

POO - PROGRAMAÇÃO ORIENTADA A OBJETOS

Construtores e Associação Entre Classes

Rodrigo R Silva

Instituto Federal de Educação, Ciência e Tecnologia Sul-Rio-Grandense
Campus Bagé

Nesta Aula Veremos...

1 Introdução

2 Construtores

3 Associação Entre Classes

Introdução

Contextualizando - Declaração de Objetos

- Após a definição da estrutura de uma classe podemos criar aplicações Java que manipulam objetos.
- Objetos são declarados em Java da mesma forma que declaramos variáveis de tipo primitivo.
- Depois de declarada a variável que referencia um objeto, devemos instanciá-la.
- A instanciação é realizada pelo operador **new**.

```
package classe.executavel;
import classes.java.Conta;

public class ClasseExecutavel {

    public static void main(String[] args) {
        int matricula;
        float nota;
        Conta conta;
        conta = new Conta();
    }
}
```

Declara uma variável denominada **conta** a qual é um tipo da classe **Conta**.

Contextualizando - Criação de objetos

- O operador `new` aloca um espaço na memória, com tamanho suficiente para manter todos os atributos definidos na classe de um objeto.
- Ex: para um objeto da classe `Conta`, será alocado um espaço suficiente para manter: código (int), correntista (String) e saldo (float) da conta.
- Para isso, o operador **new** invoca uma operação definida na classe do objeto, cuja finalidade é realizar a criação do objeto.
- A criação do objeto envolve a alocação de memória, bem como, a inicialização dos valores dos atributos.

Construtores

Construtores

```
package classe.executavel;  
import classes.java.Conta;  
  
public class ClasseExecutavel {  
  
    public static void main(String[] args) {  
        int matricula;  
        float nota;  
        Conta conta = new Conta();  
        conta.depositar(500.00f);  
    }  
}
```

Conta() é um construtor da classe **Conta**. Esta operação que irá criar o objetos **conta** na memória.

Conceito de construtor

- O construtor é uma operação descrita no interior de uma classe, cuja única finalidade é criar um objeto na memória.
- Notem que o construtor não é um método convencional.
- Trata-se de uma operação com algumas características específicas.
- Conta conta = **new** Conta ();
- O que há de curioso no construtor?

Características do Construtor

- Por ser uma operação com o único propósito de criar objetos de uma classe, o construtor tem as seguintes características:
- tem o mesmo nome da classe;
- o construtor não tem retorno, nem mesmo void;
- por ser público, privado ou protegido;
- por convenção é a primeira operação descrita em uma classe (depois dos atributos).

Escrita na Classe

```
package classes.java;  
public class Conta {  
    private int codigo;  
    private String correntista;  
    private float saldo;  
  
    public Conta() {  
    }  
}
```

Construtor da classe **Conta** (Construtor padrão - default) sem argumentos e códigos.

Criando Objetos do Tipo Conta

- Uma aplicação Java que cria um objeto do tipo Conta e, posteriormente, inicializa os atributos.

```
package classe.executavel;  
import classes.java.Conta;  
  
public class ClasseExecutavel {  
  
    public static void main(String[] args) {  
        int matricula;  
        float nota;  
        Conta conta = new Conta();  
        conta.setCodigo(112233);  
        conta.setCorrentista("Ciclano Asteróides");  
        conta.setSaldo(500.00f);  
        conta.depositar(500.00f);  
    }  
}
```

Construtor da classe
Conta

Ajusta os valores
dos atributos

- Mas é possível criar o objeto e já informar os valores dos atributos em uma única instrução?

Criando Novos Construtores

- Podemos escrever novos construtores em uma classe.

```
package classes.java;
public class Conta {
    private int codigo;
    private String correntista;
    private float saldo;

    public Conta() {
    }

    public Conta(int codigo, String correntista, float saldo) {
        this.codigo = codigo;
        this.correntista = correntista;
        this.saldo = saldo;
    }
}
```

Construtor padrão da classe **Conta**

Novo construtor da classe **Conta**

- Segundo construtor de Conta:
- possui uma assinatura diferente do construtor.
- inicializa os atributos da classe.

Sobrecarga de Métodos

- A sobrecarga é a reescrita de um método no interior de uma classe.
- Sobrecarga pode ser aplicada para métodos e construtores de uma classe.
- O que muda em métodos sobrecarregados:
- a assinatura dos métodos.

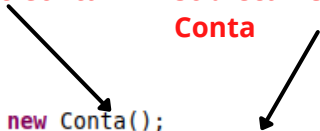
1 `public Conta()`

2 `public Conta(int codigo, String correntista, float saldo)`

Usando Métodos Sobrecarregados

Criando um objeto **conta1** usando o construtor **default** da classe **Conta**

Criando um objeto **conta2** usando o construtor **sobrecarregado** da classe **Conta**



```
Conta conta1 = new Conta();  
Conta conta2 = new Conta(332211, "Fulano", 850.00f);  
conta1.setCodigo(112233);  
conta1.setCorrentista("Ciclano Asteróides");  
conta1.setSaldo(500.00f);  
conta1.depositar(500.00f);
```

Associação Entre Classes

Objetos como Atributos

- Em determinados modelos de classes alguns atributos também são objetos de outra classe.
- Analisando a estrutura da classe Conta:
- Notamos que o atributo correntista na verdade é mais complexo.
- Na realidade, um correntista é também um objeto, visto que possui características.
- Portanto, a classe Conta não pode ser modelada na forma:

```
package classes.java;  
public class Conta {  
    private int codigo;  
    private String correntista;  
    private float saldo;  
  
    public Conta() {  
  
    }  
}
```


Tornando o Correntista um Objeto

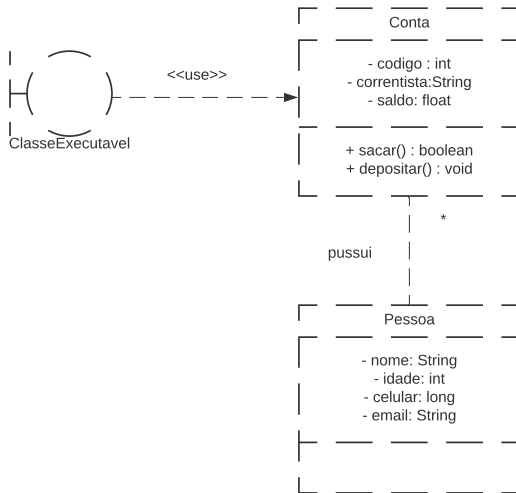
- Sendo assim, criaremos uma classe para representar um correntista de uma conta.
- A classe será denominada Pessoa.

```
package classe.modelo;  
public class Pessoa {  
  
    private String nome;  
    private int idade;  
    private long celular;  
    private String email;  
  
    public Pessoa() {  
  
    }  
  
}
```

Associação Entre Classes

- Trata-se de uma relação entre dois elementos.
- Dois elementos da mesma classe (autorelaciomanento);
- Ou classes diferentes.
- No modelo de classes da aplicação bancária surgirá uma relação entre duas classes:
 - Pessoa
 - Conta

Representação Gráfica



A classe Conta

- A classe Conta passa a ser relacionada a Pessoa.

```
package classes.java;  
public class Conta {  
    private int codigo;  
    private Pessoa correntista;  
    private float saldo;  
  
    public Conta() {  
  
    }  
}
```

A classe ClasseExecutavel(Aplicação)

- Uma aplicação Java manipulando objetos do tipo Conta. Agora com a relação de Conta com Pessoa.

```
package classe.executavel;
import classes.java.Conta;

public class ClasseExecutavel {

    public static void main(String[] args) {

        Pessoa pessoa = new Pessoa();

        pessoa.setNome("Ciclano");
        pessoa.setIdade(38);
        pessoa.setCelular(999999999);
        pessoa.setEmail("ciclano@gmail.com");
        Conta conta = new Conta(332211, pessoa, 850.00f);

        System.out.println("Número: "+conta.getCodigo());
        System.out.println("Correntista: "+conta.getCorrentista().getNome());
        System.out.println("Saldo R$ "+conta.getSaldo());

    }
}
```

OBRIGADO!