

Programação Orientada a Objetos



Dr^a. Alana Moraes
alanamm.prof@gmail.com

Vamos pensar em
um exemplo real
Hummm E se eu
precisasse calcular a
área de um
retângulo?



ROTEIRO DE HOJE

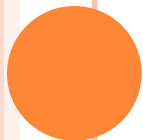
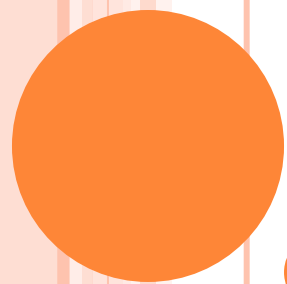
- Definições importantes de Orientação a Objeto.



PARADIGMAS DE PROGRAMAÇÃO

<i>Paradigma</i>	<i>Princípios</i>	<i>Linguagem</i>
Orientado a procedimento	Decomposição e Modularização	Basic, Fortran, Pascal, C, Cobol, Clipper, php
Orientado a funções	Tudo é função	Lisp,
Orientado a Texto	Casamento de padrões de texto	Snobol,
Orientado a lógica	Dedução sobre Regras e fatos	Prolog,
Orientado a objeto	Abstração e Reuso	C++, Eiffel, Java, Phyton, C#, Lua, Ruby





CONCEITOS DE OO

CONCEITOS IMPORTANTES

- Classe
- Objeto
- Atributo
- Métodos
- Instância



CONCEITOS IMPORTANTES

- Classe
- Objeto
- Atributo
- Métodos
- Instância



CLASSE

- Uma classe é um gabarito para a definição dos objetos, definido pelo programador.
- Pode conter o método *main()* ou não.
 - `public static void main(String[] args){ ... }`
- Composta por:
 - Nome da Classe
 - Métodos
 - Atributos



CLASSE

- **Nome da classe:** identificador para a classe, que permite referenciá-la.
- **Atributos:** descreve características da classe.
- **Métodos:** definem as funcionalidades da classe.
- **Modificador de Acesso:** determina quem pode acessar a classe.

Sintaxe:

```
[modificador de acesso] class [nomeDaClasse]
{
    [atributos e métodos]
}
```



CLASSE

- Os modificadores quem definem o tipo da classe
- Tipos:
 - **Classe Pública:**
 - É conhecida apenas no escopo delimitado pelo arquivo que a contém.



CLASSE

- Declaração:

- **Classe pública:**

```
public class ClasseTestePublico  
{  
    (...)  
}
```

ou

```
class ClasseTestePublico  
{  
    (...)  
}
```



Será que eu posso ter mais de
uma Classe dentro de um .java?



Será que eu posso ter mais de uma Classe dentro de um .java?

R=> Via IDE você não consegue!!

Esta não é a melhor técnica de programação.

Cuidado com o acoplamento das classes.



CONCEITOS IMPORTANTES

- Classe
- Objeto
- Atributo
- Métodos
- Instância



OBJETO

- É uma representação computacional de uma entidade do mundo real.
- Uma particular instância de uma classe é chamada objeto
- Comparamos as **classes** às fábricas e os **objetos** aos produtos feitos por elas.



OBJETO

- As classes não ocupam espaço na memória por serem abstrações.
- Enquanto isso, os objetos ocupam espaço de memória por serem concretizações dessas abstrações.



OBJETO

- Um objeto representa qualquer coisa do mundo real que seja manipulada pelo nosso programa, ou então representa blocos de construção do próprio programa
 - O programa
 - Uma Conta-corrente
 - Um cliente
 - Uma janela
 - Um botão



OBJETO

- Assim como as coisas no mundo real, os objetos tem “estado” e “comportamento”
 - **Estado** são informações sobre o objeto, como a sua cor, seu peso, o saldo da conta-corrente, etc.
 - **Comportamento** são coisas que podem ser feitas com ou pelo objeto, como depositar em uma conta-corrente ou mudar a cor de uma janela

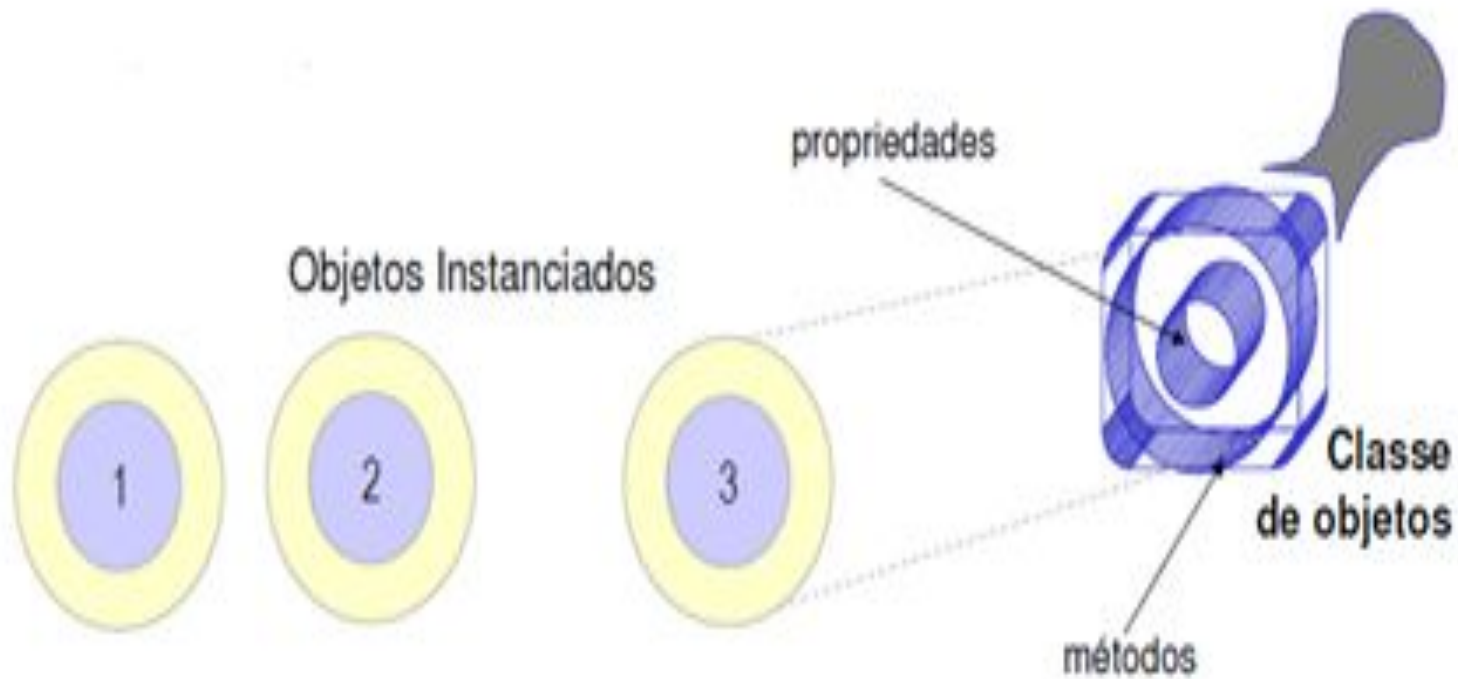


QUAL A DIFERENÇA ENTRE UMA CLASSE E UM OBJETO?

- A **classe** define as propriedades do **objeto**.
 - A Classe informa a JVM como criar o objeto.
- Vários objetos semelhantes possuem o mesmo tipo de informação em seu estado e tem o mesmo comportamento.
- Exemplo:
 - Polígonos



QUAL A DIFERENÇA ENTRE UMA CLASSE E UM OBJETO?



CONCEITOS IMPORTANTES

- Classe
- Objeto
- **Atributo**
- Métodos
- Instância



ATRIBUTO

- Também conhecido como variável de instância.
- Sintaxe: **[modificador] tipo nome [= default];**
- **[modificador] = private, public e protected**
- Cada atributo é identificado por um **nome** e tem um **tipo** associado.
 - Tipo associado pode ser:
 - Tipo primitivo
 - Outra classe Java.



ATRIBUTO

- O conjunto de **atributos** descreve as propriedades da classe.
- Dados de tipo primitivos são sempre referenciados por valor.
- Os objetos são sempre referenciados por meio de sua referência.
- O atributo pode ainda ter um **valor default opcional**.



CONCEITOS IMPORTANTES

- Classe
- Objeto
- Atributo
- Métodos
- Instância



MÉTODOS

- Representam as funcionalidades da classe.
- O comportamento de um objeto é tudo o que ele sabe fazer, e tudo o que pode ser feito com ele
- Equivalente às funções em linguagens estruturadas
- Manipulam:
 - Variáveis locais;
 - Atributos dos objetos



MÉTODOS

- Sintaxe:
**[modificador] tipo nome(argumentos) {
 corpo do método
}**
- O modificador de visibilidade pode estar presente tanto para atributos como para métodos.
- O tipo é um indicador do valor de retorno, sendo *void* se o método não tiver um valor de retorno;



MÉTODOS

- Em princípio, três categorias de visibilidade podem ser definidas:
 - Público;
 - Privativo;
 - Protegido;
- Exemplo
 - Público: **public void methodExample(){...}**
 - Privativo: **private void methodExample(){...}**
 - Protegido: **protected void methodExample(){...}**



CONCEITOS IMPORTANTES

- Classe
- Objeto
- Atributo
- Métodos
- Instância



INSTÂNCIA

- É a declaração de um objeto que foi definido por uma classe.
- Utiliza o operador **new**
 - `Scanner input = new Scanner(System.in);`
- Dizemos que um objeto em particular de uma dada classe é uma instância desta classe.



INSTÂNCIA

- Cada instância possui o seu próprio conjunto de atributos, independente de outras instâncias da mesma ou de outras classes.
- Todas as instâncias de uma mesma classe compartilham as mesmas definições de métodos.



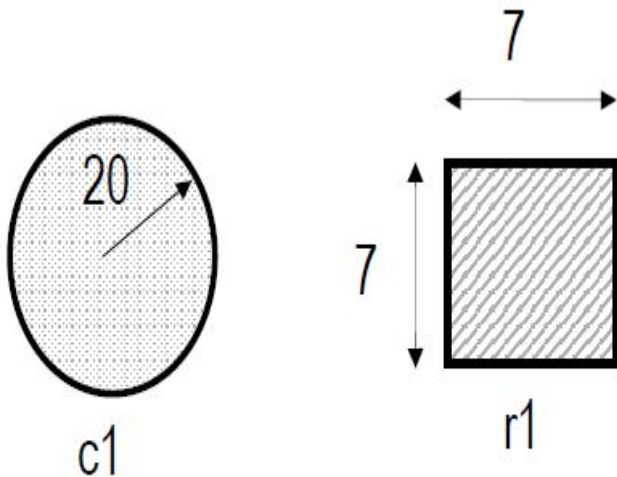
DECLARAR X INSTANCIAR.

- A diferença mais evidente entre a **instância de um objeto** de uma classe e a **declaração de um dado** primitivo reside na necessidade de reservar memória para o objeto através do uso do operador **new**.
- Na verdade, esse operador realiza uma série de tarefas:
 - Reserva espaço para a instância da classe **Vértice**, o qual deve ser suficiente para conter seu estado, isto é, os valores dos seus campos;
 - Realiza a chamada do método **construtor**;
 - Retorna uma referência para o novo objeto, o qual é atribuído à variável **v**.



INSTÂNCIA

Instancie os seguintes objetos:



Retangulo r1 =

Circulo c1 =



CICLO DE VIDA DE UM OBJETO

- **Instanciação:** o objeto é criado na memória e passa a ser referenciado por uma variável de referência;
- **Uso:** o objeto recebe mensagens de outros objetos e, com isso, executa parte da funcionalidade do sistema;
- **Destruição:** quando o objeto não é mais referenciado (inacessível) ele torna-se elegível para a coleta de lixo.
- **Coletor de Lixo:** Limpa a memória ocupada pelos objetos inacessíveis (quando há falta de memória).



CICLO DE VIDA DE UM OBJETO

Exemplo:

```
Retangulo r;  
r = new Retangulo(5,30);  
r.setLargura(10);  
r = null;
```



ROTEIRO DE CRIAÇÃO DE CLASSE

Criar a classe

1. Pensar nos atributos
 - ▣ Privativos
2. Criar construtor
 - ▣ Composto por todos os atributos (por enquanto)
3. Planejar e implementar os métodos

Verificar se outras classes são necessárias

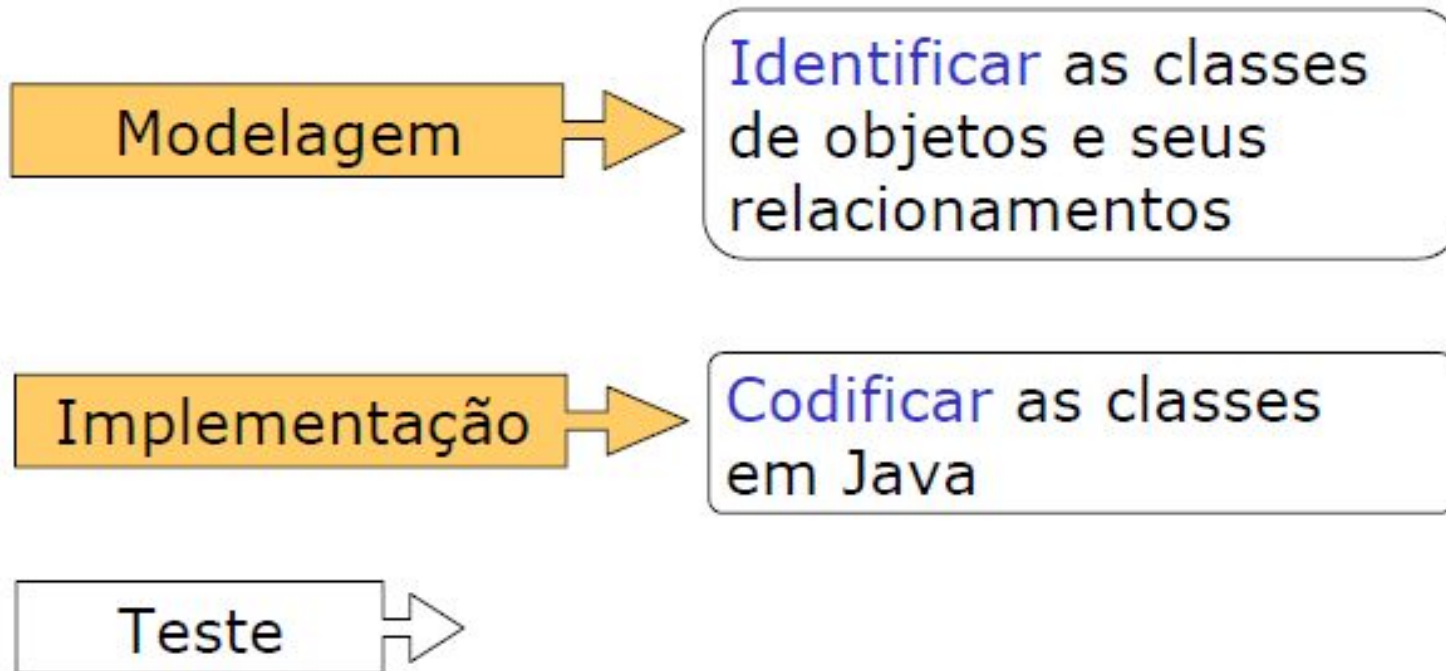
Repetir sub-etapas anteriores

Criar classe Teste

Classe que tem o método: `public static void main(String args[]){ ... }`

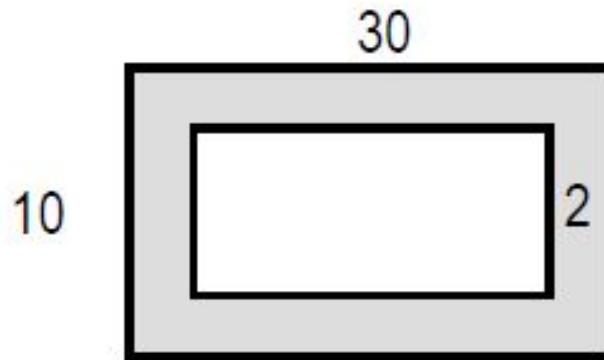


IMPLEMENTANDO OO



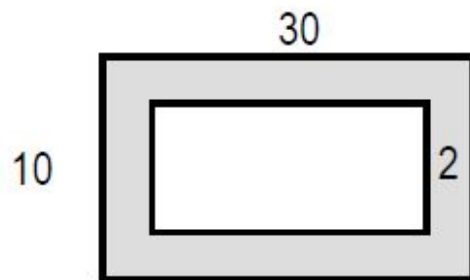
EXERCÍCIO

- Implemente um programa que calcule a área de uma moldura.



EXERCÍCIO

- Crie a classe Moldura
 - Exemplo de instanciação:



Moldura
Retangulo rinterno Retangulo reexterno
area()

```
Moldura m1 = new Moldura(10, 30, 2);  
Moldura m2 = new Moldura(6,26,10,30);
```



SUA VEZ ...

- E se a moldura fosse circular?
- Calcule este caso!



DÚVIDAS ?

