

Unidade 3

Seção 2

Programação Orientada a Objetos

Webaula 2

Definição e uso de classes abstratas.

Nesta webaula, trataremos das classes abstratas, sua relação com herança e abordaremos brevemente herança múltipla.

Classes abstratas

A classe abstrata é uma classe que não permite a geração de instâncias a partir dela, isto é, não permite que sejam criados objetos. Ao contrário, uma classe concreta permite a geração de instâncias.

Usando classes abstratas, o desenvolvedor pode declarar classes que definem somente parte de uma implementação, deixando para as classes estendidas o oferecimento de implementações específicas.

Exemplo

Imagine que estamos modelando a classe `Pessoa`, com os atributos: nome, data de nascimento, endereço e estado civil, entre vários outros. Se considerarmos a necessidade da divisão do modelo “pessoa” em outros dois (pessoa física e pessoa jurídica, por exemplo), chegaríamos à conclusão que o ente “pessoa” é capaz de agrupar alguns atributos comuns aos dois subtipos de pessoa e, por isso, pode

servir de base para a geração de pessoa física e pessoa jurídica. Assim, na condição de classe que servirá apenas como plataforma para a criação de outras, a classe `Pessoa` pode ser declarada como **abstrata**:

```
public abstract class Pessoa
{

    // corpo da classe.

}
```

Há, contudo, a possibilidade de que sejam criados também **métodos abstratos**, que compartilham comportamentos com outros objetos. Dessa maneira, cada método não implementado na classe abstrata também é indicado como `abstract`, embora esse expediente possa ser efetivado por meio das interfaces.

Exemplo de declaração de método abstrato é:

```
public abstract void imprime();
```



Características de classes abstratas

A característica principal das classes abstratas é não poderem passar por instanciação. Além disso, há a possibilidade de definirmos um comportamento padrão – mas com certo grau de especificidade – para um grupo de outras classes.

Veja o exemplo adaptado de Furgeri (2013):

Considere a superclasse `Veiculo` e as subclasses `Automovel` e `Aviao`. A superclasse `Veiculo`, que é abstrata, implementa (ou seja, efetivamente codifica) os métodos `ligar`, `desligar` e `mostrarStatus`, que representam comportamentos comuns, tanto para um automóvel, quanto para um avião.



Contudo, o método `acelerar` não é implementado na classe `Veiculo`. Ao invés disso, ele é apenas definido.

Temos então, que a classe abstrata `Veiculo` define um método abstrato chamado `acelerar`, que deve, obrigatoriamente, ser implementado em todas as subclasses de `Veiculo`, pois o comportamento de `acelerar` se efetiva de modos diferentes para um avião e para um veículo.

Esboço do código que implementa a classe
Veiculo e seus métodos

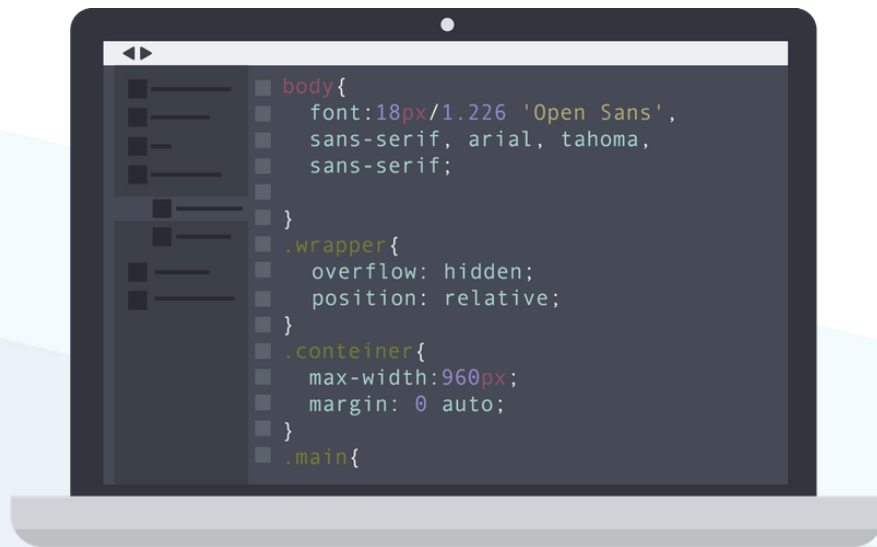
```
public abstract class Veiculo {  
    public int velocidade;  
    public boolean status;  
    public void ligar() {  
        status = true;  
    }  
    public void desligar() {  
        status = false;  
    }  
    public void mostrarStatus() {  
        System.out.println(status);  
    }  
    public abstract void acelerar();  
}
```

Fonte: adaptado de Furgeri (2013).

*Explore a galeria para ver o
conteúdo:*

Deste código podemos extrair
que:

✓ Não há método `main()`
definido. Portanto, do jeito que
se encontra, o código não poderá
ser executado.



Relação entre classes abstratas e herança

Sabemos que as classes abstratas servem como base para criação de outras classes, que conterão dados e comportamentos comuns. No entanto, este efeito não poderia ser conseguido simplesmente pelo uso da herança, na forma como a conhecemos?

CAELUM (s.d.) argumenta que a única diferença é que não se pode instanciar um objeto do tipo classe abstrata, o que já confere mais consistência ao sistema.

A utilização das classes abstratas deve se basear no critério de necessidade ou não de instanciação da classe. Essa escolha depende, via de regra, da situação que se pretende implementar, e a decisão ficará por conta de uma boa avaliação da situação.

Definição de herança múltipla: Santos (2003) nos ensina que o mecanismo de herança funciona em apenas um sentido: da classe pai para a classe filha. Explore a galeria para ver a relação entre as classes Pessoa e Paciente:

Pessoa
- nome
- identidade
- nascimento
- cadastraPessoa()

Paciente
- planoDeSaude
- consultaPlano()

Fonte: adaptado de Santos (2003).

Nesta webaula, vimos o conceito de classe abstrata e descobrimos que trata-se de uma classe que não permite a geração de instâncias a partir dela, isto é, não permite que sejam criados objetos.

Você já conhece o Saber?

Aqui você tem na palma da sua mão a **biblioteca digital** para sua **formação profissional**.

Estude no celular, tablet ou PC em qualquer hora e lugar sem pagar mais nada por isso.

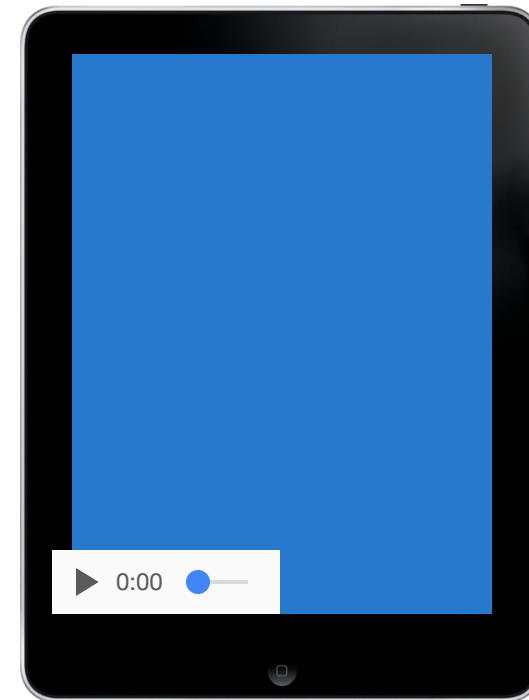
Mais de 450 livros com interatividade, vídeos, animações e jogos para você.



Android:
<https://goo.gl/yAL2Mv>



iPhone e iPad - IOS:
<https://goo.gl/OFWqcq>





Bons estudos!

