Programação Orientada a Objetos

Objetos e Classes

2021/01

Sumário

Classes e Objetos

- Objetos: partes do modelo do domínio de determinado problema.
- Objetos são criados (instanciados) a partir de classes.
- A classe descreve o tipo do objeto; os objetos representam instanciações individuais da classe.

Classes e Objetos

```
// Declara e inicializa a variável x,
// do tipo int (inteiro), com o valor 10.
int x = 10;

// Declara e instancia o objeto s,
// da classe String, com o valor "abc".
String s = new String("abc");
```

Classes e Objetos

- Por convenção, nomes de classes possuem a letra inicial maiúscula; nomes de objetos, letra inicial minúscula.

- É recomendável a utilização de *camel case* nos nomes de classes e objetos. Exemplos: ItemPedido, CategoriaProduto (classes), itemPedido, dataNascimento (objetos).

Métodos

- Objetos se comunicam por meio da chamada (invocação) de métodos.
- Por exemplo, o método *length* da classe *String* retorna o tamanho (número de caracteres) de determinado objeto.

```
// Instancia a string "minhaString"
String minhaString = new String("abcde");

// A variável n recebe o valor 5
int n = minhaString.length();
```

Parâmetros

- Parâmetros são utilizados para fornecer informações adicionais à invocação de um método.

- Por exemplo, o método *replace* da classe *String* permite substituir os caracteres informados como parâmetros em determinado objeto.

```
// Instancia o objeto "original", com o conte
   údo "abcdef"
String original = new String("abcdef");

// Substitui os caracteres 'c' pelo caractere
    'x'.
// O objeto "nova" possuirá como valor "
   abxdef"
String nova = original.replace('c', 'x');
```

Parâmetros

- Cada parâmetro possui um tipo, que define os tipos de valores que o mesmo pode assumir.

- A linguagem Java possui tipos primitivos (int, double, boolean) e classes *wrapper* para esses tipos (Integer, Double, Boolean).

Parâmetros

- As classes *wrapper* permitem, dentre outros, a conversão entre tipos e a atribuição de valor nulo (*null*).

```
// Converte a String 5 para o inteiro 5
String s = new String("5");
int n1 = Integer.parseInt(s);

// Atribui o valor null ao objeto n2
Integer n2 = null;

// Erro: tipo primitivo não pode receber null
int n3 = null;
```

Parâmetros

- O cabeçalho de um método é denominado assinatura.

- Na assinatura do método, são apresentadas as informações necessárias para sua invocação, tais como tipo de retorno, nome e parâmetros (tipo e ordem).

```
// O método "replace" recebe dois caracteres
// e retorna a string correspondente à
// substituição de "oldChar" por "newChar"
// String replace(char oldChar, char newChar)

// O método startsWith recebe uma string e
// retorna true se o objeto começa com
// "prefix" ou false caso contrário
// boolean startsWith(String prefix)
```

Construtores

- Construtores são tipos especiais de métodos, utilizados para se instanciar objetos de uma classe.

- Construtores possuem o mesmo nome da classe, tipo de retorno implícito (objetos da classe) e, opcionalmente, parâmetros de inicialização.

```
// Cria uma string vazia
// String s1 = new String();

// Cria a string "abc"
// String s2 = new String("abc");
```

Estado

- Objetos têm estado, que é representado pelos valores de seus campos.
- Alguns métodos, quando invocados, alteram o estado dos objetos.
- Objetos da mesma classe têm os mesmos campos em termos de número, tipo e nome. Os valores dos campos, entretanto, podem ser diferentes para cada objeto.

Exemplo

```
public class Aluno {

   private Integer matricula;
   private String nome;

   public Integer getMatricula() {
      return matricula;
   }

   public void setMatricula(Integer matricula) {
      this.matricula = matricula;
   }

   // continua ...
```

Exemplo

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

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

Exemplo

```
// Instancia o objeto aluno1
Aluno aluno1 = new Aluno();

// Altera o campo número de matrícula
aluno1.setMatricula(123456);

// Altera o campo nome
aluno1.setNome("Aluno Um");
```

Configuração do Ambiente de Desenvolvimento

- JDK: `https://www.oracle.com/br/java/technologies/javase-jdk11-downloads.html`.
- Eclipse: `https://www.eclipse.org/downloads/`.
- Scene Builder: `https://gluonhq.com/products/scene-builder/`.
- PostgreSQL: `https://www.postgresql.org/download/`.

Referências

- Barnes, D. J., Kolling, M. *Programação orientada a objetos com Java*, 4ª edição. São Paulo, Pearson Prentice Hall, 2009.

- Deitel, P.; Deitel, H. *Java: como programar*, 8ª edição. São Paulo, Pearson Prentice Hall, 2010.