

Criação de Objetos e Acesso à Métodos



Prof. Bruno Gomes
bruno.gomes@ifrn.edu.br

Programação Orientada a Objetos

Introdução

- Classe
 - Classe é onde conceituamos o objeto
 - É a essência do objeto
 - Define os atributos e métodos
- Objeto é uma **instância de uma classe**
 - Objetos semelhantes pertencem a mesma classe

Criando um Objeto

- Operador ***new*** cria um novo objeto a partir de uma classe especificada (cria uma instância)
- Retorna uma referência para esse objeto

```
new <tipo_classe> ([parametro, parametro, ...]);
```

[] = Opcionais

< > = Identificadores e palavras reservadas

Criando um Objeto

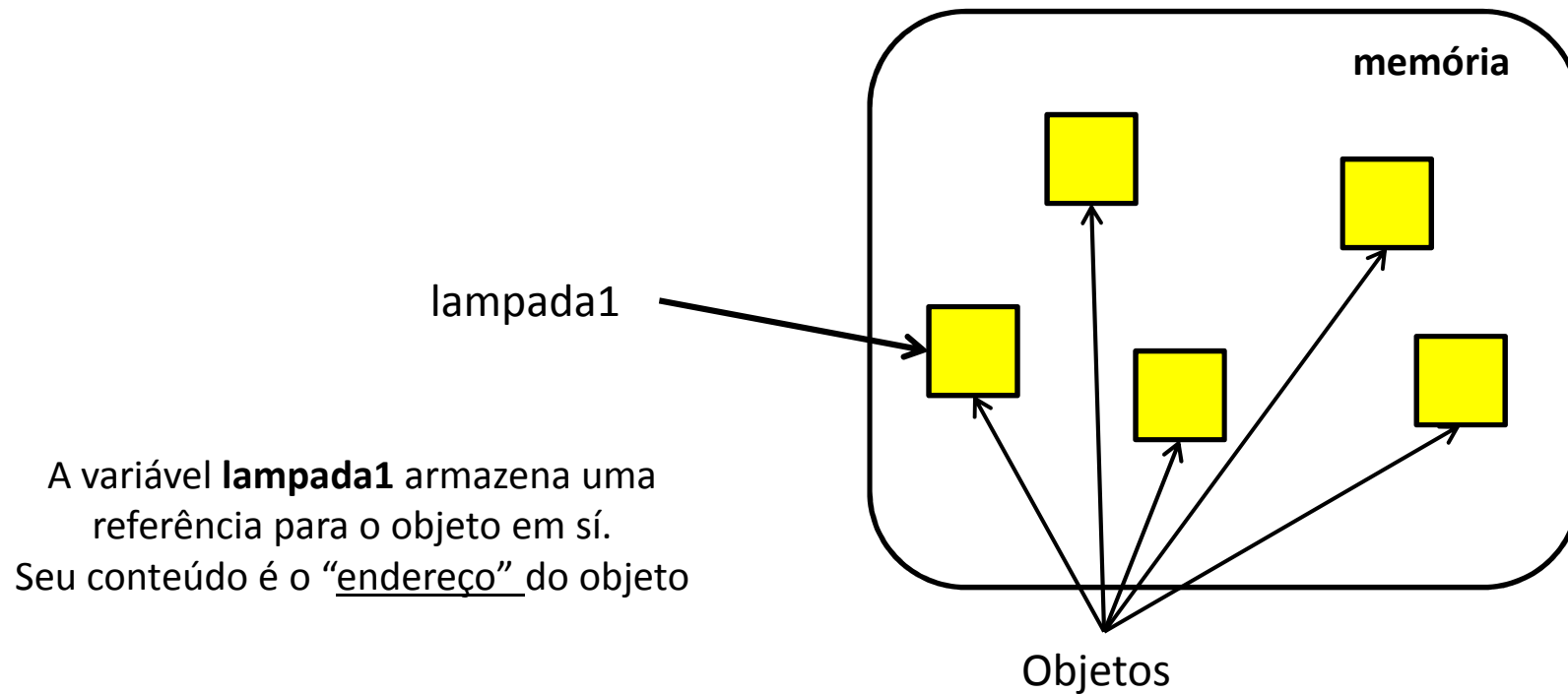
- Passos:
 - Declarar variável, associando variável a tipo (classe):
 - NomeClasse nomeVariável;
 - Ex.: Lampada lampada1;
 - Criar objeto (instanciar) e fazer variável referenciar o objeto:
 - Ex.: lampada1 = new Lampada();
 - Ex2.:Lampada lampada1 = new Lampada();

Criando um Objeto

- Ao utilizar o operador **new**:
 - Novo objeto é alocado dinamicamente na memória, e todas as suas variáveis de instancia são inicializadas com valores-padrão predefinidos.
 - **null** para variáveis objeto
 - **0** para todos os tipos básicos (exceto boolean)
 - **false** para boolean
 - O **construtor** do novo objeto é ativado
 - Após a execução do construtor, o operador **new** retorna uma **referência** (endereço de memória) para o objeto recém criado.

Criando um Objeto

```
Lampada lampada1 = new Lampada();
```

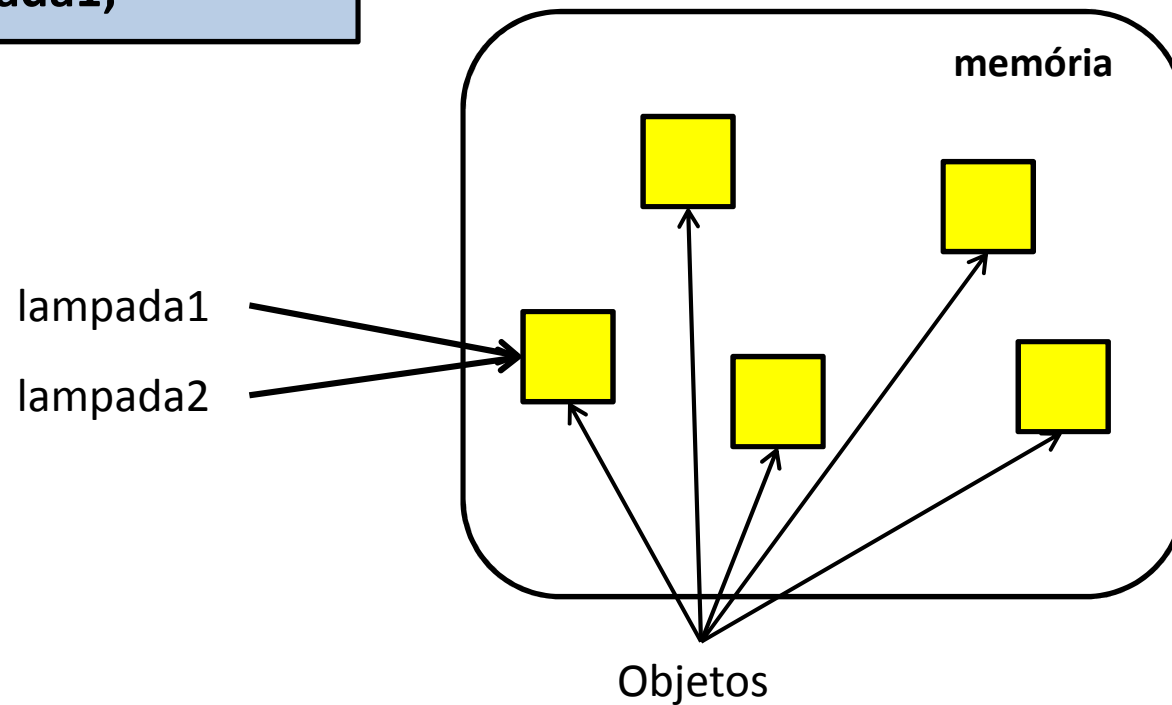


Criando um Objeto

- Observações:
 - **Lampada lampada1;**
 - Declaração da lampada1 como referência para objeto da classe Lampada
 - **lampada1 = new Lampada();**
 - Cria objeto e faz lampada1 referenciar o objeto recém-criado

Criando um Objeto

```
Lampada lampada1, lampada2;  
lampada1 = new Lampada();  
lampada2 = lampada1;
```



Acessando um Método

- Utiliza o operador “.” (ponto):
 - objeto.método();
 - Ex.: lampada.acender();
- Executa método em objeto
- Objeto deve existir
 - A variável deve referenciar objeto válido (se null, ocorre erro)
- Exemplos:
 - obj1.nomeMetodo();
 - obj1.nomeMetodo(arg1, arg2);
 - (new NomeClasse()).nomeMetodo();

Acessando um Método

Classe TesteLampada

```
public class Lampada {  
  
    public boolean estadoLampada = false;  
  
    public void acenderLampada() {  
        estadoLampada = true;  
    }  
  
    public void apagarLampada() {  
        estadoLampada = false;  
    }  
  
    public boolean verEstadoLampada() {  
        return estadoLampada;  
    }  
}
```

Classe Lampada

```
public class TesteLampada {  
    public static void main(String[] args) {  
        Lampada lampada1 = new Lampada();  
        boolean valorAtual = lampada1.verEstadoLampada();  
        System.out.println(valorAtual);  
        lampada1.acenderLampada();  
        System.out.println(lampada1.verEstadoLampada());  
        lampada1.apagarLampada();  
        System.out.println(lampada1.verEstadoLampada());  
    }  
}
```

```
false  
true  
false
```

Saída da Classe TesteLampada

Método main

- `public static void main(String args[])`
 - Método que serve de ponto inicial de um programa JAVA
 - Inicie a JVM com o nome de uma classe e ela irá executar o main da classe
 - Os parâmetros podem ser passados pela chamada

Método main

- Compile o programa abaixo e execute com a seguinte linha de comando:
 - java Aula texto

```
public class Aula {  
    public static void main(String[] args) {  
        System.out.println(args[0]);  
    }  
}
```

Método main

- Compile o programa abaixo e execute com a seguinte linha de comando:
 - java NomeClasse par1 par2 par3

```
public class Aula {  
    public static void main(String[] args) {  
        System.out.print("Quantidade de parâmetros: ");  
        System.out.print(args.length);  
        for (int i = 0 ; i < args.length ; i++){  
            System.out.println(args[i]);  
        }  
    }  
}
```

Exercício

- Criar uma classe correspondente a uma Conta de Banco
- A classe terá como atributos **nome, numero da conta, saldo**
- Terá os métodos **sacar, depositar, consultar saldo, consultar nome, alterar nome** onde:
 - Sacar -> Diminui o valor sacado do valor do saldo
 - Depositar -> Soma o valor depositado com o valor do saldo
 - Consultar Saldo -> Retorna o valor do saldo atual
 - Consultar Nome -> Retorna o nome atual
 - Alterar Nome -> Altera o nome cadastrado
- Saldo inicial será de 100 reais.

Exercício

- Criar uma nova classe que:
 - Vai conter o método **main** do java
 - Dentro do **main**, cria um objeto da classe conta bancária, e logo em seguida solicita para o usuário digitar um nome; pega o valor digitado e altera na variável **nome** do objeto da Conta Bancária, através do método **Alterar Nome**
- Após essas operações, criar um menu que exibirá 4 opções:
 - 1 – Sacar; 2 – Depositar; 3 – Consultar Saldo; 4 - Sair;

Exercício

- Se o usuário digitar 1 ou 2, então pedir para digitar o valor que irá sacar/depositar. Realiza saque/depósito, e depois exibe o novo saldo
- Se digitar 3, imprime o saldo atual
- Se digitar 4, sai da aplicação
- **Observação:** Após ter realizado a tarefa de uma das opções do menu, exibe novamente as opções ao usuário (não finaliza a aplicação, exceção para a opção 4)

Exercício

- **Dicas:**

- Para o menu, criar um **laço**, e utilizar **switch** para a verificação da opção
- Para
- Utilizar a Classe **Scanner** para a **entrada de valores**
- Utilizar **String** para armazenar o **nome**