошибка! Введите действительное число.");
119                 }
120             });
121         }
122
123         const a = await getCoefficient("Введите коэффициент A: ");
124         const b = await getCoefficient("Введите коэффициент B: ");
125         const c = await getCoefficient("Введите коэффициент C: ");
126
127         const roots = solver.solve(a, b, c);
128
129         if (roots.length > 0) {
130             console.log(`Действительные корни уравнения: ${roots.sort((x, y) => x - y)}`);
131         } else {
132             console.log("Уравнение не имеет действительных корней.");
133         }
134     }
135
136     mainNode();
137 }

```

```
119      |      |
120      |      }|
121      }|
122
123      const a = await getCoefficient("Введите коэффициент A: ");
124      const b = await getCoefficient("Введите коэффициент B: ");
125      const c = await getCoefficient("Введите коэффициент C: ");
126
127      try {
128          const roots = solver.solve(a, b, c);
129          if (roots.length > 0) {
130              console.log(`Действительные корни уравнения: ${roots.sort((x, y) => x - y)}`);
131          } else {
132              console.log("Уравнение не имеет действительных корней.");
133          }
134      } catch (error) {
135          console.log(`Ошибка: ${error.message}`);
136      }
137
138      rl.close();
139
140
141      mainNode();
142 }
```

Примеры выполнения программ

lab_proc.py

- @artem-hedgehog → /workspaces/ProgLangParadigms_BoikoA_2025/lab_1 (main) \$ python lab_proc.py
Решение биквадратного уравнения вида: $Ax^4 + Bx^2 + C = 0$

Введите коэффициент A: 1

Введите коэффициент B: 0

Введите коэффициент C: -4

Решение уравнения: $1.0x^4 + 0.0x^2 + -4.0 = 0$

Дискриминант D = $0.0^2 - 4*1.0*-4.0 = 16.0$

Корни для t = x²: t1 = 2.0000, t2 = -2.0000

Из t1 = 2.0000 получаем корни: x = ±1.4142

Действительные корни уравнения: [-1.4142135623730951, 1.4142135623730951]

- @artem-hedgehog → /workspaces/ProgLangParadigms_BoikoA_2025/lab_1 (main) \$ python lab_proc.py 1 0
Решение биквадратного уравнения вида: Ax⁴ + Bx² + C = 0

Решение уравнения: $1.0x^4 + 0.0x^2 + -4.0 = 0$

Дискриминант D = $0.0^2 - 4*1.0*-4.0 = 16.0$

Корни для t = x²: t1 = 2.0000, t2 = -2.0000

Из t1 = 2.0000 получаем корни: x = ±1.4142

Действительные корни уравнения: [-1.4142135623730951, 1.4142135623730951]

- @artem-hedgehog → /workspaces/ProgLangParadigms_BoikoA_2025/lab_1 (main) \$ python lab_proc.py
Решение биквадратного уравнения вида: Ax⁴ + Bx² + C = 0

Введите коэффициент A: qwerty

Ошибка! Введите действительное число.

Введите коэффициент A: 208f2-

Ошибка! Введите действительное число.

Введите коэффициент A: 1

Введите коэффициент B: de

Ошибка! Введите действительное число.

Введите коэффициент B: 0

Введите коэффициент C: 23/9

Ошибка! Введите действительное число.

Введите коэффициент C: 4

Решение уравнения: $1.0x^4 + 0.0x^2 + 4.0 = 0$

Дискриминант D = $0.0^2 - 4*1.0*4.0 = -16.0$

Дискриминант отрицательный. Действительных корней нет.

lab_oop.py

```
● @artem-hedgehog →/workspaces/ProgLangParadigms_BoikoA_2025/lab_1 (main) $ python lab_oop.py
Решение биквадратного уравнения вида: Ax4 + Bx2 + C = 0
Введите коэффициент A: 1
Введите коэффициент B: 0
Введите коэффициент C: -4

Решение уравнения: 1.0x4 + 0.0x2 + -4.0 = 0
Дискриминант D = 0.02 - 4*1.0*-4.0 = 16.0
Действительные корни уравнения: [-1.4142135623730951, 1.4142135623730951]
● @artem-hedgehog →/workspaces/ProgLangParadigms_BoikoA_2025/lab_1 (main) $ python lab_oop.py 1 0 -4
Решение биквадратного уравнения вида: Ax4 + Bx2 + C = 0

Решение уравнения: 1.0x4 + 0.0x2 + -4.0 = 0
Дискриминант D = 0.02 - 4*1.0*-4.0 = 16.0
Действительные корни уравнения: [-1.4142135623730951, 1.4142135623730951]
● @artem-hedgehog →/workspaces/ProgLangParadigms_BoikoA_2025/lab_1 (main) $ python lab_oop.py
Решение биквадратного уравнения вида: Ax4 + Bx2 + C = 0
Введите коэффициент A: qwerty
Ошибка! Введите действительное число.
Введите коэффициент A: 2,kf04nmf
Ошибка! Введите действительное число.
Введите коэффициент A: 1
Введите коэффициент B: de
Ошибка! Введите действительное число.
Введите коэффициент B: 0
Введите коэффициент C: 23/9
Ошибка! Введите действительное число.
Введите коэффициент C: 4

Решение уравнения: 1.0x4 + 0.0x2 + 4.0 = 0
Дискриминант D = 0.02 - 4*1.0*4.0 = -16.0
Дискриминант отрицательный. Действительных корней нет.
```

lab_java.js

```
● @artem-hedgehog →/workspaces/ProgLangParadigms_BoikoA_2025/lab_1 (main) $ node lab_java.js
Решение биквадратного уравнения вида: Ax4 + Bx2 + C = 0
Введите коэффициент A: 1
Введите коэффициент B: 0
Введите коэффициент C: -4
Дискриминант D = 02 - 4*1*-4 = 16
Корни для t = x2: t1 = 2, t2 = -2
Из t1 = 2 получаем корни: x = ±1.4142135623730951
Действительные корни уравнения: -1.4142135623730951,1.4142135623730951
● @artem-hedgehog →/workspaces/ProgLangParadigms_BoikoA_2025/lab_1 (main) $ node lab_java.js
Решение биквадратного уравнения вида: Ax4 + Bx2 + C = 0
Введите коэффициент A: qwerty
Ошибка! Введите действительное число.
Введите коэффициент A: d-04mf
Ошибка! Введите действительное число.
Введите коэффициент A: 1
Введите коэффициент B: dev
Ошибка! Введите действительное число.
Введите коэффициент B: 0
Введите коэффициент C: cp3--e
Ошибка! Введите действительное число.
Введите коэффициент C: 4
Дискриминант D = 02 - 4*1*4 = -16
Дискриминант отрицательный. Действительных корней нет.
Уравнение не имеет действительных корней.
```