|  |  |
| --- | --- |
| Pedro Henrique Padilha Portella da Cruz | Programação Orientada a Objetos |
| Professora Luciana Zapparolli | FATEC Zona Sul |

LISTA DE EXERCÍCIOS – LP05

1ª) Criar uma Classe chamada Triangulo que contenha como atributos a base e altura de um triângulo. Na classe definir um método para leitura da base e altura, um método construtor e um método para Calcular a área do triângulo (Area=Base \* Altura)/2. No método principal do programa (main) , criar um objeto a partir da classe criada, solicitar e atribuir os valores da base e altura e invocar o método que calcula a área a partir do objeto criado.

import javax.swing.JOptionPane;

public class ex01 {

    public static void main(String[] args) {

        try {

            double base = Integer.parseInt(JOptionPane.showInputDialog("Base do triângulo"));

            double altura = Integer.parseInt(JOptionPane.showInputDialog("Altura do triângulo"));

            Triangulo t = new Triangulo(base, altura);

            double area = t.calcularArea();

            JOptionPane.showMessageDialog(null, "Área do triângulo: " + area);

        } catch(Exception e) {

            JOptionPane.showMessageDialog(null, "Houve um erro, o valor enviado é inválido");

        }

    }

}

Classe Triangulo

public class Triangulo {

    private double base;

    private double altura;

    public Triangulo() {

    }

    public Triangulo(double base, double altura) {

        this.base = base;

        this.altura = altura;

    }

    public double calcularArea() {

        return (this.base \* this.altura) / 2;

    }

    public double getBase() {

        return this.base;

    }

    public void setBase(double base) {

        this.base = base;

    }

    public double getAltura() {

        return this.base;

    }

    public void setAltura(double altura) {

        this.altura = altura;

    }

}

2ª) Criar uma Classe chamada Equacao que contenha como atributos 3 valores inteiros, A, B e C. Na classe definir um método para leitura dos valores, um método construtor e um método para calcular as raízes da equação. R1 = -b+ (raiz de delta) / (2\*a) R2 = -b-(raiz de delta) / (2\*a) Considerar que: - Se delta = 0, as raízes são iguais; - Se delta <0, não existem raízes reais; - Se delta >=, existem 2 raízes diferentes No método principal do programa (main) , criar um objeto a partir da classe criada, solicitar e atribuir os valores e invocar o método que calcula as raízes a partir do objeto criado.

import javax.swing.JOptionPane;

public class ex02 {

    public static void main(String[] args) {

        try {

            Integer a = Integer.parseInt(JOptionPane.showInputDialog(null, "Valor de A:"));

            Integer b = Integer.parseInt(JOptionPane.showInputDialog(null, "Valor de B:"));

            Integer c = Integer.parseInt(JOptionPane.showInputDialog(null, "Valor de C:"));

            Equacao equacao = new Equacao(a, b, c);

            equacao.calcularRaiz();

        } catch(Exception e) {

            JOptionPane.showMessageDialog(null, "Houve um erro, o valor enviado é inválido");

        }

    }

}

Classe Equacao

import javax.swing.JOptionPane;

public class Equacao

{

    private Integer a;

    private Integer b;

    private Integer c;

    public Equacao() { }

    public Equacao(int a, int b, int c) {

        this.a = a;

        this.b = b;

        this.c = c;

    }

    public void calcularRaiz() {

        double delta = Math.pow(this.b, 2) - (4 \* this.a \* this.c);

        double r1 = 0;

        double r2 = 0;

        if(delta == 0) {

            r1 = this.b + (Math.sqrt(delta) / (2 \* a));

            JOptionPane.showMessageDialog(null, "R1 = " + r1);

        } else if(delta > 0) {

            r1 = ((this.b \* -1) + Math.sqrt(delta)) / (2 \* a);

            r2 = ((this.b \* -1) - Math.sqrt(delta)) / (2 \* a);

            JOptionPane.showMessageDialog(null, "R1 = " + r1 + ", R2 = " + r2);

        } else {

            JOptionPane.showMessageDialog(null, "Não existem raizes reais");

        }

    }

    public Integer getA() {

        return this.a;

    }

    public void setA(Integer a) {

        this.a = a;

    }

    public Integer getB() {

        return this.b;

    }

    public void setB(Integer b) {

        this.b = b;

    }

    public Integer getC() {

        return this.c;

    }

    public void setC(Integer c) {

        this.c = c;

    }

}