

Programação Orientada a Objetos I

Profa. Angela Abreu Rosa de Sá, Dra.

Contato: angelaabreu@gmail.com

Horário das Aulas

Terça-feira: 19:10 hs - 20:50 hs

----> Lista de presença

Avaliações

2 Avaliações Oficiais

1000 pontos

4000 pontos

Trabalhos: 2500 pontos

Total: 7500 pontos



MARÇO / 2023 D Q s s





01. Início do Período Letivo

13 a 15. Semana Acadêmica

20 a 21. Carnaval

23. Início das Aulas

MARÇO

ABRIL

07. Paixão de Cristo

10 a 14. Avaliação Oficial - 1º Bimestre

21. Tiradentes

28. Lançamento das notas - 1º Bimestre

MAIO / 2023 s D T Q Q s S



JUNHO

JULHO / 2023						
S	T	Q	Q	S	S	
					1	
3	4	5	6	7	8	
10	11	12	13	14	1 5	
17	18	19	20	21	22	
24	25	26	27	28	29	
31						
	3 10 17 24	3 4 10 11 17 18 24 25	3 4 5 10 11 12 17 18 19 24 25 26	3 4 5 6 10 11 12 13 17 18 19 20 24 25 26 27	S T Q Q S 3 4 5 6 7 10 11 12 13 14 17 18 19 20 21 24 25 26 27 28	

MAIO

01. Dia do Trabalho

01 a 07. Avaliação Oficial - 2º Bimestre

05 a 17. Avaliação Oficial DI

08. Corpus Christi

*Poderá haver atividades ou compensação de horas, exceto em caso de férias ou recessos previstos em acordos sindicais.

JULHO

Material Didático



Ementa

Fundamentos da orientação a objetos

Histórico e introdução à orientação a objetos Conceitos básicos de orientação a objetos Construtores e sobrecarga

Estruturas de programação orientadas a objetos

Estruturas de decisão e controle em Java Estruturas de repetição em Java Reutilização de classes em Java

Exceções, classes abstratas e interfaces

Definição e tratamento de exceções Definição e uso de classes abstratas Definição e uso de interfaces

Aplicações orientadas a objetos

Arrays em Java Strings em Java Coleções e arquivos

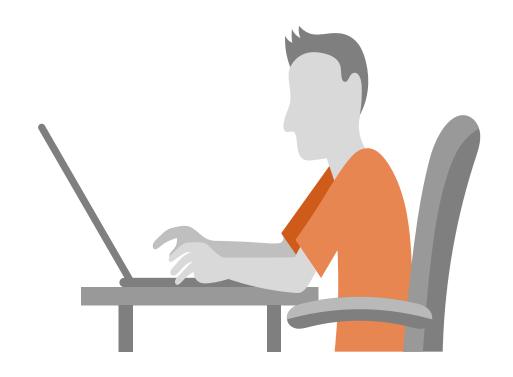
Material Didático



Sumário

Unidade 1 Fundamentos da orientação a objetos			
Seção 1.1 - Histórico e introdução à orientação a objetos	9		
Seção 1.2 - Conceitos básicos de orientação a objetos	22		
Seção 1.3 - Construtores e sobrecarga	37		
Unidade 2 Estruturas de programação orientadas a objetos	59		
Seção 2.1 - Estruturas de decisão e controle em Java	61		
Seção 2.2 - Estruturas de repetição em Java			
Seção 2.3 - Reutilização de classes em Java	93		
Unidade 3 Exceções, classes abstratas e interfaces	111		
Seção 3.1 - Definição e tratamento de exceções	113		
Seção 3.2 - Definição e uso de classes abstratas			
Seção 3.3 - Definição e uso de interfaces			
Unidade 4 Aplicações orientadas a objetos			
Seção 4.1 - Arrays em Java	157		
Seção 4.2 - Strings em Java	173		
Seção 4.3 - Coleções e arquivos	188		

O que é? Para que serve?



Programação Orientada a Objetos

Orientação a Objetos é um Paradigma de Programação.

Paradigma?

Dicionáro Aurélio: modelo, padrão

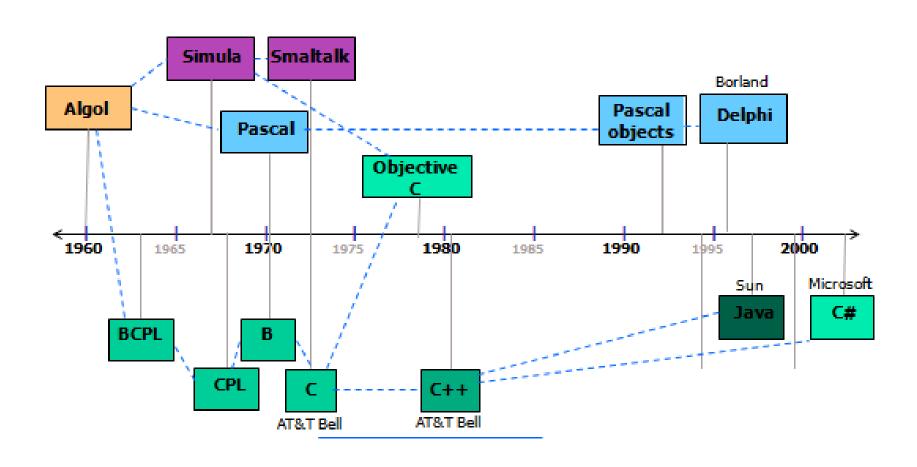
É um "jeito", um "estilo" de programar

Paradigmas de Programação



Tradicional OO Funcional Lógico Fortran, Algol, Algol-68, Pascal, C, Cobol, PL/I Simula-67, Smalltalk, C++, Eiffel, Object Pascal, Java, C# Lisp, ML, Scheme, Miranda, Haskell Prolog, Mercury

Evolução das Linguagens de Programação



Vantagens da POO

- Possuem vantagens sobre o paradigma estruturado:
 - Reutilização de Software (Reusibilidade)
 - reaproveitamento de código
 - Manutenção de Software
 - facilidade de manutenção de sistemas
 - Tamanho de Código Gerado
 - menor código

Qual é a necessidade da POO?





Casa



Arranha-Ceús

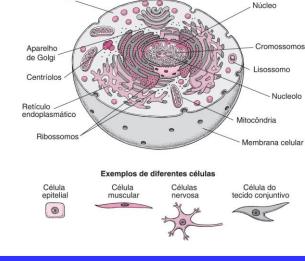
Aumento da complexidade

Origem

• O paradigma O.O. surgiu no fim dos anos 60.

Alan Kay formulou a chamada analogia biológica.

"Como seria um sistema de software que funcionasse como um ser vivo?"



Origem

 Cada "célula" interagiria com outras células através do envio de mensagens

 Cada célula se comportaria como uma unidade autônoma.



 Kay pensou <u>em como construir um sistema</u> de software a partir de agentes autônomos que interagem entre si.

Fundamentos da O.O.

- 1. Qualquer coisa é um objeto.
- 2. Objetos realizam tarefas através da requisição de serviços a outros objetos.
- 3. Cada **objeto pertence a uma determinada** *classe*. Uma <u>classe</u> agrupa objetos similares.
- 4. A classe é um repositório para comportamento associado ao objeto.
- 5. Classes são organizadas em hierarquias.

Origem

Um sistema de software orientado a objetos consiste de objetos em colaboração



com o objetivo de realizar as funcionalidades deste sistema.



Cada **objeto** é responsável por **tarefas específicas**





É através da cooperação entre objetos que a computação do sistema se desenvolve.

Já aprendemos os conceitos da O.O na infância



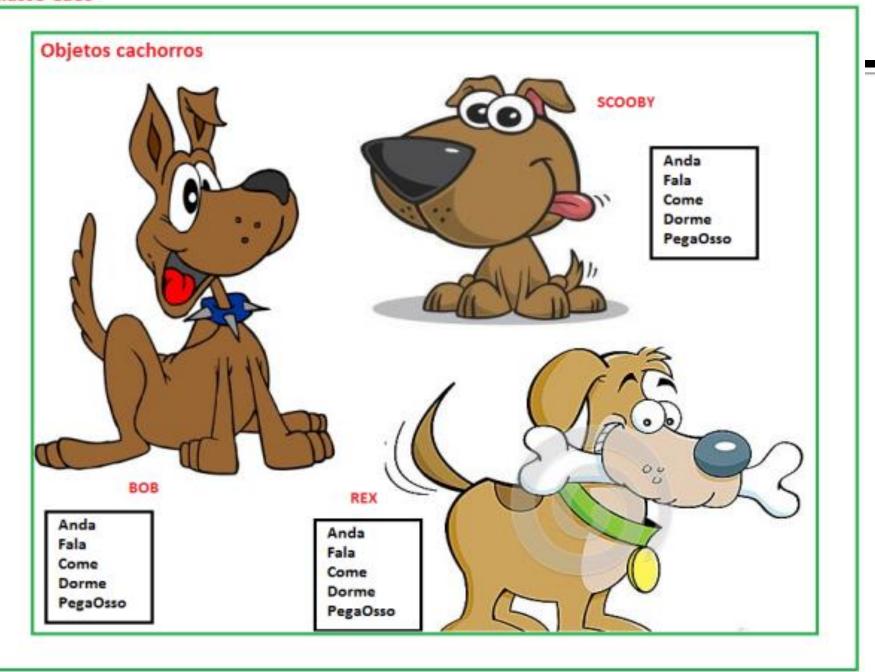
•••

O que é um objeto??????

O que é um objeto?



Possuem características comuns?



Classe e Objeto

Diferentes classes = Diferentes Objetos



Características próprias => ATRIBUTOS

Classe e Objeto

■ Objetos e Classes

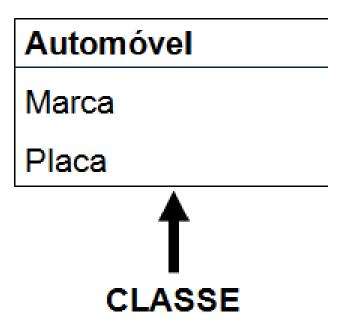
Palio JWO-4567

Parati KLJ-0978

Celta JDK-6543

OBJETOS

(Instâncias da classe Automóvel)



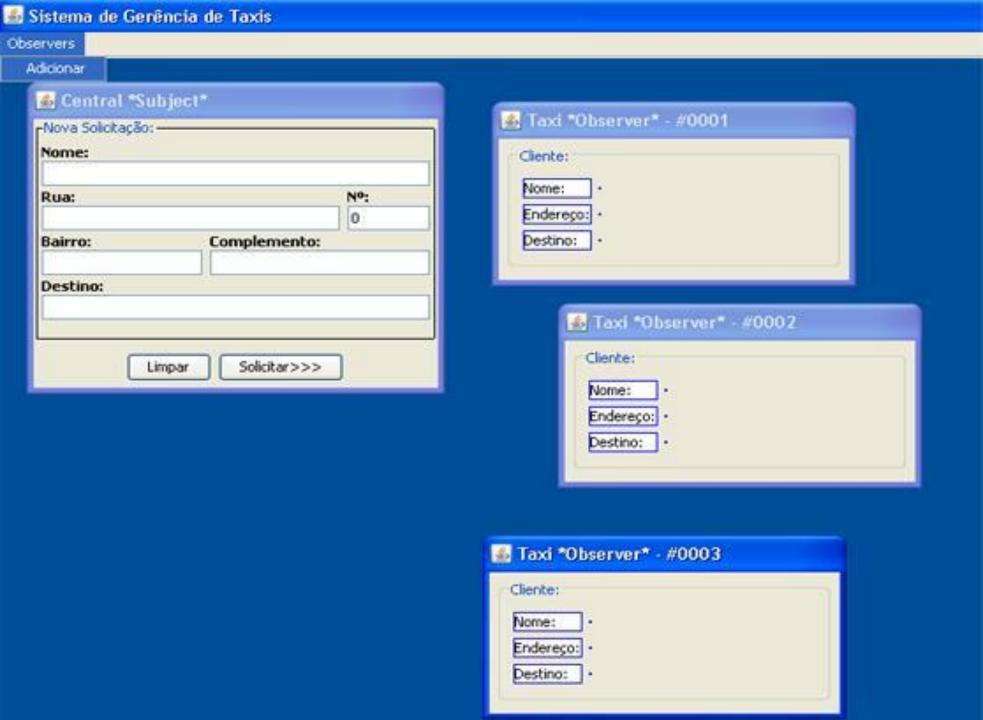
Classe e Objeto



CLASSE Objetos







Conceitos Fundamentais

- Classe / Objeto
- Construtor
- Atributos
- Métodos
- Encapsulamento
- Generalização/Especialização
- Herança
- Polimorfismo e sobrecarga

Conceitos Fundamentais

- Uma classe é um molde para objetos.
- Um objeto é uma instância de uma classe.
- Uma classe é uma *abstração* das características *relevantes* de um grupo de coisas do mundo real.

Pilares da O.O.

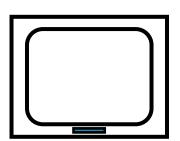


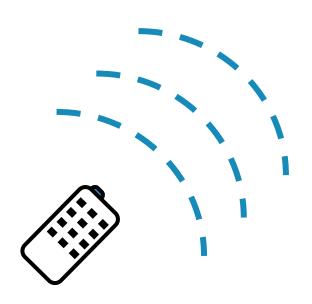
Abstração

Uma abstração é uma representação das características e do comportamento relevantes de um conceito do mundo real para um determinado problema.

- Objetos devem "esconder" a sua complexidade.
 - Legibilidade
 - Clareza
 - Reuso

Deixar **PÚBLICO** somente o que é necessário para enviar mensagens para o objeto.





VISIBILIDADE da classe

Public

Private

Modificadores de Acesso



Quem pode "ver" e acessar o conteúdo de cada atributo e método? Somente a classe ou o programa principal?

Tipos de Visibilidade

■ Público Public

Os Atributos e Métodos que PÚBLICOS, são visíveis para o **programa** que está utilizando a classe

■ Privado Private

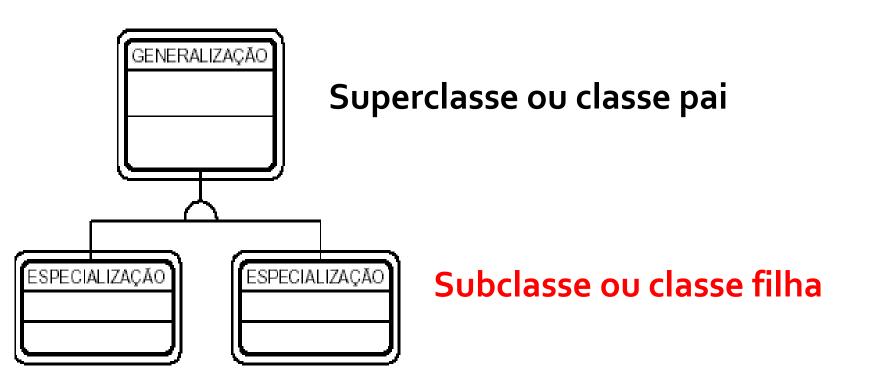
Os Atributos e Métodos que PRIVATE, são visíveis para apenas **DENTRO da classe**

Ocultação de detalhes internos

Tornar a classe uma CÁPSULA, isto é, uma Caixa Preta

Deixar visível somente o **NECESSÁRIO** para a manipulação da classe.

"É um mecanismo para derivar novas classes a partir de classes existentes"



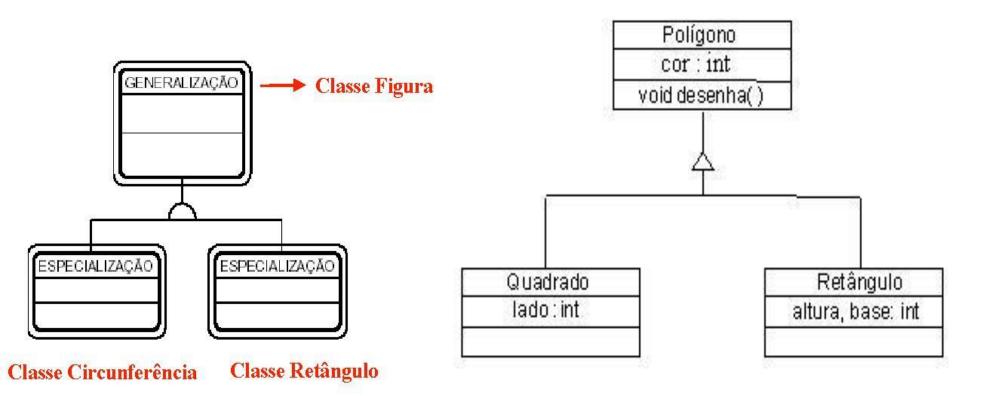
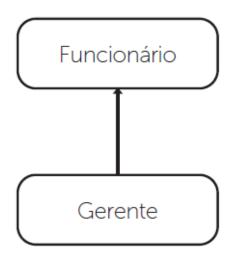


Figura 1.3 | Herança na orientação a objetos



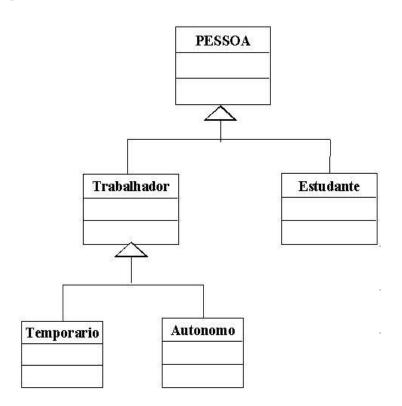
Nome, cargo, data de admissão

Nome, cargo, data de admissão, Quantidade de funcionários que gerencia, senha de acesso.

-

- Redução no custo de manutenção;
- Aumento na reutilização de código ;
- Modularização;

Jantadens



POLIMORFISMO

MUITAS FORMAS

■ Resumindo: "Vamos nos adaptar".

Permite a um método ter várias implementações as quais são selecionadas com base na quantidade de parâmetros e seus tipos que é passado para a invocação do método.

```
Janela ()
Janela (1 x 2 , 2)
Janela (1 x 2 , 2, Azul)
```

Ou sobrecarga?

Permite a existência de vários métodos de mesmo nome, porém com assinaturas levemente diferentes (número, tipo e qtd de parâmetros).

Pessoa nome endereco cpf rg divida atualizarDivida() atualizarDivida(valor) atualizarDivida(valor1, valor2)

Três métodos com o mesmo nome, porém, são diferenciados devido a quantidade de parâmetros passados

 Observe que são declarados 3 construtores e todos eles têm o mesmo nome da classe. Entretanto, cada um deles recebe um número diferente de parâmetros.

```
public class Pessoa {
    private String rg;
    private String nome;

public Pessoa() {
    }
    public Pessoa(String rg, String nome) {
        this.rg = rg;
        this.nome = nome;
    }
    public Pessoa(String nome) {
        this.nome = nome;
}
```

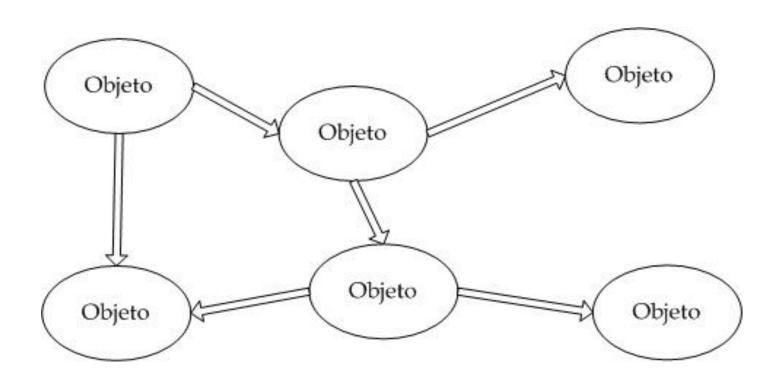
Mensagem

• Para que um objeto realize alguma tarefa, deve haver um estímulo enviado a este objeto.



 Uma mensagem é uma requisição enviada de um objeto a outro para que este último realize alguma operação.

Mensagem



Linguagem de Modelagem Universal (UML)

UML (Unified Modeling Language)

Nome da classe



Pessoa

Atributos



nome: String
CPF: String
DataNascimento: String

Métodos



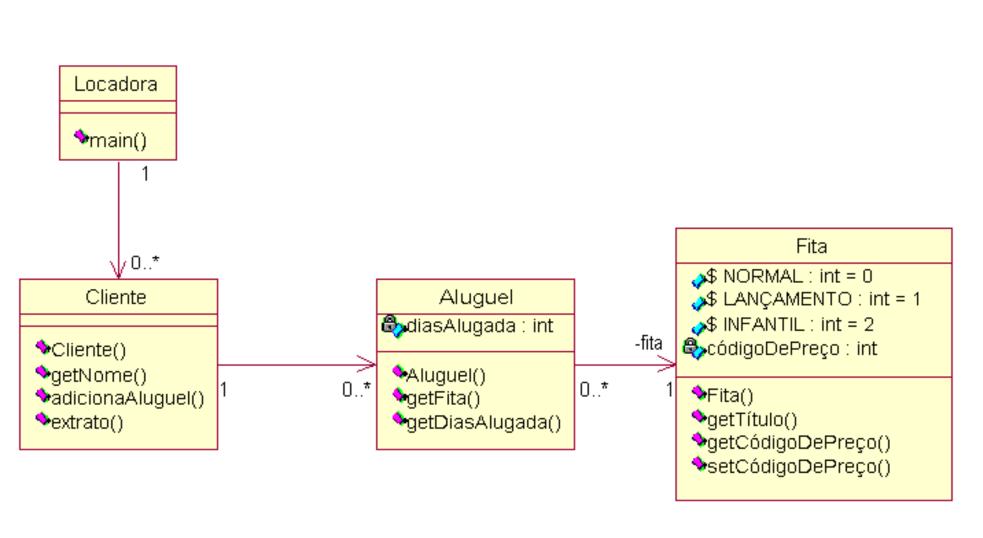
Pessoa(); Calculaldade(): int

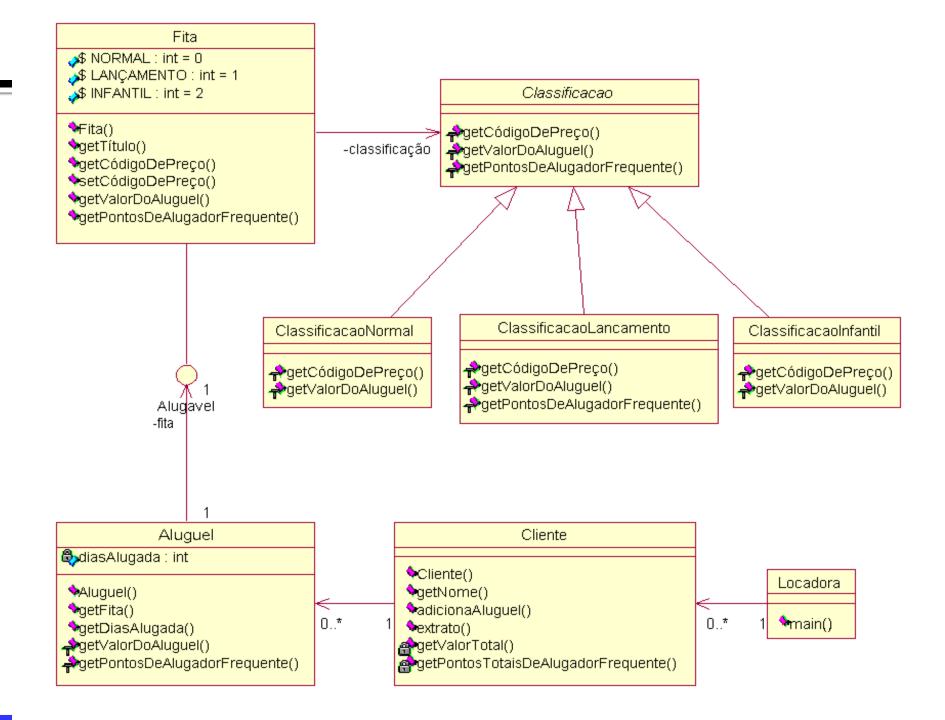
Modelagem O.O. em UML

Linguagem UML (Unified Modeling Language)

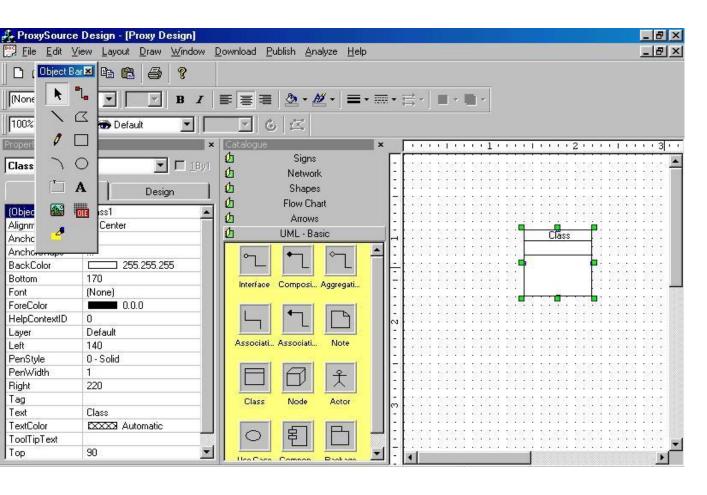
	Lâmpada
-	Estado da lâmpada
-	Acende
-	Apaga
-	Altera cor
-	Mostra estado

Modelagem O.O. em UML





Modelagem O.O. em UML



- -Star UML
- -Proxy design
- Argo UML
- Visio
- Visual UML
- -Poseidon UML
- -Top Coder UML
- -Violet UML
- -jUML
- -Visual Paradigm
- etc

Lembrar....

Classes: estruturas que <u>contém os dados que devem ser representados e as operações</u> <u>que devem ser efetuadas com esses dados para determinado modelo.</u>

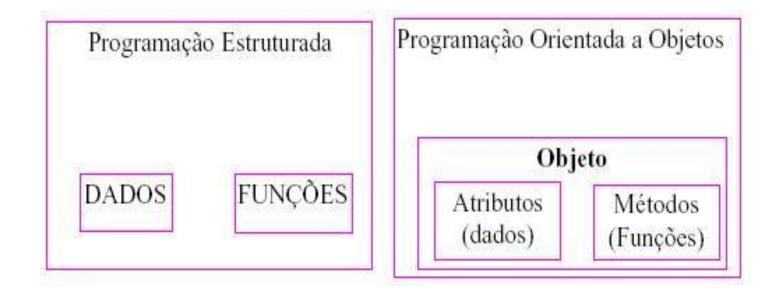
Objetos: é como chamamos a **materialização da classe**, que assim poderá ser usada para representar dados e executar operações.

Obs.: podemos ter vários objetos de uma única classe.

Atributos: são as propriedades (ou dados) da classe. ("variáveis")

Métodos: os métodos expressam as funcionalidades da classe ("funções")

Lembrar....



Implementação de classe

```
public class RegistroAcademicoSimples {
    String nomeDoAluno; //conter o nome do aluno.
    int numeroDeMatrícula; // representará o RA do aluno.
    boolean Bolsista; //assume true ou false.
    int anoDeMatricula; //representará o ano da matrícula.
```

```
//atributos
float lado1;
float lado2;
float lado3;
//construtor
Triangulo (float primeiro, float segundo, float terceiro)
    lado1 = primeiro;
    lado2 = segundo;
    lado3 = terceiro;
//métodos
boolean VerificaEquilatero()
    boolean resultado;
    if((lado1 == lado2)&& (lado2 == lado3))
        resultado = true;
    else resultado = false;
    return resultado;
float calculaPerímetro()
    float resultado = lado1 + lado2 + lado3;
    return resultado;
```

public class Triangulo {

```
2 public class Calculadora {
     //atributos
     int numerol;
     int numero2;
     //construtor
     Calculadora(int n1, int n2)
         numero1 = n1;
         numero2 = n2;
     //métodos
     float Media()
          float m;
         m = (numero1 + numero2)/2;
          return m;
     int Multiplica()
          int m;
         m = numero1 * numero2;
          return m;
```

Antes de iniciar...

É necessário instalar:

Java Development Kit (JDK)

https://www.oracle.com/java/technologies/downloads/#jdk18-windows

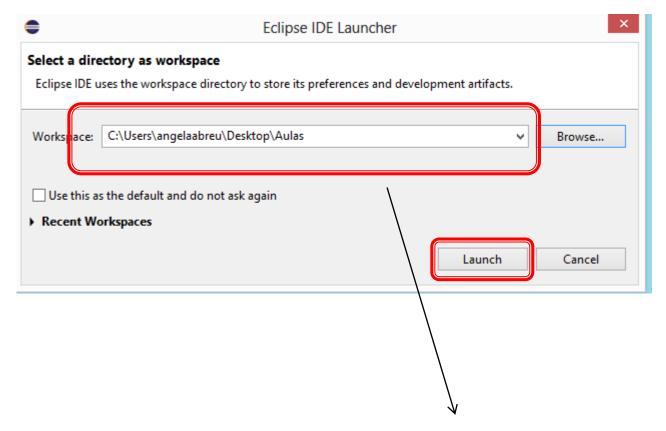
Integrated Development Environment (IDE) Eclipse

https://www.eclipse.org/downloads/download.php?file=/oomph/epp/2022-03/R/eclipse-inst-jre-win64.exe

eclipse IDE for Java Developers



Orientação a Objetos



Diretório que irá armazenar o seu projeto.

Para cada novo programa, CRIEM uma nova pasta para armazenar o projeto.



8



Welcome to the Eclipse IDE for Java Developers



Review IDE configuration settings

Review the IDE's most fiercely contested preferences



Overview

Get an overview of the features



Create a Hello World application

A guided walkthrough to create the famous Hello World in Eclipse



Tutorials

Go through tutorials



Create a new Java project

Create a new Java Eclipse project



Samples

Try out the samples



Checkout projects from Git

Checkout Eclipse projects hosted in a Git repository



What's New

Find out what is new



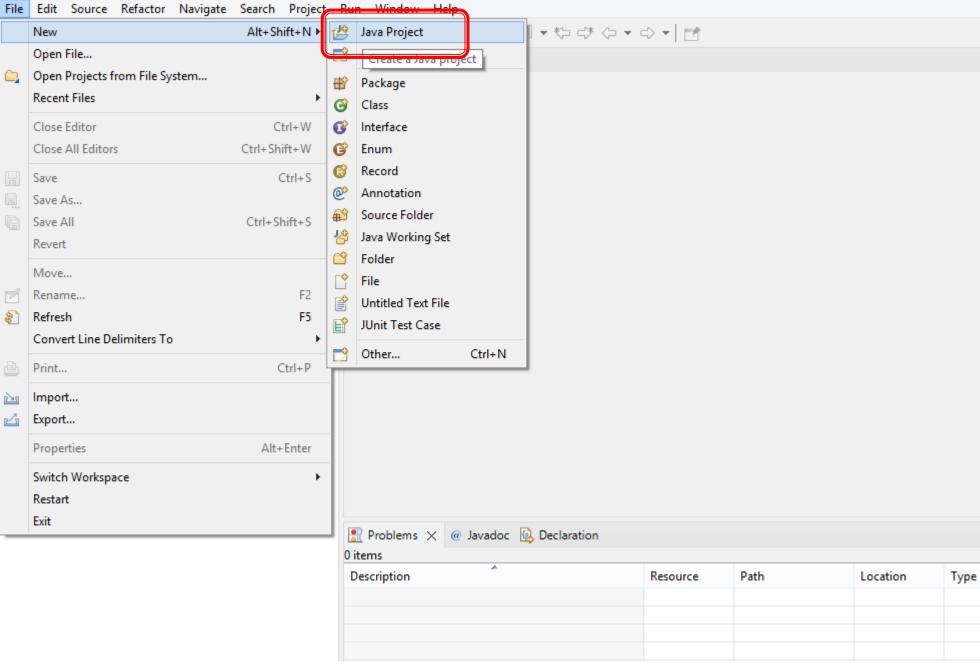
Import existing projects

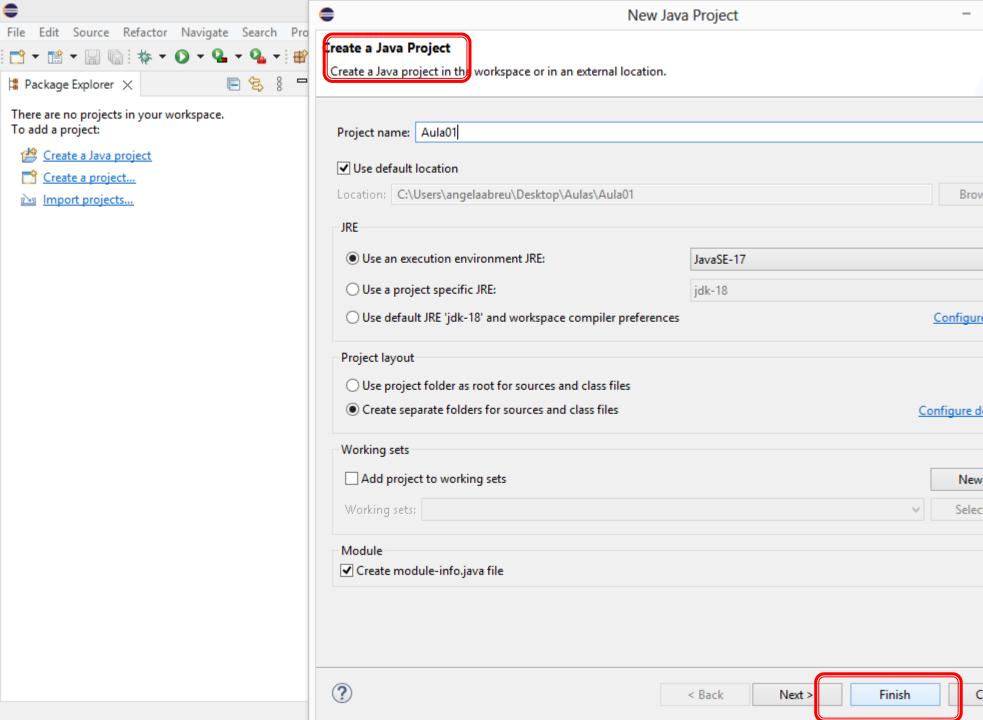
Import existing Eclipse projects from the filesystem or archive

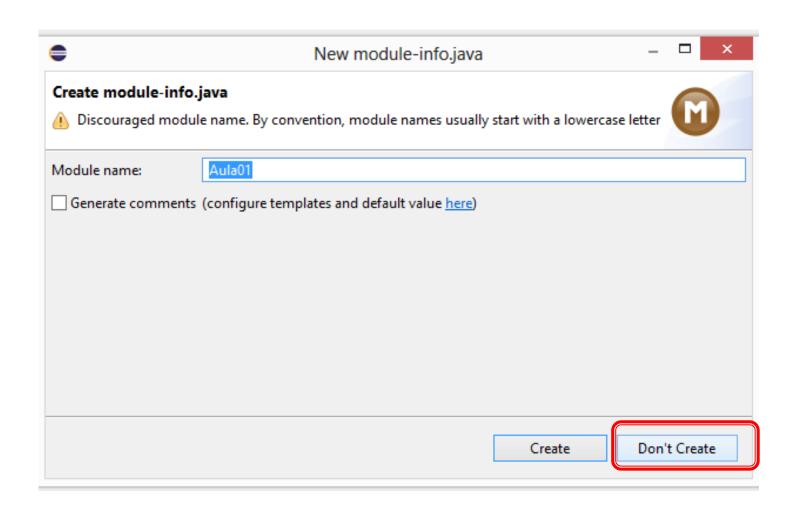


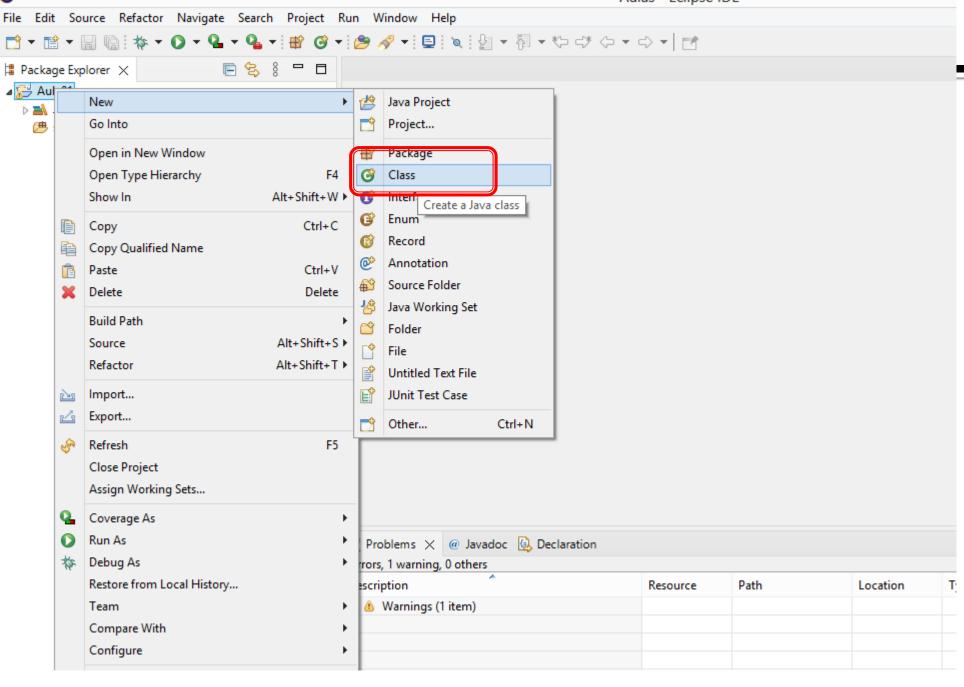
Launch the Eclipse Marketplace

Enhance your IDE with additional plugins and install your Marketplace favorites

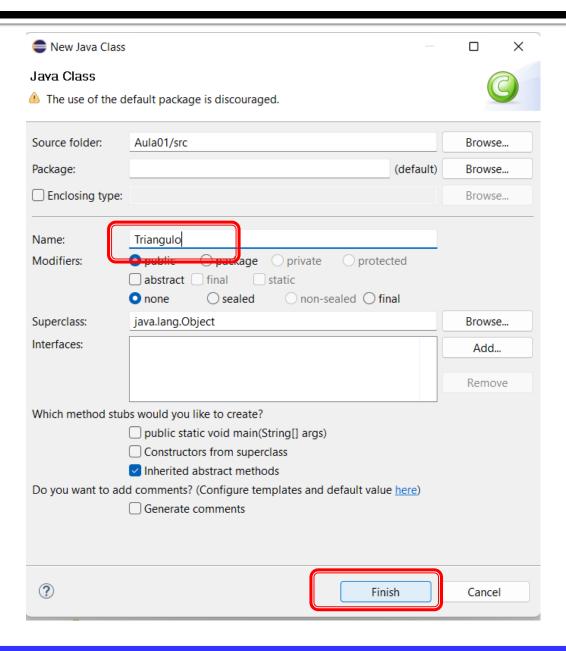








Orientação a Objetos



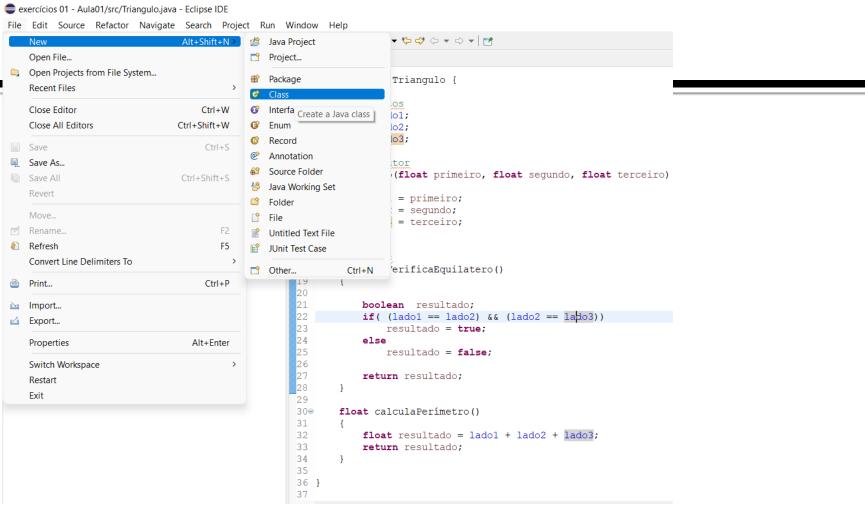
Orientação a Objetos

```
exercícios 01 - Aula01/src/Triangulo.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
🖹 😫 🖁 🗖 🖟 Triangulo.java 🗵

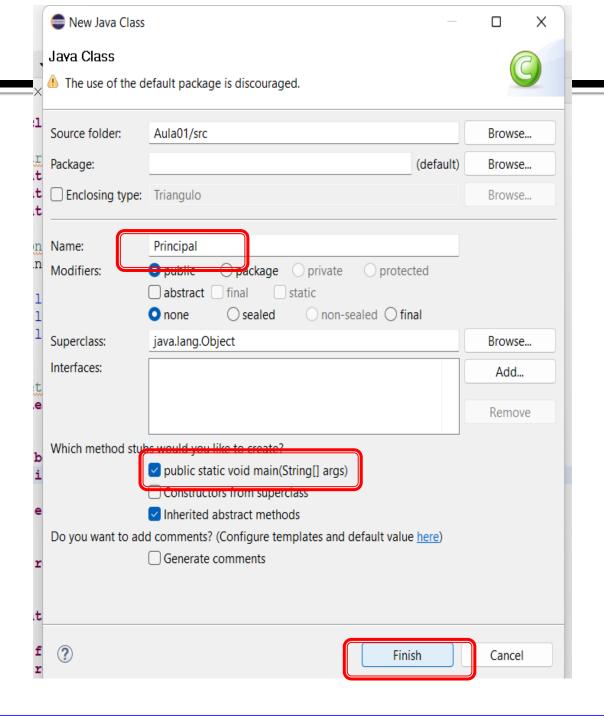
☐ Package Explorer ×

2 public class Triangulo {
 ⇒ Mark System Library [JavaSE-17]
 4
   > 🗾 Triangulo.java
```

```
2 public class Triangulo {
4
      //atributos
       float lado1;
6
      float lado2;
       float lado3;
9
       //construtor
10⊝
       Triangulo (float primeiro, float segundo, float terceiro)
11
12
           lado1 = primeiro;
13
           lado2 = segundo;
14
           lado3 = terceiro;
15
16
17
       //métodos
18⊜
       boolean VerificaEquilatero()
19
20
           boolean resultado;
           if( (lado1 == lado2) && (lado2 == lado3))
23
               resultado = true;
24
           else
               resultado = false;
26
27
           return resultado;
28
29
30⊜
       float calculaPerimetro()
31
32
           float resultado = lado1 + lado2 + lado3;
33
           return resultado;
34
35
36 }
```



Agora vamos criar a classe Principal para executar o programa → método main



```
public class Principal {
    public static void main(String[] args) {
        // TODO Auto-generated method stub
    }
}
```

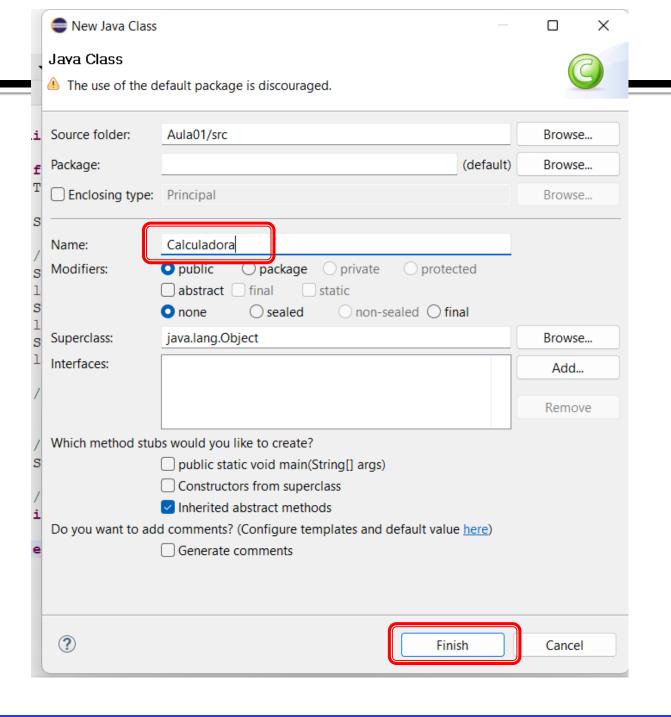
Agora vamos instanciar objetos da Classe Triângulo no programa principal.

```
3
 4 public class Principal {
 6⊝
       public static void main(String[] args) {
 7
 8
           float lado1,lado2,lado3;
 9
           Triangulo objeto1; //declaração do objeto da classe Triangulo
10
11
           Scanner teclado = new Scanner(System.in); //leitura do teclado
12
13
           //solicitar que o usuário digite os 3 lados do triangulo
           System.out.println("Digite o lado 1 do Triangulo: ");
14
15
           lado1 = teclado.nextFloat();
           System.out.println("\n Digite o lado 2 do Triangulo: ");
16
17
           lado2 = teclado.nextFloat();
18
           System.out.println("\n Digite o lado 3 do Triangulo: ");
19
           lado3 = teclado.nextFloat();
20
21
           //Instanciar o objeto triangulo
22
            objeto1 = new Triangulo(lado1,lado2,lado3);
23
24
           //chamar o método para calcular o perímetro
25
           System.out.println("\n O perimetro do triangulo e " + objeto1.calculaPerimetro());
26
27
           //chamar o método para verificar se é equilátero ou não
28
           if( objeto1.VerificaEquilatero() == true)
29
               System.out.println("\n 0 triangulo é equilatero");
30
           else System.out.println("\n 0 triangulo nao é equilatero");
31
32
33
34
35
36 }
```

Iriangulo.java

2 import java.util.Scanner;

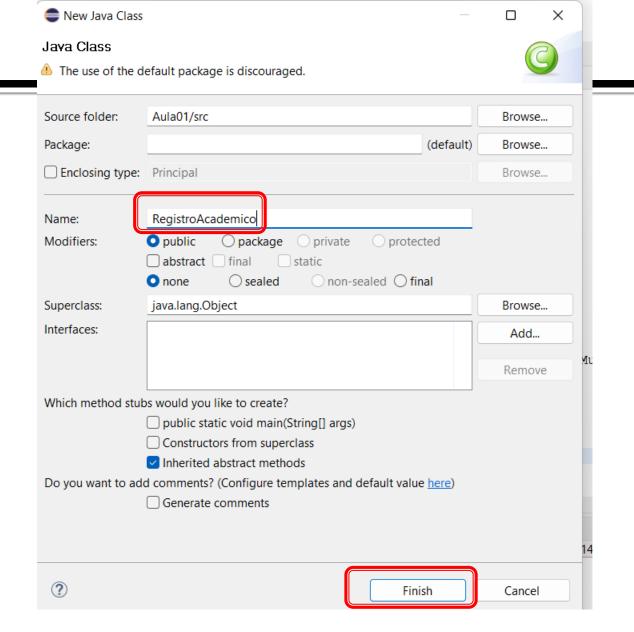
```
exercícios 01 - Aula01/src/Principal.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
        □ 🛮 📴 Outline ×
                                    □ Package
                                                Triangulo.java
                                                             Principal.java ×
                Run Principal
                                                                                                                                                            Principal
             Library [JavaSE-17]
                                                         public static void main(String[] args) {
                                                                                                                                                                   § main(String[])
                                                             float lado1,lado2,lado3;
    9
                                                             Triangulo objetol; //declaração do objeto da classe Triangulo
      > 🔑 Principal.java
      > / Triangulo.java
                                                 11
                                                             Scanner teclado = new Scanner(System.in); //leitura do teclado
                                                 12
                                                 13
                                                             //solicitar que o usuário digite os 3 lados do triangulo
                                                 14
                                                             System.out.println("Digite o lado 1 do Triangulo: ");
                                                 15
                                                             lado1 = teclado.nextFloat();
                                                 16
                                                             System.out.println("\n Digite o lado 2 do Triangulo: ");
                                                 17
                                                             lado2 = teclado.nextFloat();
                                                 18
                                                             System.out.println("\n Digite o lado 3 do Triangulo: ");
                                                 19
                                                             lado3 = teclado.nextFloat();
                                                 20
                                                 21
                                                             //Instanciar o objeto triangulo
                                                 22
                                                              objeto1 = new Triangulo(lado1, lado2, lado3);
                                                 23
                                                 24
                                                             //chamar o método para calcular o perímetro
                                                 25
                                                             System.out.println("\n O perimetro do triangulo e " + objeto1.calculaPerímetro());
                                                 26
                                                 27
                                                             //chamar o método para verificar se é equilátero ou não
                                                 28
                                                             if( objeto1.VerificaEquilatero() == true)
                                                 29
                                                                 System.out.println("\n 0 triangulo e equilatero");
                                                 30
                                                             else System.out.println("\n 0 triangulo nao e equilatero");
                                                 31
                                                 32
                                                 33
                                                 34
                                                 35
                                                 36 }
                                                                                                                                                    m × % |
                                                Problems @ Javadoc  Declaration  Console ×
                                                                                                                                                              <terminated> Principal [Java Application] C:\Program Files\Java\jdk-18.0.1.1\bin\javaw.exe (10 de ago. de 2022 14:08:13 – 14:08:26) [pid: 5088]
                                                Digite o lado 1 do Triangulo:
                                                Digite o lado 2 do Triangulo:
                                                Digite o lado 3 do Triangulo:
                                                O perimetro do triangulo e 6.0
                                                O triangulo nao e equilatero
```



```
2 public class Calculadora {
4
       //atributos
       int numerol;
       int numero2;
6
       //construtor
8
9⊝
       Calculadora (int n1, int n2)
10
           numero1 = n1;
           numero2 = n2;
13
14
15
       //métodos
16⊜
       float Media()
17
18
           float m;
           m = (numero1 + numero2)/2;
20
           return m;
21
23⊜
       int Multiplica()
24
           int m;
26
           m = numero1 * numero2;
27
           return m;
28
30 }
```

Na classe/programa principal...

```
🔑 *Principal.java 🔀 🛂 Calculadora.java
30
31
32
33
34
            int numero1, numero2;
35
            //solicitar que o usuário digite 2 numeros
36
            System.out.println("\n Digite o primeiro numero: ");
37
            numero1 = teclado.nextInt();
38
            System.out.println("\n Digite o segundo numero: ");
39
            numero2 = teclado.nextInt();
40
41
            Calculadora objetoCalculadora = new Calculadora(numero1, numero2);
42
43
            //chamar o método para calcular a multiplicacao
44
45
            System.out.format("\n 0 produto dos numeros e %d", objetoCalculadora.Multiplica());
46
47
```

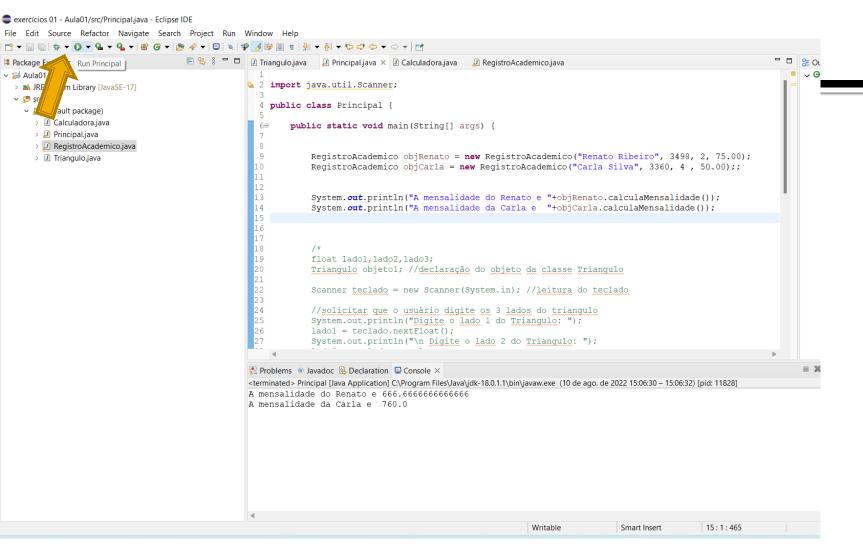


```
3
       //Atributos COM ENCAPSULAMENTO
 4
       private String nomeDoAluno; //nome do aluno
       private int numeroDeMatricula; //número de matrícula
       private int códigoDoCurso; //código do curso (1 .. 4)
 6
 7
       private double percentualDeCobranca; //percentual em relação ao preço cheio, de 0 a 100%
 9
       //construtor
       RegistroAcademico(String n, int m, int c, double p) {
10⊝
11
               nomeDoAluno = n:
12
               numeroDeMatricula = m;
13
               códigoDoCurso = c;
14
               percentualDeCobranca = p;
15
16
       //Metodo
17⊝
       public double calculaMensalidade() {
           double mensalidade = 0:
18
19
           //determinação do valor cheio da mensalidade, dependendo do curso.
20
21
           if (códigoDoCurso == 1)// Redes
22
               mensalidade = 450.00;
23
           if (códigoDoCurso == 2) // Ciência da Computação
               mensalidade = 500.00:
24
25
           if (códigoDoCurso == 3) // Engenharia da Computação
               mensalidade = 550.00;
26
27
           if (códigoDoCurso == 4) //Sistemas de Informacao
               mensalidade = 380.00;
28
29
30
           //calcula o desconto. Se o percentual for zero, a mensalidade também o será.
31
           if (percentualDeCobranca == 1)
32
               mensalidade = 450.00;
33
           else mensalidade = mensalidade * 100.0 / percentualDeCobranca;
34
           return mensalidade:
35
36
37
```

1 public class RegistroAcademico {

```
☑ Triangulo.java
                                       RegistroAcademico.java
 2 import java.util.Scanner;
 4 public class Principal {
      public static void main(String[] args) {
 6⊖
          RegistroAcademico objRenato = new RegistroAcademico("Renato Ribeiro", 3498, 2, 75.00);
          RegistroAcademico objCarla = new RegistroAcademico("Carla Silva", 3360, 4 , 50.00);;
12
13
          System.out.println("A mensalidade do Renato e "+objRenato.calculaMensalidade());
          System.out.println("A mensalidade da Carla e "+objCarla.calculaMensalidade());
15
16
```

illuow ileip



...

Exercícios 😜

```
//atributos
float lado1;
float lado2;
float lado3;
//construtor
Triangulo (float primeiro, float segundo, float terceiro)
    lado1 = primeiro;
    lado2 = segundo;
    lado3 = terceiro;
//métodos
boolean VerificaEquilatero()
    boolean resultado;
    if((lado1 == lado2)&& (lado2 == lado3))
        resultado = true;
    else resultado = false;
    return resultado;
float calculaPerímetro()
    float resultado = lado1 + lado2 + lado3;
    return resultado;
```

public class Triangulo {

```
2 public class Calculadora {
     //atributos
     int numerol;
     int numero2;
     //construtor
     Calculadora(int n1, int n2)
         numero1 = n1;
         numero2 = n2;
     //métodos
     float Media()
          float m;
         m = (numero1 + numero2)/2;
          return m;
     int Multiplica()
          int m;
         m = numero1 * numero2;
          return m;
```

```
3
       //Atributos COM ENCAPSULAMENTO
 4
       private String nomeDoAluno; //nome do aluno
       private int numeroDeMatricula; //número de matrícula
       private int códigoDoCurso; //código do curso (1 .. 4)
 6
 7
       private double percentualDeCobranca; //percentual em relação ao preço cheio, de 0 a 100%
 9
       //construtor
       RegistroAcademico(String n, int m, int c, double p) {
10⊝
11
               nomeDoAluno = n:
12
               numeroDeMatricula = m;
13
               códigoDoCurso = c;
14
               percentualDeCobranca = p;
15
16
       //Metodo
17⊝
       public double calculaMensalidade() {
           double mensalidade = 0:
18
19
           //determinação do valor cheio da mensalidade, dependendo do curso.
20
21
           if (códigoDoCurso == 1)// Redes
22
               mensalidade = 450.00;
23
           if (códigoDoCurso == 2) // Ciência da Computação
               mensalidade = 500.00:
24
25
           if (códigoDoCurso == 3) // Engenharia da Computação
               mensalidade = 550.00;
26
27
           if (códigoDoCurso == 4) //Sistemas de Informacao
               mensalidade = 380.00;
28
29
30
           //calcula o desconto. Se o percentual for zero, a mensalidade também o será.
31
           if (percentualDeCobranca == 1)
32
               mensalidade = 450.00;
33
           else mensalidade = mensalidade * 100.0 / percentualDeCobranca;
34
           return mensalidade:
35
36
37
```

1 public class RegistroAcademico {

Exercício 1

Faça um programa Orientado a Objetos que receba 3 números e calcule as seguintes operações: soma, média, subtração, multiplicação e determine o maior número entre eles.

Dica: construa a classe "OperacoesMatematicas"

Construtor: informar os 3 números

Métodos:

- Para somar;
- Para calcular a média;
- Para subtrair;
- Para multiplicar;
- Para determinar o maior número.

Exercício 2

Crie uma classe Retangulo conforme a descrição abaixo:

Retangulo - lado1: double - lado2: double - area: double - perimetro: double + Retangulo() + Retangulo(lado1: double, lado2: double) + calcularArea(): void + calcularPerimetro(): void

- •O método calcularArea deve realizar o cálculo da área do retângulo (area = lado1 * lado2) e retornar o valor calculado;
- •O método calcularPerimetro faz o cálculo do perimetro (perimetro = 2*lado1 + 2*lado2) e retornar o valor calculado;

Crie o um programa principal e instancie um objeto da classe Retangulo:

- •Atribua o valor 10 ao atributo lado1.
- Atribua o valor 5 ao atributo lado2.
- •Chame o método calcularArea e mostre o resultado na tela:
- Chame o método calcularPerimetro.



Muito Obrigada!

Profa. Angela Abreu Rosa de Sá, Dra.

Contato: angelaabreu@gmail.com