《Java》课程实验报告

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 专业名称 | 计算机科学与技术 | 年级 | 2023 | 班级 | 01 |
| 学生姓名 | 颜丙超 | 学号 | 202311000415 | 指导教师 | 梁成 |
| 实验题目 | Java 类设计：正多边形类的实现与测试 二次方程求解类的实现与测试 | | | 提交时间 | 2024.11.10 |

1. 实验目的和要求

实验一

1.掌握Java面向对象编程的基本概念和方法。

2.学会使用Java类来封装数据和操作。

3.实现一个正多边形类，能够计算其周长和面积，并能记录其中心点的坐标。

4.通过控制台输入和输出，测试类的功能。

实验二

1.掌握Java面向对象编程的基本概念和方法。

2.学会使用Java类来封装数据和操作。

3.实现一个二次方程类，能够计算其判别式，并根据判别式的值求解方程的根。

4.通过控制台输入和输出，测试类的功能。

1. 实验环境

Window系统，eclpise软件

三、实验内容及实施

**实验1**

**实验内容**

**1.设计一个名为 RegularPolygon 的类，用于表示正多边形。**

**2.类的属性包括：边数 n、边长 side、中心点坐标 (x, y)。**

**3.提供构造方法，用于初始化对象。**

**4.提供获取和设置属性的方法（getter 和 setter）。**

**5.实现计算正多边形周长和面积的方法。**

**6.在 main 方法中，通过控制台输入正多边形的参数，创建对象并输出其周长、面积和中心点坐标。**

**实验步骤**

**1.设计类结构：**

**确定类的属性：边数 n、边长 side、中心点坐标 (x, y)。确定类的方法：构造方法、getter 和 setter 方法、计算周长和面积的方法。**

**2.编写代码：**

**根据设计，编写 RegularPolygon 类。在 main 方法中，使用 Scanner 类获取用户输入，创建两个 RegularPolygon 对象。输出两个对象的周长、面积和中心点坐标。**

**3.测试代码：**

**运行程序，输入不同的参数，验证输出的正确性**

**实验二**

**1.设计一个名为 QuadraticEquation 的类，用于表示二次方程 ax^2 + bx + c = 0。**

**2.类的属性包括：系数 a、b 和 c。**

**3.提供构造方法，用于初始化对象。**

**4.提供获取系数的方法（getter）。**

**5.实现计算判别式的方法。**

**6.实现求解方程根的方法。**

**7.在 main 方法中，通过控制台输入二次方程的系数，创建对象并输出方程的根。**

**实验步骤：**

**1.设计类结构：**

**确定类的属性：系数 a、b 和 c。**

**确定类的方法：构造方法、getter 方法、计算判别式的方法、求解方程根的方法。**

**2.编写代码：**

**根据设计，编写 QuadraticEquation 类。**

**在 main 方法中，使用 Scanner 类获取用户输入，创建 QuadraticEquation 对象。**

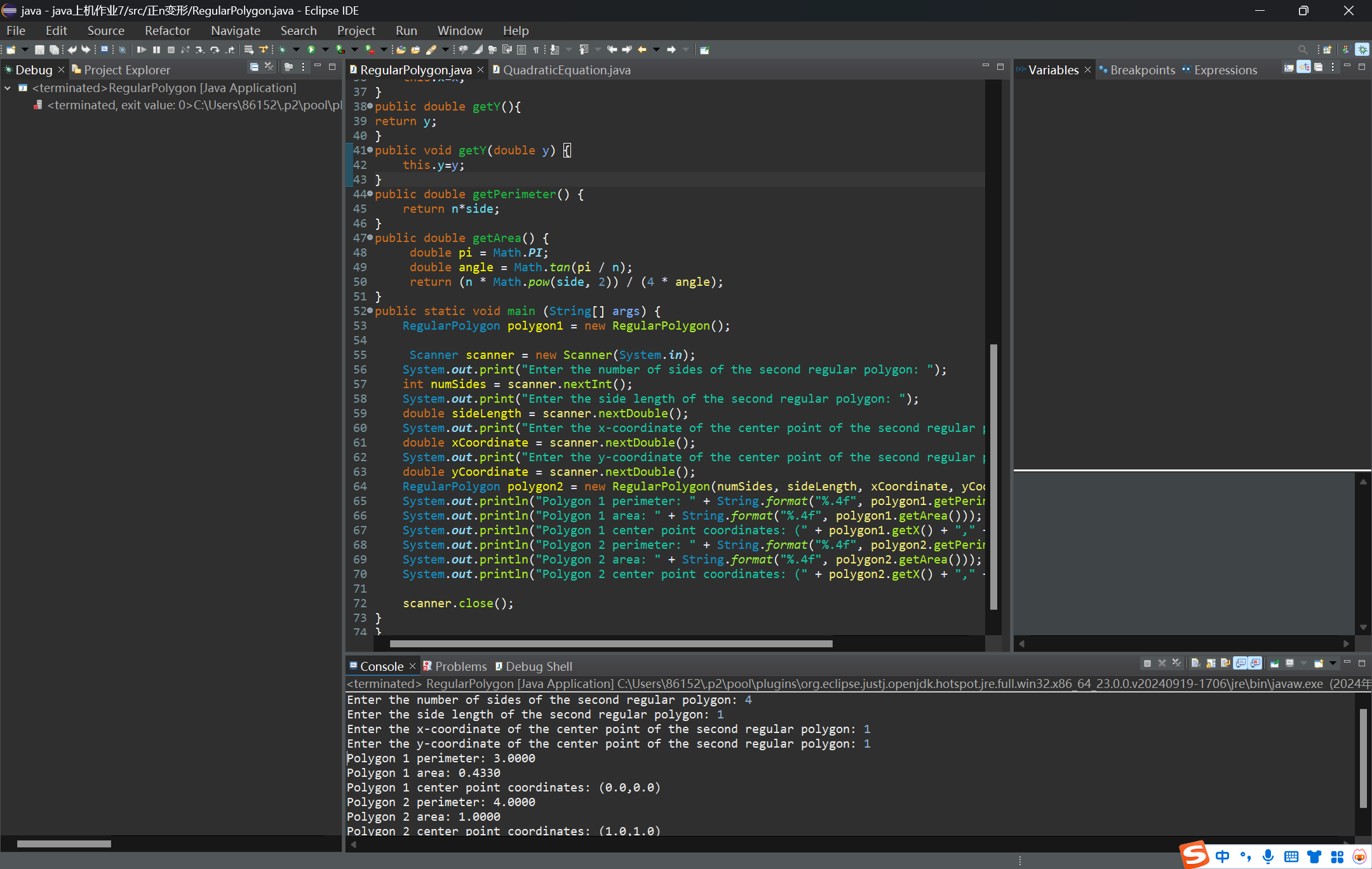
**调用对象的方法计算判别式，并根据判别式的值输出方程的根。**

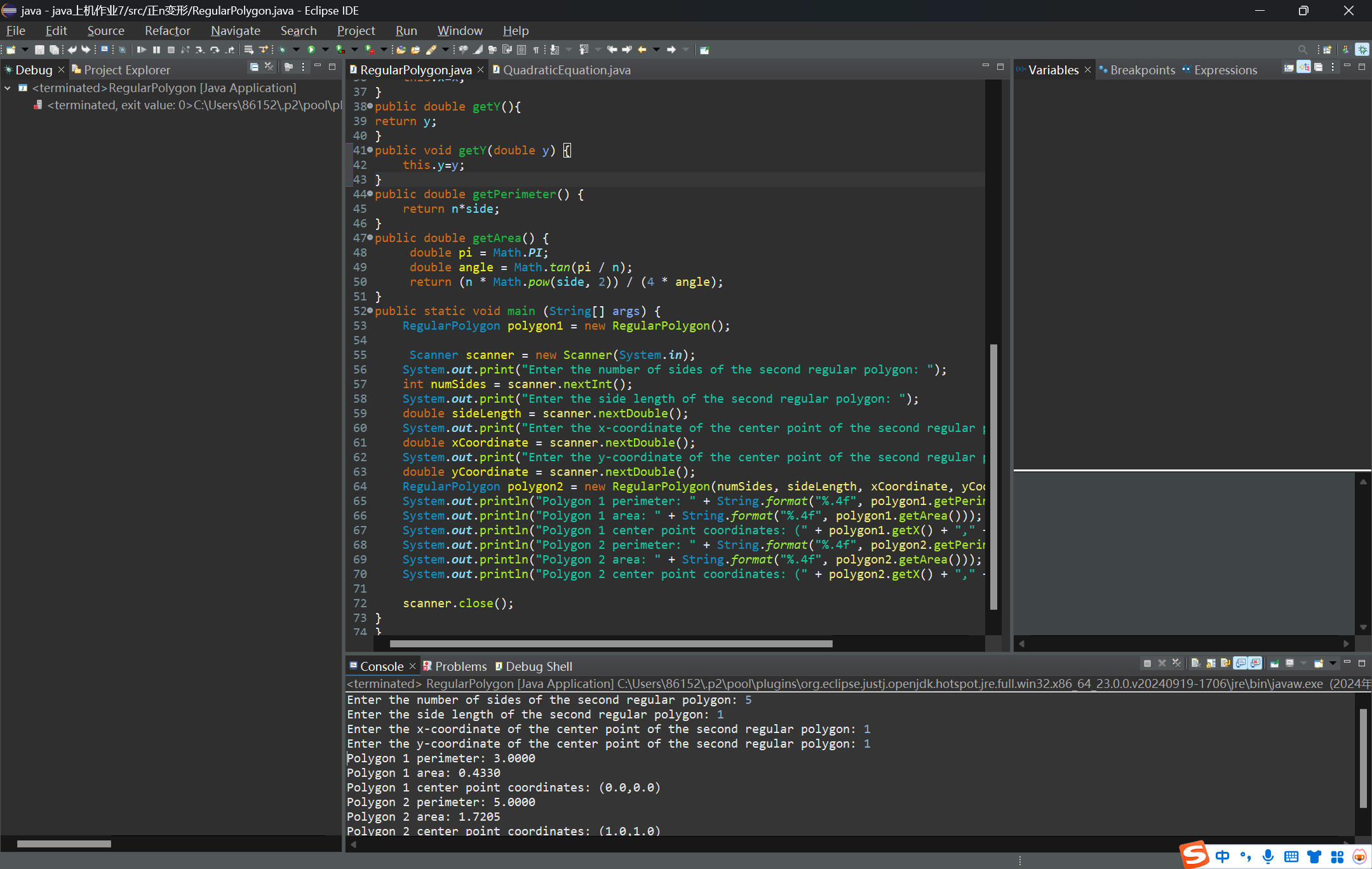
**3.测试代码：**

**运行程序，输入不同的系数，验证输出的正确性。**

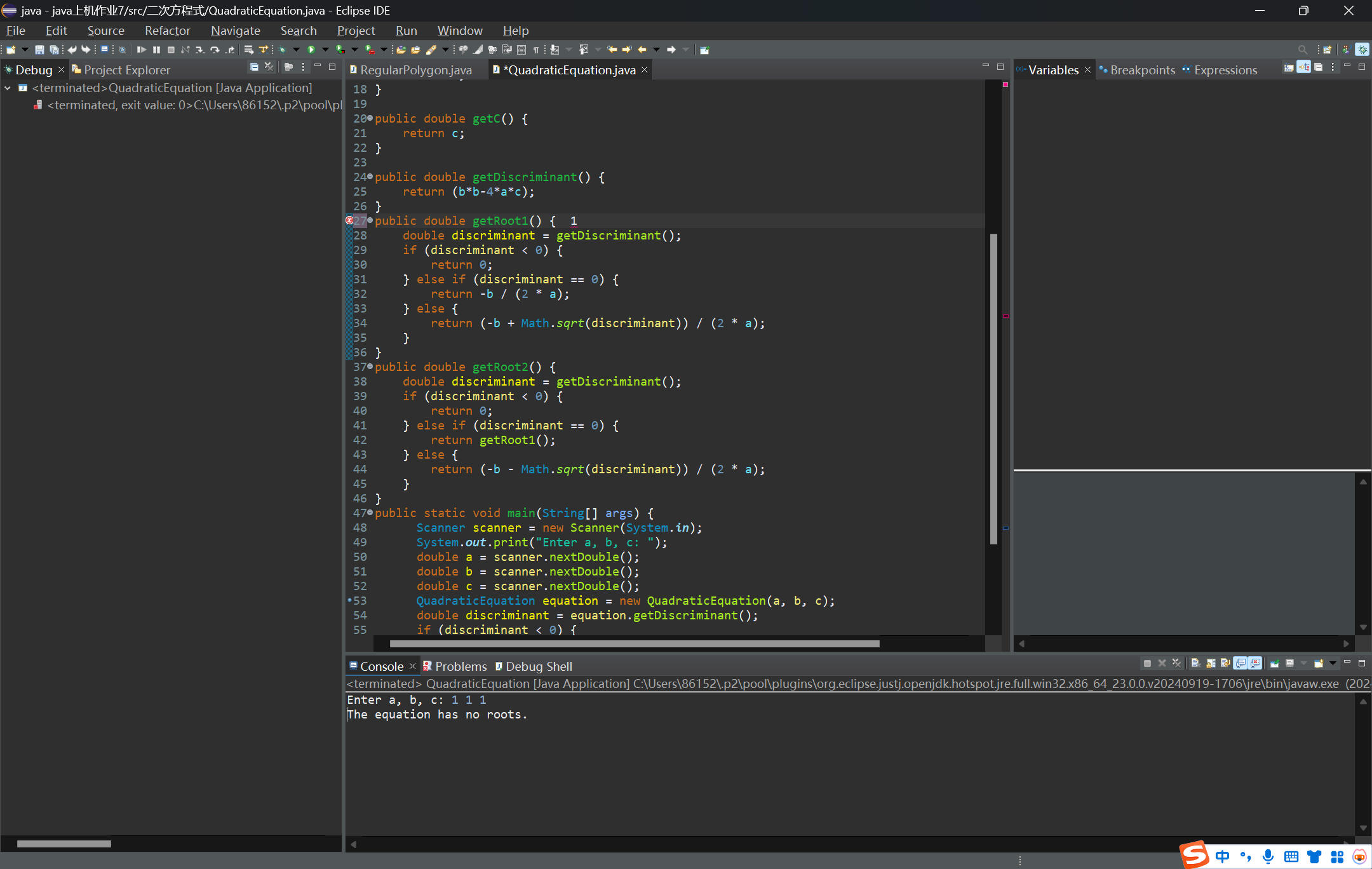
1. **实验结果 (程序的执行结果)**

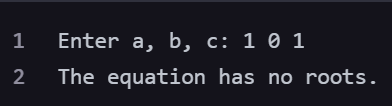
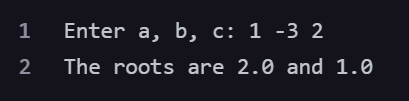
**实验一**





实验二





**五、实验讨论（可选）**

1.本次实验成功实现了正多边形类，并通过控制台输入和输出进行了功能测试。实验结果符合预期，验证了类的正确性和实用性。通过本次实验，加深了对Java面向对象编程的理解，提高了编程实践能力。

2.本次实验成功实现了二次方程类，并通过控制台输入和输出进行了功能测试。实验结果符合预期，验证了类的正确性和实用性。通过本次实验，加深了对Java面向对象编程的理解，提高了编程实践能力。

实验一

01 package 正n变形;

02 import java.util.Scanner;

03 public class RegularPolygon

04 {

05 private int n;

06 private double side;

07 private double x;

08 private double y;

09 public RegularPolygon()

10 {

11 this.n= 3;

12 this.side=1;

13 this.x=0;

14 this.y=0;

15 }

16 public RegularPolygon(int n,double side, double x,double y)

17 {

18 this.n=n;

19 this.side=side;

20 this.x=x;

21 this.y=y;

22 }

23 public int getN()

24 {

25 return n;

26 }

27 public void getN(int n)

28 {

29 this.n=n;

30 }

31 public double getSide()

32 {

33 return side;

34 }

35 public void getSide(double side)

36 {

37 this.side=side;

38 }

39 public double getX()

40 {

41 return x;

42 }

43 public void getX(double x)

44 {

45 this.x=x;

46 }

47 public double getY()

48 {

49 return y;

50 }

51 public void getY(double y)

52 {

53 this.y=y;

54 }

55 public double getPerimeter()

56 {

57 return n\*side;

58 }

59 public double getArea()

60 {

61 double pi = Math.PI;

62 double angle = Math.tan(pi / n);

63 return (n \* Math.pow(side, 2)) / (4 \* angle);

64 }

65 public static void main (String[] args)

66 {

67 RegularPolygon polygon1 = new RegularPolygon();

68

69 Scanner scanner = new Scanner(System.in);

70 System.out.print("Enter the number of sides of the second regular polygon: ");

71 int numSides = scanner.nextInt();

72 System.out.print("Enter the side length of the second regular polygon: ");

73 double sideLength = scanner.nextDouble();

74 System.out.print("Enter the x-coordinate of the center point of the second regular polygon: ");

75 double xCoordinate = scanner.nextDouble();

76 System.out.print("Enter the y-coordinate of the center point of the second regular polygon: ");

77 double yCoordinate = scanner.nextDouble();

78 RegularPolygon polygon2 = new RegularPolygon(numSides, sideLength, xCoordinate, yCoordinate);

79 System.out.println("Polygon 1 perimeter: " + String.format("%.4f", polygon1.getPerimeter()));

80 System.out.println("Polygon 1 area: " + String.format("%.4f", polygon1.getArea()));

81 System.out.println("Polygon 1 center point coordinates: (" + polygon1.getX() + "," + polygon1.getY() + ")");

82 System.out.println("Polygon 2 perimeter: " + String.format("%.4f", polygon2.getPerimeter()));

83 System.out.println("Polygon 2 area: " + String.format("%.4f", polygon2.getArea()));

84 System.out.println("Polygon 2 center point coordinates: (" + polygon2.getX() + "," + polygon2.getY() + ")");

85

86 scanner.close();

87 }

88 }

89

实验二

01 package 二次方程式；

02 import java.util.Scanner;

03

04 public class QuadraticEquation

05 {

06

07 private double a;

08 private double b;

09 private double c;

10

11 public QuadraticEquation(double a,double b,double c)

12 {

13

14 this.a=a;

15

16 this.b=b;

17

18 this.c=c;

19

20 }

21

22 public double getA()

23 {

24

25 return a;

26

27 }

28

29

30

31 public double getB()

32 {

33

34 return b;

35

36 }

37

38

39

40 public double getC()

41 {

42

43 return c;

44

45 }

46

47

48

49 public double getDiscriminant()

50 {

51

52 return (b\*b-4\*a\*c);

53

54 }

55

56 public double getRoot1()

57 {

58

59 double discriminant = getDiscriminant();

60

61 if (discriminant < 0)

62 {

63

64 return 0;

65

66 }

67 else if (discriminant == 0)

68 {

69

70 return -b / (2 \* a);

71

72 }

73 else

74 {

75

76 return (-b + Math.sqrt(discriminant)) / (2 \* a);

77

78 }

79

80 }

81

82 public double getRoot2()

83 {

84

85 double discriminant = getDiscriminant();

86

87 if (discriminant < 0)

88 {

89

90 return 0;

91

92 }

93 else if (discriminant == 0)

94 {

95

96 return getRoot1();

97

98 }

99 else

100 {

101

102 return (-b - Math.sqrt(discriminant)) / (2 \* a);

103

104 }

105

106 }

107

108 public static void main(String[] args)

109 {

110

111 Scanner scanner = new Scanner(System.in);

112

113 System.out.print("Enter a, b, c: ");

114

115 double a = scanner.nextDouble();

116

117 double b = scanner.nextDouble();

118

119 double c = scanner.nextDouble();

120

121 QuadraticEquation equation = new QuadraticEquation(a, b, c);

122

123 double discriminant = equation.getDiscriminant();

124

125 if (discriminant < 0)

126 {

127

128 System.out.println("The equation has no roots.");

129

130 }

131 else if (discriminant == 0)

132 {

133

134 System.out.println("The root is " + equation.getRoot1());

135

136 }

137 else

138 {

139

140 System.out.println("The roots are " + equation.getRoot1() + " and " + equation.getRoot2());

141

142 }

143

144 scanner.close();

145

146 }

147

148 }

149

150