Programação Orientada a Objetos

Definições de Classe

2021/01

Sumário

Definições

Campos

Construtores

Métodos

Instruções Condicionais

Exemplo

Exercícios

Referências

Campos, construtores e métodos

- Campos: armazenam dados para o uso de cada objeto.
- Construtores: permitem configurar um objeto no ato de sua criação.
- Métodos: implementam o comportamento dos objetos.

Campos, construtores e métodos

```
public class NomeDaClasse {

    // Campos

    // Construtores

    // Métodos

}
```

Campos

```
public class Conta {
    private Integer agencia;

    private Integer numero;

    private String titular;

    private Integer saldo;

    // ...
}
```

Campos

- O modificador de acesso *private* restringe o acesso ao campo apenas para objetos da mesma classe.

- O tipo *Integer* é um *wrapper* para o tipo inteiro.

Construtores

```
public class Conta {
    // campos omitidos ...

    public Conta(Integer agencia, Integer numero,
        String titular) {
        this.agencia = agencia;
        this.numero = numero;
        this.titular = titular;
        this.saldo = 0;
    }

    // ...
}
```

Construtores

- Uma classe pode possuir vários construtores, desde que os parâmetros sejam diferentes (tipo, número e/ou ordem).
- O construtor possui o mesmo nome da classe, e seu tipo de retorno é implícito (a própria classe).
- Com raras exceções (padrão de projeto *singleton*, por exemplo), o construtor é público, permitindo que objetos sejam instanciados por objetos de outras classes.
- A palavra reservada *this* é utilizada para indicar o campo, resolvendo a ambiguidade entre este e o parâmetro.

Métodos

```
public class Conta {
    // campos e construtores omitidos ...

    public void deposito(Integer valor) {
        saldo += valor;
    }

    public void saque(Integer valor) {
        saldo -= valor;
    }
    // ...
}
```

Métodos

- Os especificadores de acesso (*public*, *private* e *protected*) definem quais objetos podem realizar a invocação do método.
- O tipo *void* indica a ausência de retorno.
- Assim como os construtores, uma classe pode possuir vários métodos com o mesmo nome, desde que os parâmetros sejam diferentes.

Métodos de acesso

- Retornam o valor dos campos:

```
public class Conta {
   // campos e construtores omitidos ...

   public Integer getSaldo() {
      return saldo;
   }

   public String getTitular() {
      return titular;
   }

   // ...
}
```

Métodos modificadores

- Alteram os valores dos campos:

```
public class Conta {
   // campos e construtores omitidos ...

   public void setAgencia(Integer agencia) {
      this.agencia = agencia;
   }

   public void setTitular(String titular) {
      this.titular = titular;
   }
   // ...
}
```

Instruções Condicionais

```
public class Conta {
   // campos e construtores omitidos ...

   public void saque(Integer valor) {
      if (valor > saldo) {
         System.err.println("Saldo insuficiente.");
         return;
      }

      saldo -= valor;
   }
   // ...
}
```

Exemplo

1. Completar a classe *Conta*, acrescentando um método que imprima os campos da mesma.

2. Criar uma aplicação que permita ao usuário interagir com a classe *Conta*, instanciando objetos e invocando seus métodos.

Exercícios

1. Barnes, D. J., Kolling, M. *Programação orientada a objetos com Java*: página 40, exercícios 2.59 - 2.67.

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.