

Técnico em Desenvolvimento de Sistemas

## Lógica de Programação II

### Classes e Objetos em Java

Alex Helder Cordeiro do Rosário de Oliveira

Instituto Federal de Brasília - *Campus Brasília*

# Objetos em Java

- Assim como temos variáveis de tipos primitivos e cada variável tem seu valor próprio;
- Temos os objetos, que são “variáveis” do tipo da classe específica e cada objeto tem seu “Estado”, isto é, seu conjunto de valores em atributos.
- 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.

# Instanciamento de Objetos em Java

- Para declarar um novo objeto, usamos o nome da classe seguido do nome que desejamos dar ao objeto.

```
Random geradorDeAleatorios;  
String texto;  
Integer inteiro;
```

- Para construir um novo objeto, usamos a palavra chave `new` seguida do nome da classe à qual este objeto pertence, seguido de parenteses\*;
- Neste caso, é como se a classe fosse o “modelo” a partir do qual o objeto será criado.

```
geradorDeAleatorios = new Random();  
texto = new String();  
inteiro = new Integer();
```

---

\*Dependendo da classe, o parenteses podem vir vazios ou com argumentos. Isto será discutido mais à frente

# Acessando membros dos Objetos

- Para acessar um membro\*, usamos o nome do objeto seguido de um ponto ('.'), seguido do nome do membro.
- A convenção de código é de que o nome da classe começa com letra maiúscula, o nome de objetos, atributos e métodos deve começar com letra minúscula.
- Reconhecemos o que é um método pela presença de parenteses.

```
int numeroAleatorio = geradorDeAleatorios.nextInt();  
char c = texto.charAt(i);  
int tamanho = inteiro.SIZE;
```

---

\*Atributo ou Método.

# Individualidade dos Objetos

- Os atributos de cada objeto é independente dos atributos de outros objetos.
- O resultado da execução do método pode ter resultado diferente para cada resultado.

```
String s1 = "azul";  
String s2 = "vermelho";  
String s3 = "amarelo";  
JOptionPane.showMessageDialog(null, s2.length());  
JOptionPane.showMessageDialog(null, s1.length());  
JOptionPane.showMessageDialog(null, s3.length());  
s2 = "preto";  
JOptionPane.showMessageDialog(null, s1.length());  
JOptionPane.showMessageDialog(null, s3.length());  
JOptionPane.showMessageDialog(null, s2.length());  
JOptionPane.showMessageDialog(null, s1);  
JOptionPane.showMessageDialog(null, s3);  
JOptionPane.showMessageDialog(null, s2);
```

# Escrevendo Classes

- Declarada com a palavra-chave `class` seguida do nome da classe;
- O “corpo” da classe vêm escrito entre chaves;
- Em um mesmo arquivo fonte, pode ter uma ou mais classes;
- Só pode haver uma classe pública por arquivo, e ela têm de ter o mesmo nome do arquivo.

# Atributos

- São as características do objeto;
- Consiste no conjunto de variáveis definidas dentro da classe;
- A classe define quais os atributos e seu tipo, mas o valor é próprio do objeto;
- Sua declaração deve acontecer logo após a declaração da classe;
- Recomenda-se que sua inicialização esteja dentro da Construtora\* da classe.

---

\*As construtoras serão discutidas mais a frente

# Atributos

- Importante lembrar: variáveis locais precisam de inicialização explícita, ou não compilam;
- Atributos não precisam de inicialização explícita\*.
- Atributos não inicializados explicitamente são implicitamente inicializados com os valores abaixo:

Objeto:	<code>null</code>
Inteiro:	<code>0</code>
Ponto flutuante:	<code>0.0</code>
Booleano:	<code>false</code>
Caractere:	<code>'\u0000'</code>

---

\*Compilam e executam sem inicialização, mas é recomendável que se inicie explicitamente.



# Métodos

- São as ações que o objeto pode realizar;
- Cada método consiste em um conjunto de instruções sequenciais que podem retornar um resultado;
- Todos os objetos de uma mesma classe possuem as mesmas definições de método;
- A execução do método utiliza os atributos do objeto cujo método está sendo executado;
- Os métodos são declarados e implementados dentro de classes.

# Carro.java

```
public class Carro {  
    String cor;  
    String placa;  
    void mudaCor(String novaCor) {  
        cor = novaCor;  
    }  
    String corAtual() {  
        return cor;  
    }  
    void mudaPlaca(String novaPlaca) {  
        placa = novaPlaca;  
    }  
    String placaAtual() {  
        return placa;  
    }  
}
```

# Main.java

```
public class Main {  
    public static void main(String[] args) {  
        Carro fusca = new Carro();  
        Carro brasilia = new Carro();  
        fusca.mudaCor("preto");  
        brasilia.mudaCor("amarela");  
        fusca.mudaPlaca("ABC-1234");  
        brasilia.mudaPlaca("XYZ-5555");  
        JOptionPane.showMessageDialog(null, brasilia.corAtual());  
        JOptionPane.showMessageDialog(null, brasilia.placaAtual());  
        JOptionPane.showMessageDialog(null, fusca.corAtual());  
        JOptionPane.showMessageDialog(null, fusca.placaAtual());  
    }  
}
```