

Лабораторная работа №1

Программа на Python с применением процедурной парадигмы программирования:

```
import sys

def input_coefficient(coefficient_name):
    while True:
        try:
            coefficient = float(input(f"Введите коэффициент {coefficient_name}: "))
            return coefficient
        except ValueError:
            print("Некорректный ввод. Пожалуйста, введите число.")

def calculate_discriminant(a, b, c):
    return b**2 - 4*a*c

def calculate_roots(a, b, discriminant):
    if discriminant > 0:
        root1 = (-b + discriminant**0.5) / (2*a)
        root2 = (-b - discriminant**0.5) / (2*a)
        return (root1, root2)
    elif discriminant == 0:
        root = -b / (2*a)
        return (root,)
    else:
        return ()

def main():
    if len(sys.argv) == 4:
        try:
            a = float(sys.argv[1])
            b = float(sys.argv[2])
            c = float(sys.argv[3])
        except ValueError:
            print("Некорректный ввод коэффициентов из командной строки.")
            a = input_coefficient('A')
            b = input_coefficient('B')
```

```

        c = input_coefficient('C')
    else:
        a = input_coefficient('A')
        b = input_coefficient('B')
        c = input_coefficient('C')

    discriminant = calculate_discriminant(a, b, c)
    roots = calculate_roots(a, b, discriminant)

    if len(roots) == 0:
        print("Уравнение не имеет действительных корней.")
    else:
        print("Действительные корни уравнения:", ', '.join(str(root) for root in roots))

if __name__ == "__main__":
    main()

```

Программа на Python с применением объектно-ориентированной парадигмы программирования:

```

import sys

class QuadraticEquationSolver:
    def __init__(self, a, b, c):
        self.a = a
        self.b = b
        self.c = c

    def calculate_discriminant(self):
        return self.b**2 - 4*self.a*self.c

    def calculate_roots(self):
        discriminant = self.calculate_discriminant()
        if discriminant > 0:
            root1 = (-self.b + discriminant**0.5) / (2*self.a)
            root2 = (-self.b - discriminant**0.5) / (2*self.a)
            return (root1, root2)
        elif discriminant == 0:
            root = -self.b / (2*self.a)
            return (root,)
        else:
            return ()

```

```

def input_coefficient(coefficient_name):
    while True:
        try:
            coefficient = float(input(f"Введите коэффициент {coefficient_name}: "))
            return coefficient
        except ValueError:
            print("Некорректный ввод. Пожалуйста, введите число.")

def main():
    if len(sys.argv) == 4:
        try:
            a = float(sys.argv[1])
            b = float(sys.argv[2])
            c = float(sys.argv[3])
        except ValueError:
            print("Некорректный ввод коэффициентов из командной строки.")
            a = input_coefficient('A')
            b = input_coefficient('B')
            c = input_coefficient('C')
    else:
        a = input_coefficient('A')
        b = input_coefficient('B')
        c = input_coefficient('C')

    equation_solver = QuadraticEquationSolver(a, b, c)
    roots = equation_solver.calculate_roots()

    if len(roots) == 0:
        print("Уравнение не имеет действительных корней.")
    else:
        print("Действительные корни уравнения:", ', '.join(str(root) for root in roots))

if __name__ == "__main__":
    main()

```

Реализация данной программы на языке C:

```

#include <stdio.h>
#include <stdlib.h>
#include <math.h>

```

```

float input_coefficient(char* coefficient_name) {

```

```

while (1) {
    char input[100];
    float coefficient;
    printf("Введите коэффициент %s: ", coefficient_name);
    fgets(input, sizeof(input), stdin);
    if (sscanf(input, "%f", &coefficient) == 1) {
        return coefficient;
    }
    printf("Некорректный ввод. Пожалуйста, введите число.\n");
}

float calculate_discriminant(float a, float b, float c) {
    return b*b - 4*a*c;
}

void calculate_roots(float a, float b, float discriminant, float* roots) {
    if (discriminant > 0) {
        roots[0] = (-b + sqrt(discriminant)) / (2*a);
        roots[1] = (-b - sqrt(discriminant)) / (2*a);
    } else if (discriminant == 0) {
        roots[0] = -b / (2*a);
    }
}

int main(int argc, char* argv[]) {
    float a, b, c;
    float discriminant;
    float roots[2];

    if (argc == 4) {
        char* endptr;
        a = strtod(argv[1], &endptr);
        if (*endptr != '\0') {
            printf("Некорректный ввод коэффициентов из командной строки.\n");
            a = input_coefficient("A");
            b = input_coefficient("B");
            c = input_coefficient("C");
        } else {
            b = strtod(argv[2], &endptr);
            if (*endptr != '\0') {
                printf("Некорректный ввод коэффициентов из командной строки.\n");

```

```

        a = input_coefficient("A");
        b = input_coefficient("B");
        c = input_coefficient("C");
    } else {
        c = strtod(argv[3], &endptr);
        if (*endptr != '\0') {
            printf("Некорректный ввод коэффициентов из командной строки.\n");
            a = input_coefficient("A");
            b = input_coefficient("B");
            c = input_coefficient("C");
        }
    }
}
} else {
    a = input_coefficient("A");
    b = input_coefficient("B");
    c = input_coefficient("C");
}

discriminant = calculate_discriminant(a, b, c);
calculate_roots(a, b, discriminant, roots);

if (discriminant < 0) {
    printf("Уравнение не имеет действительных корней.\n");
} else {
    if (discriminant == 0) {
        printf("Действительный корень уравнения: %f\n", roots[0]);
    } else {
        printf("Действительные корни уравнения: %f, %f\n", roots[0], roots[1]);
    }
}

return 0;
}

```