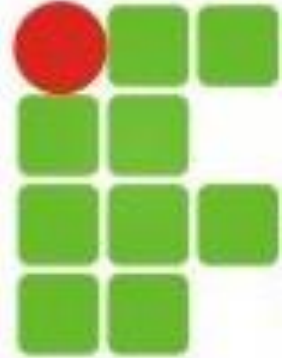


Programação Orientada a Objetos

Professor Eng. Dr. Will Roger Pereira





Objetivos da aula

- Classes Abstratas;
- Métodos Abstratos;
- Abstração no diagrama de classes.

Problematização - Abstração: Contas em um Banco



- Situação:
 - Em um banco, todas as contas são de um tipo específico: Conta corrente, conta poupança, conta salário, etc.
 - Elas podem ser modeladas, utilizando-se o conceito de herança, por exemplo, através das seguintes classes:

```
class Conta {  
    // Atributos  
    // Construtores  
    // Métodos  
}
```

```
class ContaPoupanca extends Conta {  
    // Atributos  
    // Construtores  
    // Métodos  
}
```

```
class ContaCorrente extends Conta {  
    // Atributos  
    // Construtores  
    // Métodos  
}
```

Problematização:

Contas em um Banco



- Para cada conta do domínio do banco, deve-se criar um objeto correspondente ao tipo da conta. Ex: ContaPoupança.
- Faz sentido criar objetos da classe ContaPoupança, pois existem contas poupança no domínio do banco.
- ContaPoupança → Classe concreta. Serão criados objetos dela.
- Classe Conta → Não existe de fato no banco. Serve apenas de base para outras contas.
- Não faz sentido criar um objeto da classe Conta. Não seria suficiente para representar uma conta no domínio bancário.
- Como restringir a criação de objetos da classe Conta?



Classes Abstratas

- Uma classe abstrata é desenvolvida para representar entidades e conceitos abstratos;
- A classe abstrata é sempre uma superclasse que não possui instâncias, ou seja, não existem objetos criados a partir de uma classe abstrata;
- Sua função é servir de base para classes específicas.



Classes Abstratas

- Uma classe abstrata define um modelo para uma funcionalidade e fornece uma implementação incompleta;
- A implementação de uma classe abstrata compreende somente a parte genérica dessa funcionalidade;
- Cada uma das classes derivadas, completa a funcionalidade da classe abstrata adicionando um comportamento específico.



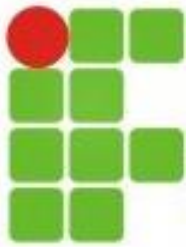
Classes Abstratas

- Classe concreta → Utilizada para instanciar objetos.
- Classe abstrata → Não é possível instanciar objetos;
→ Base para outras classes.
- Definir uma classe como abstrata → Modificador **abstract**.

```
abstract class Conta {  
    // Atributos  
    // Construtores  
    // Métodos  
}
```

- Caso seja tentada a instanciação de objetos de uma classe abstrata → ERRO!!!

Problematização: Geração de extrato



- Situação:
 - Suponha as classes do exemplo anterior;
 - É oferecido o serviço de geração de extrato para os clientes;
 - Para cada tipo de conta, as informações e o formato do extrato são diferentes;
- Problema:
 - Faz sentido ter um método de geração de extrato na classe Conta?
 - Se não houver o método de geração de extrato na classe Conta:
 - Qual a garantia de ter este método nas subclasses?
 - Qual a garantia em relação a assinatura (nome e parâmetros) deste método nas subclasses?



Métodos Abstratos

- Um método abstrato é o método contido em uma classe abstrata que não possui corpo (implementação);
- Nele está definida apenas sua assinatura.
- Deve ter o modificador **abstract**.

```
abstract class Conta {  
    // Atributos  
    // Construtores  
    // Métodos  
  
    public abstract void imprimeExtratoDetalhado();  
}
```

- Só podem existir métodos abstratos em classes abstratas;



Métodos Abstratos

- As classes concretas que derivam de uma classe abstrata, **devem** possuir os métodos abstratos implementados com a mesma assinatura. Caso contrário → ERRO!!!
- A implementação de um método abstrato é feita na subclasse, classe filha, por intermédio de sobrescrita de método (override);
- É como se uma subclasse concreta herdasse a responsabilidade de ter aquele método;
- Classes abstratas podem possuir métodos abstratos e concretos.



Abstração no diagrama de classes

- Para definir abstração em um diagrama de classes:
 - **Classe abstrata:** Nome em itálico ou um **A** à esquerda do nome da classe;
 - **Método abstrato:** Nome do método em itálico.

