



**INSTITUTO FEDERAL DE
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA**
RIO GRANDE DO SUL
Câmpus Feliz

Classes Abstratas

Prof. Moser Fagundes

Programação II
Técnico em Informática

O que é abstrair?

O que é abstrair?

... os dicionários definem abstrair como separar mentalmente, considerar isoladamente, simplificar...

Abstração

Não devemos projetar o sistema como uma grande peça **monolítica**, mas sim devemos **separar-lo em partes**.



Classes abstratas

- Neste contexto, encontraremos **classes estruturais**, que servirão de **base para outras classes**.

São classes que nunca serão instanciadas!

- Devemos "marcar" estas classes como **abstratas** para que seja **impossível** criar **objetos** a partir delas.

Classes abstratas

- Até o momento estudamos apenas **classes concretas**, a partir das quais podemos criar objetos.
- Uma **classe abstrata** é uma classe que **não pode ter instâncias diretas**, ou seja, não podemos criar objetos a partir delas. Podemos criar objetos apenas de suas descendentes.
- Para criar uma classe abstrata devemos usar a palavra-chave **abstract**.

Exemplo de classe abstrata

```
package exemplo1;
```

```
abstract public class Bicho {  
    protected String nome;
```

Palavra chave que
“marca” uma classe
como abstrata

```
    public Bicho(String nome) {  
        this.nome = nome;  
    }  
}
```

```
// Em outro arquivo Java...  
Bicho b = new Bicho("Cavalo");
```

Essa linha vai apresentar um **erro** pois não podemos criar objetos de classes abstratas.

Métodos abstratos

- Nas classes abstratas podemos incluir **métodos abstratos** que indicam apenas a declaração dos métodos, porém **não** fornecem o código dos mesmos.
- Para criar um método abstrato também devemos usar a palavra-chave **abstract** na frente da declaração do método.

Exemplo de método abstrato

```
package exemplo1;
```

```
abstract public class Bicho {  
    protected String nome;
```

```
    public Bicho(String nome) {  
        this.nome = nome;  
    }
```

```
    abstract public void emitirSom();  
}
```

Método abstrato, o qual **deve** ser implementado pelas subclasses de Bicho.

Subclasses concretas

- As subclasses de classes abstratas são criadas normalmente, usando a palavra-chave **extends**.
- A única diferença é que devemos **implementar os métodos abstratos** definidos na superclasse abstrata.

Exemplo de subclasse

```
package exemplo1;
```

Chamada ao construtor da
superclasse abstrata

```
public class Cachorro extends Bicho {  
    protected String raca;
```

```
    public Cachorro(String nome, String raca) {  
        super(nome);  
        this.raca = raca;  
    }
```

```
    public void emitirSom() {  
        System.out.println("Grrr!");  
    }  
}
```

Implementação do método abstrato,
porém sem a palavra-chave **abstract**.

Exemplo de instância da subclasse concreta

Agora posso criar um bicho da classe cachorro, pois **Cachorro** é concreta.

```
Cachorro c = new Cachorro("Sadan", "Bulldog");
```

Métodos concretos em classes abstratas

- Dentro de uma classe abstrata, também podemos incluir métodos concretos (aqueles que tem código).
- A herança destes métodos concretos ocorre normalmente.

Exemplo de método concreto

```
package exemplo1;
```

```
abstract public class Bicho {  
    protected String nome;
```

```
    public Bicho(String nome) {  
        this.nome = nome;  
    }
```

```
    public String getNome() {  
        return this.nome;  
    }
```

Este método concreto
estará disponível nas
subclasses

```
    abstract public void emitirSom();  
}
```

Perguntas

- Dê exemplos nos quais poderíamos usar classes e métodos abstratos?
- Em um projeto desenvolvido em equipe, quais vantagens decorrem do uso de classes e métodos abstratos?

Exercícios

Lista de exercícios no Moodle.