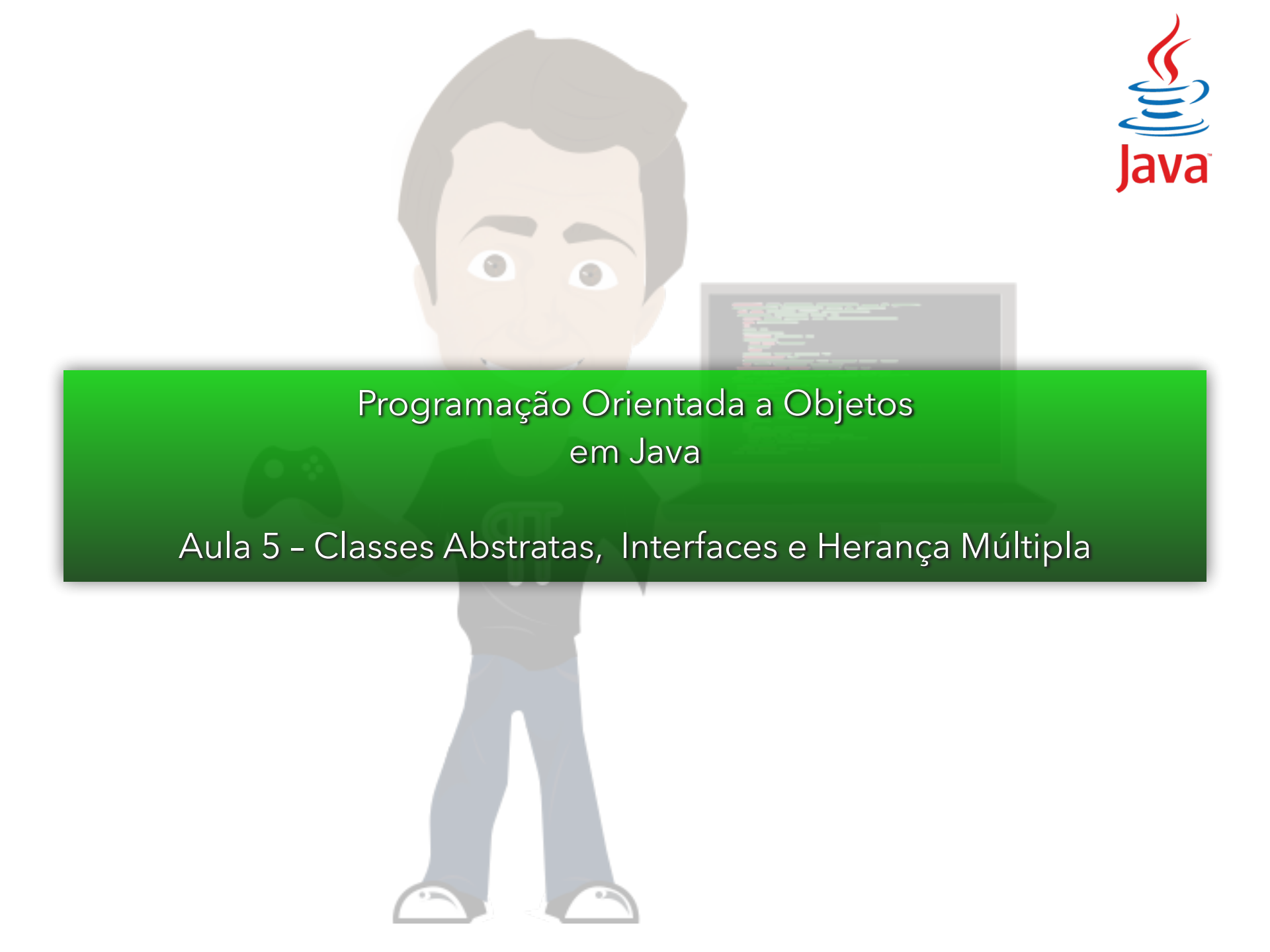

Programação Orientada a Objetos em Java

Aula 5 – Classes Abstratas, Interfaces e Herança Múltipla

Classes Abstratas

- Classes abstratas são classes que devem ser definidas com o propósito de criar apenas um modelo de implementação
- As classes abstratas não podem ter objetos instanciados
- Uso: apenas para usar uma referência genérica
- Exemplo: Classe Forma Geométrica (Aula 4)

Classes Abstratas

- As classes abstratas podem ser tornar ferramentas poderosas para a construção de sistemas complexos e que envolvam vários níveis de hierarquia de classes
- Exemplo de uso de classes abstratas
  - Sistema de folha de pagamento de instituição de ensino
    - Modo de calcular o salário de um professor não é o mesmo de um funcionário administrativo ou um estagiário
    - Entretanto todos podem ser classificados como “Funcionários”

Polimorfismo

- Mesmo com classes abstratas, o polimorfismo continua válido
- Muitas classes podem usar a referência à classe abstrata e utilizar os métodos de cada instância específica
- Algumas características:
  - Classes abstratas podem ter métodos não abstratos
  - Classes abstratas podem ter métodos abstratos
  - A classe que herda de uma classe abstratas com métodos abstratos DEVE redefinir o corpo do método
  - Classes não-abstratas não podem ter métodos abstratos
    - Se uma classe tiver métodos abstratos, a classe DEVE ser abstrata

Herança Múltipla

- Herança múltipla é um caso específico de polimorfismo
  - Uma classe possui a relação “é-um” com mais de um antecessor
  - Exemplo: Instituição de Ensino
    - Classe Funcionário
- Java não suporta Herança Múltipla!
  - Na verdade, não é possível realizar a operação **extends** com mais de uma classe

Como fazer?

- Utilizar **Interfaces**
- Interface é um “contrato” no qual o objeto “compromete-se” a implementar todos os métodos
- Interface = classe abstrata com métodos abstratos
  - Programador define interface e compilador “enxerga” uma classe abstrata com métodos abstratos
- Interface NÃO DEVE ter corpo de métodos
- Como realizar herança com interfaces?

Interfaces

- Declaração de Interface

```
public interface nome_da_interface
{
        cabeçalho_do_metodo_1(parametros);
        cabeçalho_do_metodo_2(parametros);
        cabeçalho_do_metodo_3(parametros);
}
```

Herança com Interface

- Implementando uma interface

```
public class nome_classe implements nome_da_interface
{
        cabeçalho_do_metodo_1(parametros){
                corpo do método 1
        }
        cabeçalho_do_metodo_2(parametros) {
                corpo do método 2
        }
```

Mas e a Herança Múltipla?

- A classe filha deve estender a classe pai e implementar as demais interfaces
- Neste caso é possível implementar mais de uma interface
- Lembre-se:
  - TODOS os métodos definidos nas interfaces devem ser redefinidos

Sintaxe

```
public class Classe extends classe_pai
                    implements nome_da_interface1,
                               nome_da_interface2,...
{
  ....
}
```

Exemplos

- Objeto que deve ter as características de um objeto gráfico e também de uma thread
  - Interface nativa Runnable
- Objeto de armazenamento que deve ter as características de um objeto de negócios e também ferramentas para armazenamento em bancos de dados
  - Interface DAO definida pelo usuário
- Objeto de negócios e também deve ser um elemento “serializável”
  - Interface nativa Serializable