

INSTITUTO FEDERAL DE
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
RIO GRANDE DO NORTE

AULA 11

PROGRAMAÇÃO ORIENTADA A OBJETOS (INTRODUÇÃO)

Disciplina: Programação Orientada a Objetos
Professora: Alba Lopes
alba.lopes@ifrn.edu.br

CONCEITOS FUNDAMENTAS

- Na computação, temos um problema a ser resolvido
 - Devemos analisar **o que** deve ser feito
 - Definir **como** deve ser feito
 - Escrever um algoritmo/programa que **implemente** a solução
- Um programa realiza operações sobre dados



CONCEITOS FUNDAMENTAS

- Modelagem Orientada a Objetos (OO)
 - O ser humano conhece o mundo e gerência sua complexidade através de objetos
 - É como desenvolvemos nossa cognição (conhecimento).
- Desenvolvemos o conceito de OBJETO
 - **Exemplos de objetos:** bola, carro, camisa, luz, casa, calça, música, conta bancária, poema, etc



CONCEITOS FUNDAMENTAS

- O que é um objeto?

“É a representação de uma coisa do mundo real.”
(BARNES,2009)

- Exemplos:

- O carro vermelho que está ali no estacionamento.
- Este lápis na minha mão.
- A peruca do Silvio Santos.



OBJETO = DADOS + OPERAÇÕES



CONCEITOS FUNDAMENTAS

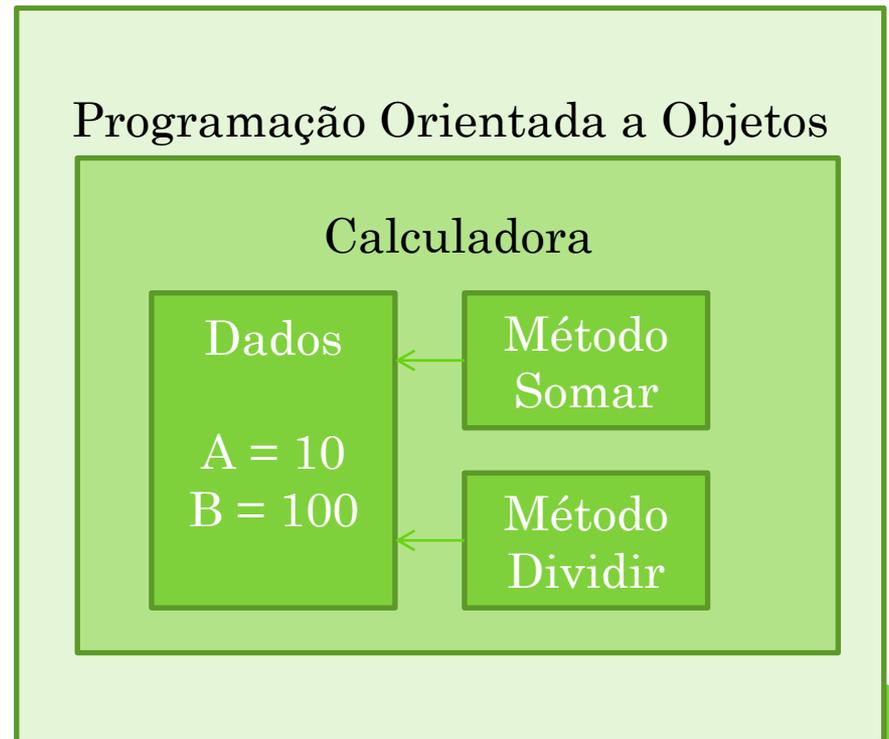
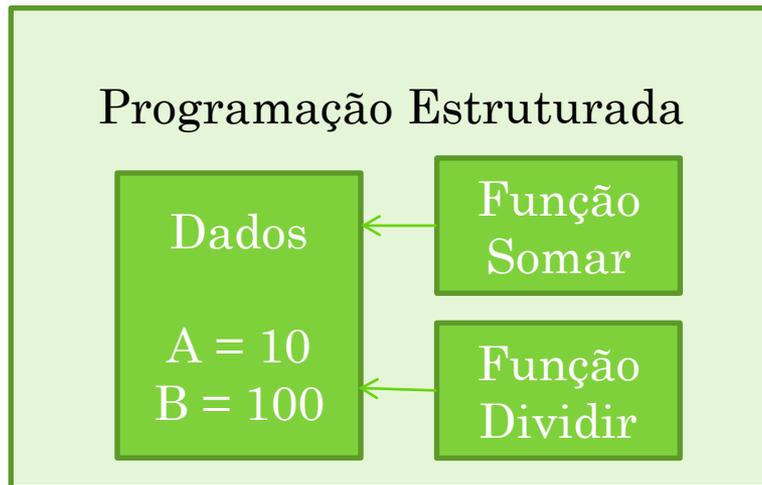
○ Objetos Possuem:

- Estado:
 - Representado pelos valores dos **atributos** de um objeto
- Comportamento:
 - Definido pelo conjunto de operações (**métodos**) do objeto;
 - Estado representa o resultado cumulativo de seu comportamento;
- Identidade:
 - Dois objetos são distintos mesmo que os valores de seus atributos sejam exatamente iguais.



CONCEITOS FUNDAMENTAS

- Enfoque estruturado x Enfoque OO
 - Exemplo: Calculadora



Agrupamento

CONCEITOS FUNDAMENTAS

- O que é uma classe?

“É um projeto de um objeto. Ela informa como cria um objeto de um tipo específico.”

(SIERRA & BATES, 2007)



CONCEITOS FUNDAMENTAS

○ Classe

- É onde conceituamos o objeto
- É a **essência** do objeto
- Define os atributos e métodos

○ Objeto

- É a **instância** de uma classe
- Objetos semelhantes pertencem a uma mesma classe



CONCEITOS FUNDAMENTAS

○ Atributos

- Atributos são as propriedades de um objeto
 - Exemplo:
 - Um objeto **carro** pode ter que propriedades?
 - **Cor**
 - **Modelo**
 - **Marca**
 - **Potência do motor**
 - **Quantidade de portas**
 - **Velocidade atual**
- Etc..



CONCEITOS FUNDAMENTAS

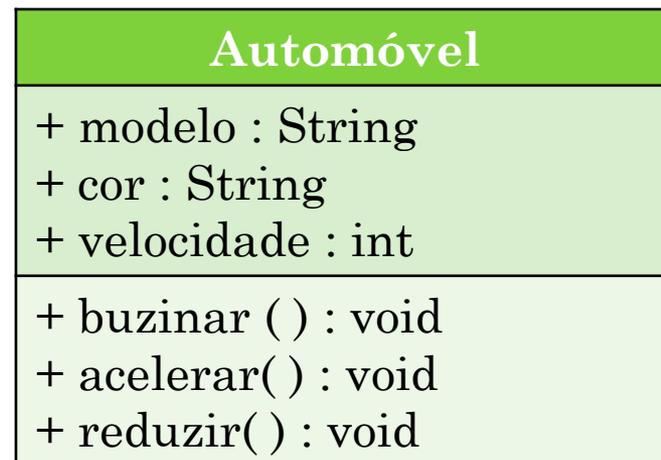
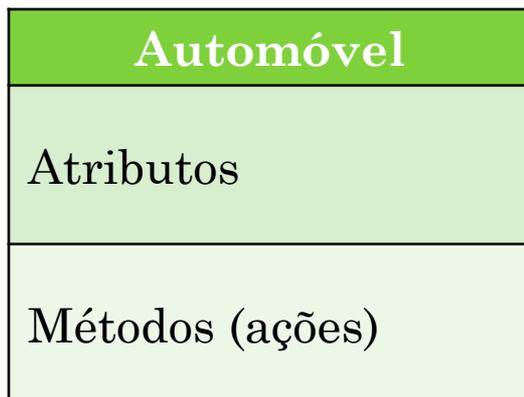
○ Métodos

- Métodos são as ações que um objeto pode realizar
 - Exemplo:
 - Um objeto **carro** pode realizar que ações?
 - **Acelerar**
 - **Frear**
 - **Buzinar**
- Etc..



CONCEITOS FUNDAMENTAS

- Como representamos uma Classe?
 - Através da UML (Unified Modeling Language)
 - Retângulo com três divisões:
 - Nome da Classe
 - Atributos
 - Métodos (ações).



CONCEITOS FUNDAMENTAS

○ O que são Instâncias?

Chamamos de instância, cada objeto criado a partir de uma classe.

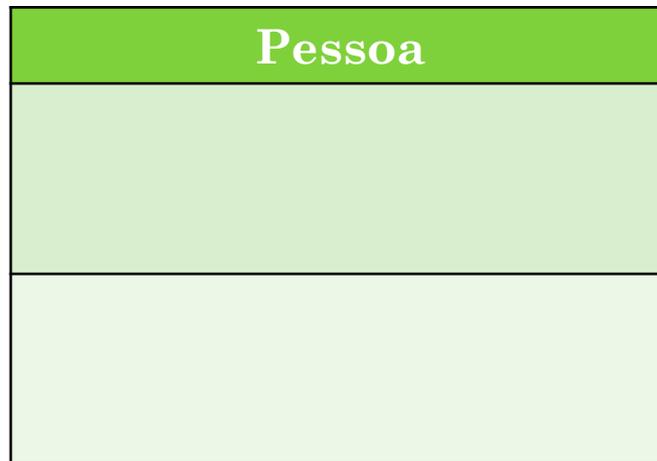
Automóvel

+ modelo : String
+ cor : String
+ velocidade : int
+ buzinar () : void
+ acelerar () : void
+ reduzir () : void



CONCEITOS FUNDAMENTAS

- Se definirmos o objeto Pessoa, que atributos ela pode ter?



CONCEITOS FUNDAMENTAS

- Se definirmos o objeto Pessoa, que atributos ela pode ter?

Pessoa
+ nome : String + idade : int + profissao :String + rg : int



CONCEITOS FUNDAMENTAS

- E em relação aos métodos?

Pessoa
+ nome : String + idade : int + profissao :String + rg : int



CONCEITOS FUNDAMENTAS

- E em relação aos métodos?

Pessoa
+ nome : String + idade : int + profissao :String + rg : int
+ andar() + falar () + trabalhar ()



EXERCÍCIOS

- Definir os atributos e métodos para as classes abaixo:
 - Conta Corrente
 - Lâmpada
 - Aluno
 - Calculadora
 - Data



CRIANDO CLASSES

- Até o momento, nossos códigos em Java eram construídos da seguinte maneira

```
<DECLARAÇÃO_DA_CLASSE>{  
  
    <MÉTODO_MAIN>{  
        <VARIÁVEIS>  
        <OPERAÇÕES>  
    }  
}
```



```
public class Calculadora{  
    public static void main(String [] args){  
        int numero1 = 10;  
        int numero2 = 20;  
        int soma;  
        soma = numero1 + numero2;  
        System.out.println("A soma é: " + soma);  
    }  
}
```



CRIANDO CLASSES

- A partir de agora, iremos abranger o nosso modo de construir as nossas classes

```
<DECLARAÇÃO_DA_CLASSE>{  
    <ATRIBUTOS>  
    <MÉTODOS>  
}
```

Nossa classe passará a ter mais métodos além do método main.



CRIANDO CLASSES

○ Definindo Atributos

- Atributo é o termo como se denomina uma variável de classe em Java
 - Para definirmos um atributo em Java, devemos identificar o tipo de dado que o representa (int, float, string, boolean, etc)
- Que tipo de dados representam os atributos abaixo da classe Automóvel?
 - **Cor**
 - **Modelo**
 - **Marca**
 - **Potência do motor**
 - **Quantidade de portas**
 - **Velocidade atual**



CRIANDO CLASSES

○ Definindo Atributos

- Que tipo de dados representam os atributos abaixo da classe Automóvel?
 - Cor **String**
 - Modelo **String**
 - Marca **String**
 - Potência do motor **float**
 - Quantidade de portas **int**
 - Velocidade atual **int**



CRIANDO CLASSES

○ Definindo atributos

- Os atributos são definidos seguindo o padrão abaixo:

```
<modificador_de_acesso> <tipo_de_dado> <nome_do_atributo> = <valor_inicial> ;
```

opcional

opcional

- Exemplo (omitindo o modificador de acesso):

```
String marca;
```

- Com modificador de acesso:

```
private String cor;
```

- Com valor inicial

```
private int velocidade = 0;
```



CRIANDO CLASSES

- Definindo atributos
 - Exemplo: classe Automovel

Automóvel

+ modelo : String
+ cor : String
+ velocidade : int

```
public class Automovel{  
  
    String modelo;  
    String cor;  
    int velocidade = 0;  
  
}
```



CRIANDO CLASSES

○ Definindo métodos

- Métodos são similares às funções em linguagens de programação
- Um método é representado por uma operação que **realiza ações** e pode **modificar** os valores dos atributos do objeto
- Pode possuir ou não uma lista parâmetros.
- Pode **retornar** algum valor ou não.
 - Caso retorne algum valor, deverá informar **que tipo de valor** o método retorna (int, float, String, boolean, etc)
 - Caso não retorne nenhum valor, o tipo informado deverá ser **void**



CRIANDO CLASSES

○ Definindo métodos

- A sintaxe de definição de métodos em Java é a seguinte:

```
<modificador_de_acesso> <tipo_de_retorno> <nome_do_método> (<lista_de_parâmetros>){  
    <operações>  
}
```

- A declaração mais simples que podemos fazer é a seguinte:

```
void <nome_do_metodo> () {  
    <operações>  
}
```

- Exemplo:

```
void acelerar () {  
    velocidade = velocidade + 1;  
}
```



CRIANDO CLASSES

- Exemplo simples de uma classe (Numero) com um atributo e um método:

```
public class Numero{  
  
    int x = 1;  
  
    void escrever () {  
        System.out.println("O valor é: "+ x);  
    }  
  
}
```



Automóvel

+ modelo : String
+ cor : String
+ velocidade : int

+ buzinar () : void
+ acelerar() : void
+ reduzir() : void

CRIANDO CLASSES

- Exemplo de código para a classe Automóvel

```
public class Automovel {
```

```
//ATRIBUTOS  
String marca;  
String cor;  
int velocidade = 0;
```

```
//MÉTODOS  
void buzinar(){  
    System.out.println("BEEEEPPPP...");  
}  
void acelerar(){  
    velocidade = velocidade + 1;  
}  
void reduzir(){  
    velocidade = velocidade - 1;  
}
```

```
}
```



INSTANCIANDO OBJETOS

- Voltamos a usar o nosso já conhecido método *main*

```
public static void main(String [] args) {  
  
  
  
  
  
  
  
  
  
}
```

```
void buzinar(){  
    System.out.println("BEEEEEEEEEEEP!!");  
}  
  
void acelerar(){  
    velocidade++;  
}  
  
void reduzir(){  
    velocidade--;  
}
```

```
public static void main(String [] args){  
    Automovel meuCarro = new Automovel();  
    meuCarro.buzinar();  
}
```

```
}
```

INSTANCIANDO OBJETOS

- Precisamos declarar uma variável qualquer como sendo do tipo da classe criada:

```
public static void main(String [] args) {  
    Automovel meuCarro;  
  
}
```

```
void buzinar(){  
    System.out.println("BEEEEEEEEEEEP!!");  
}  
  
void acelerar(){  
    velocidade++;  
}  
  
void reduzir(){  
    velocidade--;  
}
```

```
public static void main(String [] args){  
    Automovel meuCarro = new Automovel();  
    meuCarro.buzinar();  
}
```

```
}
```

INSTANCIANDO OBJETOS

- Depois atribuímos à variável criada o resultado obtido pelo instanciador **new** mais o método construtor, que é o nome da nossa classe seguido de parênteses (*mais detalhes sobre método construtor nas próximas aulas*)

```
public static void main(String [] args){
    Automovel meuCarro;
    meuCarro = new Automovel();
}
```

```
void buzinar(){
    System.out.println("BEEEEEEEEEEEP!!");
}

void acelerar(){
    velocidade++;
}

void reduzir(){
    velocidade--;
}
```

```
public static void main(String [] args){
    Automovel meuCarro = new Automovel();
    meuCarro.buzinar();
}
```

```
}
```

ACESSANDO ELEMENTOS DOS OBJETOS

- Como mencionado anteriormente, cada objeto possui seus próprios valores dos atributos e possui seu próprios métodos
- Podemos acessar os elementos (atributos ou métodos) de um objeto através da seguinte sintaxe:

```
<nome_do_objeto>.<elemento>
```



ACESSANDO ELEMENTOS DOS OBJETOS

- Para executar o método **buzinar**, por exemplo, utilizamos a seguinte sintaxe:

```
public static void main(String [] args) {  
    Automovel meuCarro;  
    meuCarro = new Automovel ();  
    meuCarro.buzinar ();  
}
```



ACESSANDO ELEMENTOS DOS OBJETOS

- Para acessar um atributo, e, por exemplo, exibir a velocidade do objeto meuCarro:

```
public static void main(String [] args) {  
    Automovel meuCarro;  
    meuCarro = new Automovel();  
    meuCarro.buzinar();  
    System.out.println("A velocidade é: " + meuCarro.velocidade );  
}
```



ACESSANDO ELEMENTOS DOS OBJETOS

- Para alterar valores dos atributos, utiliza-se a sintaxe padrão de atribuição:

```
public static void main(String [] args){  
    Automovel meuCarro;  
    meuCarro = new Automovel();  
    meuCarro.buzinar();  
    System.out.println("A velocidade é: " + meuCarro.velocidade );  
    meuCarro.marca = "Ford";  
    meuCarro.velocidade = 10;  
}
```



ACESSANDO ELEMENTOS DOS OBJETOS

- Os métodos **acelerar** e **reduzir** da classe Automovel, alteram o valor da variável velocidade.

```
public static void main(String [] args){  
    Automovel meuCarro;  
    meuCarro = new Automovel();  
    meuCarro.buzinar();  
    System.out.println("A velocidade é: " + meuCarro.velocidade );  
    meuCarro.marca = "Ford";  
    meuCarro.velocidade = 10;  
    meuCarro.acelerar();  
}
```

Após a execução dessa linha, qual o novo valor da variável velocidade?



ACESSANDO ELEMENTOS DOS OBJETOS

- Descubra o valor escrevendo o valor do atributo

```
public static void main(String [] args){
    Automovel meuCarro;
    meuCarro = new Automovel();
    meuCarro.buzinar();
    System.out.println("A velocidade é: " + meuCarro.velocidade );
    meuCarro.marca = "Ford";
    meuCarro.velocidade = 10;
    meuCarro.acelerar();
    System.out.println("A velocidade é: " + meuCarro.velocidade );
}
```



EXERCÍCIO

1. Crie uma classe **Pessoa** que obedeça a descrição apresentada na representação abaixo.

- Possua os atributos **nome** e **idade**. E os métodos: **dizerONome**, **dizerAIdade** e **fazerAniversario**.
 - O método **dizerONome** deve escrever na tela a mensagem: “Olá, meu nome é ”, e o valor do atributo **nome**.
 - O método **dizerAIdade** deve escrever na tela a mensagem: “Olá, eu tenho X anos”, onde X é o valor do atributo **idade**.
 - O método **fazerAniversario** deve adicionar 1 à idade da Pessoa
 - Crie também um método **main** que realize as seguintes operações:
 - Instancie um objeto do tipo Pessoa.
 - Atribua o seu nome ao atributo **nome**.
 - Atribua a sua idade ao atributo **idade**.
 - Chame o método **dizerONome**
 - Chame o método **dizerAIdade**
 - Chame o método **fazerAniversario**
 - Chame o método **dizerAIdade**

Pessoa
+ nome : String + idade : int
+ dizerONome() : void +dizerAIdade() : void + fazerAniversario() : void

EXERCÍCIO

2. Crie uma classe **Lampada** que obedeça a descrição apresentada na representação abaixo.

- Possua os atributos **acesa** e **potencia**. E os métodos: **acender**, **apagar**, **informarSituacao** e **informarPotencia**.
 - O método **acender** deve alterar o atributo **acesa** para *true*.
 - O método **apagar** deve alterar o atributo **acesa** para *false*
 - O método **informarSituacao** deve informar a mensagem “A luz está acesa” caso o atributo **acesa** seja igual a *true* e a mensagem “A luz está apagada” caso o atributo **acesa** seja igual a *false*
 - O método **informarPotencia** deve escrever a mensagem “A potência da lâmpara é X”, onde X é o valor do atributo **potência**
 - Crie também um método **main** que realize as seguintes operações:
 - Instancie um objeto do tipo **Lampada**.
 - Chame o método **acender**.
 - Chame o método **informarSituacao**.
 - Chame o método **apagar**
 - Chame o método **informarSituacao**
 - Chame o método **informarPotencia**

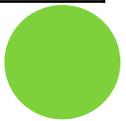
Lampada
+ acesa : boolean + potencia : int
+ acender() : void + apagar() : void + informarSituacao() : void + informarPotencia() : void

EXERCÍCIO

3. Crie uma classe **Data** que obedeça a descrição apresentada na representação abaixo.

- Possua os atributos **dia**, **mês** e **ano**. E os métodos: **escreverAData**, **escreverOMes**, **verificarAnoBissexto**, **quantoFaltaFimDoAno**.
 - O método **escreverAData** deve mostrar a data na tela no formato DD/MM/AAA
 - O método **escreverOMes** deve mostrar o nome do mês. Por exemplo , para mês 2, escrever “Fevereiro”
 - O método **verificarAnoBissexto** deve escrever uma mensagem informando se o ano é bissexto ou não.
 - O método **quantoFaltaFimDoAno** deve fazer um cálculo e escrever na tela quantos dias faltam para o fim do ano.
 - Crie também um método **main** que realize as seguintes operações

Data
+ dia : int
+ mes : int
+ ano: int
+ escreverAData(): void
+ escreverOMes(): void
+ verificarAnoBissexto(): void
+ quantoFataFimDoAno(): void



EXERCÍCIO

3. (continuação)

- Instancie um objeto do tipo **Data**.
- Atribua um valor para o atributo **dia**
- Atribua um valor para o atributo **mes**
- Atribua um valor para o atributo **ano**
- Chame o método **escreverAData**
- Chame o método **escreverOMes**
- Chame o método **verificarAnoBissexto**
- Altere o valor do atributo **ano**
- Chame o método **verificarAnoBissexto**
- Chame o método **quantoFaltaFimDoAno**
- Altere o valor do atributo **ano**
- Chame o método **quantoFaltaFimDoAno**

Data
+ dia : int + mes : int + ano: int
+ escreverAData(): void + escreverOMes(): void + verificarAnoBissexto(): void + quantoFaltaFimDoAno(): void



REFERÊNCIAS

- <http://www.hardware.com.br/artigos/programacao-orientada-objetos/>
- <http://www.fontes.pro.br/educacional/materialpaginas/java/arquivos/jdbc/jdbc.php>
- <http://www.dm.ufscar.br/~waldeck/curso/java>

