



***PARADIGMA
ORIENTADO A
OBJETO***

O QUE VEREMOS NESSE BIMESTRE?

1 – Paradigma Orientado a Objeto

1.1 – Objetos e Classes

1.1.1 – Instanciação de objetos

1.1.2 – Identidade, Estado e Métodos de um objeto

1.1.3 – Mensagens

1.2 – Encapsulamento

1.3 – Herança

1.4 – Polimorfismo

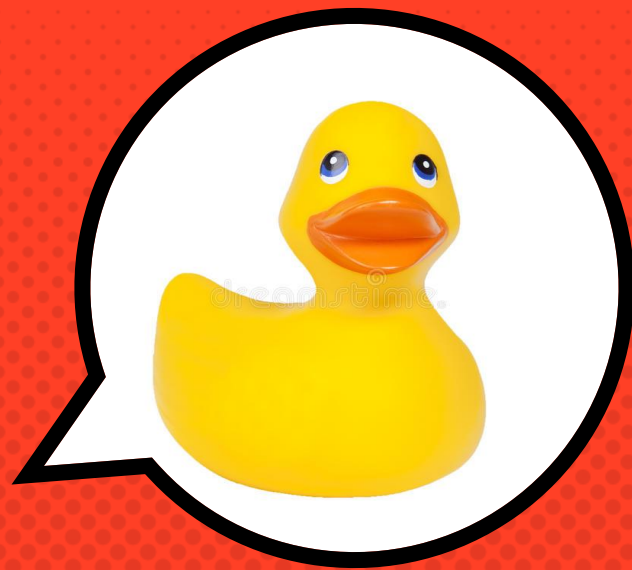
1.5 – Classes Concretas e Classes Abstratas

1.6 – Métodos Construtores e Destrutores

1.7 – Métodos de Acesso getters e setters.

OBJETO!

O QUE É UM OBJETO?



OBJETOS

- × Onde quer que você olhe no mundo real, você verá objetos (pessoas, animais, plantas, carros, edifícios, etc).



OBJETOS

- × Humanos pensam em termos de objetos
 - × Portanto, POO é alto nível
- × Os objetos de software assim como os objetos do mundo real possuem interação uns com os outros.
- × Os programas de computador, como o java que será utilizado, são compostos de muitos objetos de software interativos.

CARACTERÍSTICAS DOS OBJETOS

Classificação

Animados: possuem vida, se movem...

Inanimados: não se movem por conta própria.

Objetos possuem atributos

Tamanho, forma, cor, peso, etc.



Objetos exibem comportamentos

Uma bola pode rolar, inflar, murchar;

Um cão pode latir, morder, andar;

Um carro pode acelerar, frear, desviar.

OBJETOS

- × Os humanos aprendem sobre os objetos, analisando os seus atributos e observando seus comportamentos, ou seja, diferentes objetos podem ter atributos semelhantes e podem exibir comportamentos semelhantes.
- × Exemplo: Podemos comparar por exemplo, atributos e comportamentos entre bebês e adultos.

OBJETOS E CLASSES

- × Geralmente um objeto é representado por uma ou mais classes.
- × A unidade de programa em Java, damos o nome de Classe, que abrigam o método.
- × Em uma classe, você fornece um ou mais métodos que são projetados para realizar as tarefas da classe.



EXEMPLO DE CLASSE: ***CACHORRO***

Vamos entender na prática o que é
um objeto e como ele é
representado em Java.





EXERCÍCIOS

AGORA QUE VOCÊ SABE O QUE É UM OBJETO E UMA INSTÂNCIA DO OBJETO, FAÇA:

- × Objeto Pessoa;
- × Objeto Carro;
- × Objeto Conta bancária;

- × PS: Cada objeto deve conter no mínimo 6 atributos, 3 métodos e 4 instâncias;

ESTADOS E COMPORTAMENTO DE UM OBJETO

Estado do Objeto:

Valores que seus atributos possuem no momento.

```
public class Porta{  
    public boolean aberta = false;  
}
```

Comportamento do objeto:

Comportamento está relacionado com os métodos que a classe possui:

```
public class Porta{  
    private boolean aberta = false;  
    public void fechar(){  
        aberta = false;  
    }  
    public void abrir(){  
        aberta = true;  
    }  
}
```

MENSAGENS

- × A troca de mensagens ocorre através da chamada aos métodos de um objeto por outro objeto. Exemplo:

```
× public class Relatorio {  
    ...  
    public void imprimirTitulo(String titulo){  
        formulario.Imprimir(posicaoX, posicaoY,  
        Alinhamento.CENTRAL, titulo);  
    }  
    ...  
}
```


VAMOS INCREMENTAR NOSSAS CLASSES:

- × Crie +3 métodos que alterem os estados das classes criadas anteriormente.
- × Crie +2 métodos que troquem mensagens entre as classes pessoa e cachorro.
- × PS: Se necessário crie novos atributos

HERANÇA

E POLIMORFISMO



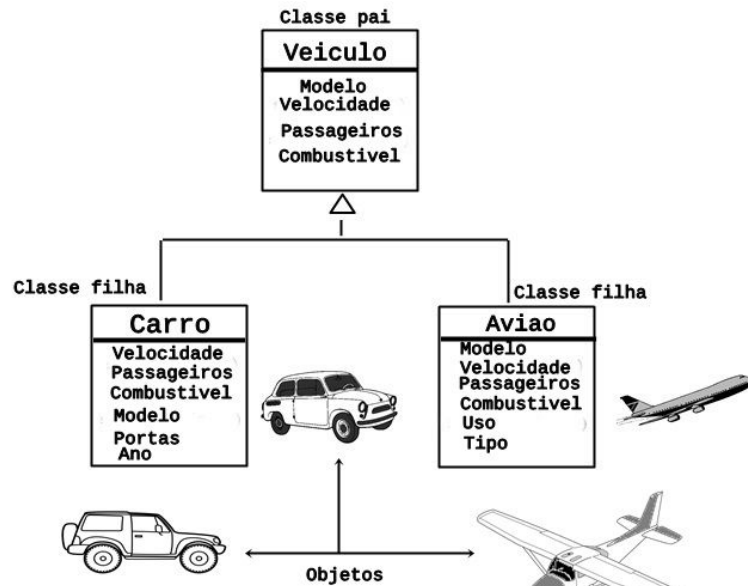
HERANÇA

- × Na Programação Orientada a Objetos o significado de herança tem o mesmo significado para o mundo real. Assim como um filho pode herdar alguma característica do pai, na Orientação a Objetos é permitido que uma classe herde atributos e métodos da outra.

HERANÇA

- × Há apenas uma restrição para a herança. Os modificadores de acessos das classes, métodos e atributos só podem estar com visibilidade **public** e **protected** para que sejam herdados.
- × Para saber se estamos aplicando a herança corretamente, realiza-se o teste “É UM”. Esse teste simples ajuda a detectar se a subclasse pode herdar a superclasse.

EXEMPLO



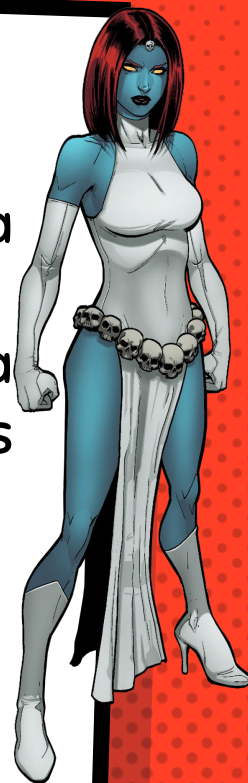
SOBREPOSIÇÃO E OCULTAÇÃO

- × Sobreposição: “Substituição de **métodos** na subclasse”;
- × Sobrescrita;
- × Sobrecarga;
- × Ocultação: “Substituição de **atributos** na subclasse”



POLIMORFISMO

- × A palavra polimorfismo vem do grego e significa aquilo que pode tomar várias formas;
- × Utilizado quando métodos herdados de uma mesma classe possuem comportamentos diferentes;
- × Para esse fim geralmente utilizamos **classes abstratas** ou **Interfaces**.
- × Exemplo conta genérica (Poupança ou Corrente)



ENCAPSULAMENTO

- × Útil pra que?
 - × Getters and Setters
- × Visibilidade:
 - × Public (Visibilidade total)
 - × Private (Visibilidade Mais Restrita)
 - × Protected (Pacote + Herança)
 - × Default (Pacote)



EXERCÍCIOS

AGORA QUE VOCÊ SABE O QUE É HERANÇA:

- × Herança carro, moto, ônibus → Veículo
- × Herança gato, cachorro (respeitando a ordem da taxonomia)
- × Herança Gerente, vendedor, secretário → Funcionário

EXERCÍCIO 1

- × Implemente a classe Funcionario e a classe Gerente.
- × **a.** crie a classe Assistente, que também é um funcionário, e que possui um número de matrícula. Sobrescreva o método `exibeDados()`.
- × **b.** sabendo que os Assistentes Técnicos possuem um bônus salarial e que os Assistentes Administrativos possuem um turno (dia ou noite) e um adicional noturno, crie as classes `Tecnico` e `Administrativo`.

EXERCÍCIO 2

- × Crie uma classe chamada Ingresso que possui um valor em reais e um método `imprimeValor()`.
- × **a.** crie uma classe VIP, que herda Ingresso e possui um valor adicional. Crie um método que retorne o valor do ingresso VIP (com o adicional incluído).
- × **b.** crie uma classe Normal, que herda Ingresso e possui um método que imprime: "Ingresso Normal".
- × **c.** crie uma classe CamaroteInferior (que possui a localização do ingresso e métodos para acessar e imprimir esta localização) e uma classe CamaroteSuperior, que é mais cara (possui valor adicional). Esta última possui um método para retornar o valor do ingresso. Ambas as classes herdam a classe VIP.

EXERCÍCIO 3

- × Crie a classe Imovel, que possui um endereço e um preço.
- × a. crie uma classe Novo, que herda Imovel e possui um adicional no preço. Crie métodos de acesso e impressão deste valor adicional.
- × b. crie uma classe Velho, que herda Imovel e possui um desconto no preço. Crie métodos de acesso e impressão para este desconto.

EXERCÍCIO 4

- × crie um assistente administrativo e um técnico. Imprima o número de matrícula e o nome de cada um deles.
- × crie um ingresso. Peça para o usuário digitar 1 para normal e 2 para VIP. Conforme a escolha do usuário, diga se o ingresso é do tipo normal ou VIP. Se for VIP, peça para ele digitar 1 para camarote superior e 2 para camarote inferior. Conforme a escolha do usuário, diga se que o VIP é camarote superior ou inferior. Imprima o valor do ingresso.
- × crie um imóvel. Peça para o usuário digitar 1 para novo e 2 para velho. Conforme a definição do usuário, imprima o valor final do imóvel.

THANKS!



CREDITS

Special thanks to all the people who made and released these awesome resources for free:

- × Presentation template by SlidesCarnival
- × Photographs by Startupstockphotos

PRESENTATION DESIGN

This presentation uses the following typographies:

- × Titles: Bangers
- × Body copy: Sniglet

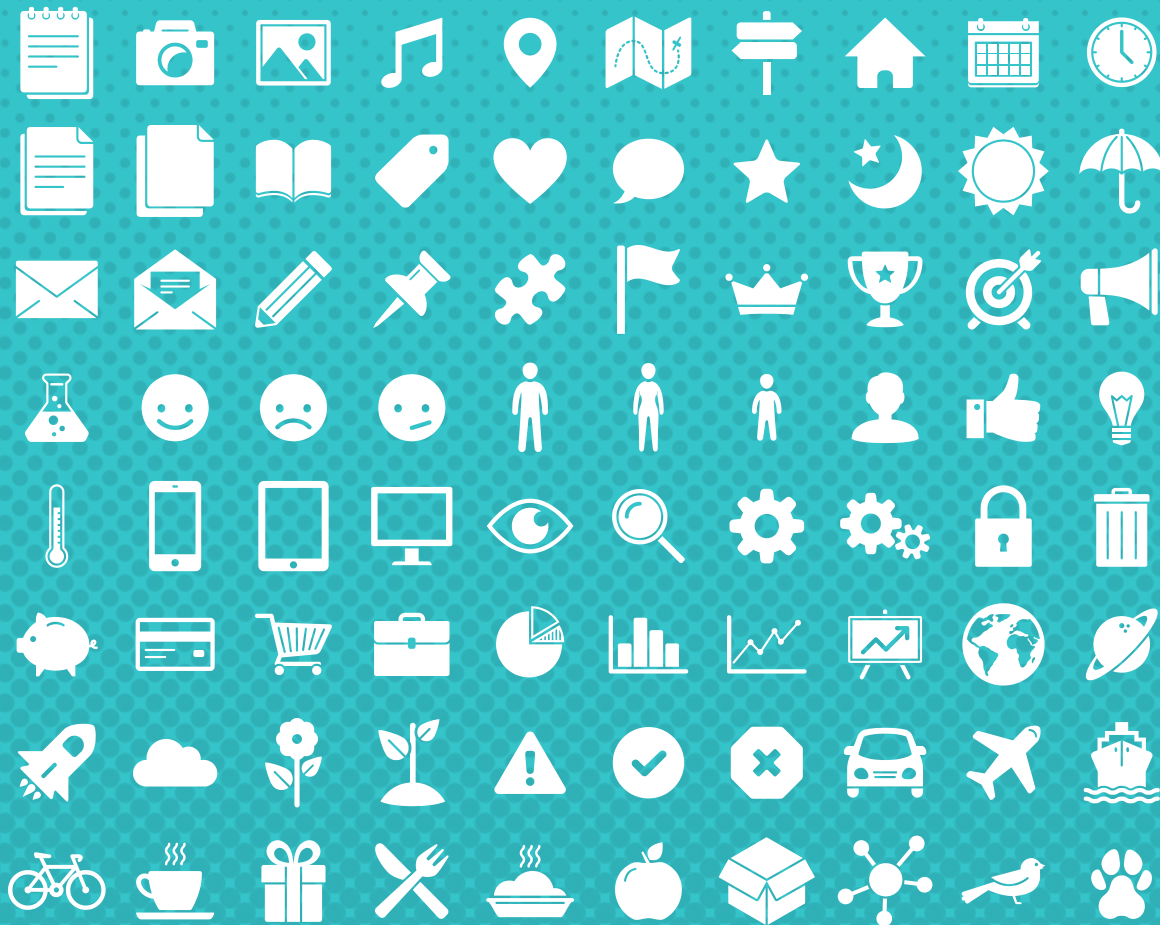
You can download the fonts on this page:

<https://www.google.com/fonts#UsePlace:use/Collection:Sniglet|Bangers>

Click on the “arrow button” that appears on the top right



You don't need to keep this slide in your presentation. It's only here to serve you as a design guide if you need to create new slides or download the fonts to edit the presentation in PowerPoint®



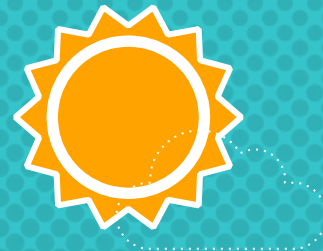
SlidesCarnival icons are editable shapes.

This means that you can:

- Resize them without losing quality.
- Change fill color and opacity.
- Change line color, width and style.

Isn't that nice?:)

Examples:



Now you can use any emoji as an icon!
And of course it resizes without losing quality and you can change the color.

How? Follow Google instructions
<https://twitter.com/googledocs/status/730087240156643328>



and many more...