Programação Orientada a Objetos

Classes e Objetos

Prof. Leandro Rodrigues Pinto
<leandrorodp@gmail.com>

Classes

- A Classe é o molde, a planta, o esquema, o modelo a ser seguido pelos objetos;
- A planta da casa é o modelo que as casas construídas terão;
- Porém não é possível morar na planta da casa, apenas na casa já construída;
- A Classe define as características da casa e as funções que ela terá: parte elétrica, hidráulica, saneamento e etc;

Objetos

- Objetos são utilizados para representar conceitos do mundo real
- Objetos seguem fielmente as especificações de suas Classes

- Os Objetos são instâncias concretas das Classes
- As casas são instâncias concretas das plantas que lhes deram origem

Representação

Casa

José

Carro

Mundo Real

Projeto

Casa

Mora

José

Dirige

Carro

Orientação a Objetos

Vantagens:

Flexibilidade - Reusabilidade - Robustez - Modularidade

Objetos – Atributos – Métodos

Objetos:

São entidades concretas ou abstratas.

Tem características (**Atributos**) e podem executar ações (**Métodos**).

OBJETOS = DADOS + OPERAÇÕES

Orientação a Objetos

Objetos – Atributos – Métodos

Atributos:

São dados relevantes para a entidade que definem as **características** do objeto.

O objeto que representa a entidade “JOSÉ”, deve possuir essas informações.

- Um **atributo** é uma variável que pertence a um **objeto**.
- Os dados de um **objeto** são armazenados nos seus **atributos**.

Orientação a Objetos

Objetos – Atributos – Métodos

Atributos:

```
[<modificadores_atributo>] <tipo_atributo> <nome_atributo> [= valor_inicial];
```

Exemplo:

```
public String marca = “Java”;

int numero = 0;

boolean estadoCurso = false;
```

Orientação a Objetos

Objetos – Atributos – Métodos

Métodos:

São as operações, ações realizadas pelo próprio objeto.

Os métodos também são utilizados para possibilitar interações entre os objetos de uma aplicação.

- As tarefas que um **objeto** pode realizar são definidas pelos seus **métodos**.
- Um **objeto** é composto por **atributos** e **métodos**.

Orientação a Objetos

Objetos – Atributos – Métodos

Métodos:

```
[<modificadores_método>] <tipo_retorno> <nome_método> ([<parâmetros>]){
        // corpo do método
}
```

Exemplo:

```
public static void main(String[] args) {
    System.err.println("Corpo do Método");
}
```

Classes

- Antes de um objeto ser criado, devemos definir quais serão os seus **atributos** e **métodos**. Essa definição é realizada através de uma **classe**.
- A partir de uma classe, podemos construir objetos (**instanciar**) na memória do computador que executa a nossa aplicação.

Modelam os objetos definindo:

Tipos de dados que o objeto armazena, ou seja, os estados possíveis que ele pode assumir (**Atributos**).

Tipos de operações que podem ser executadas pelo objeto, ou seja, o seu comportamento (**Métodos**).

Classes

- Podemos representar uma classe através de diagramas UML. O diagrama UML de uma classe é composto pelo nome da mesma e pelos atributos e métodos que ela define.

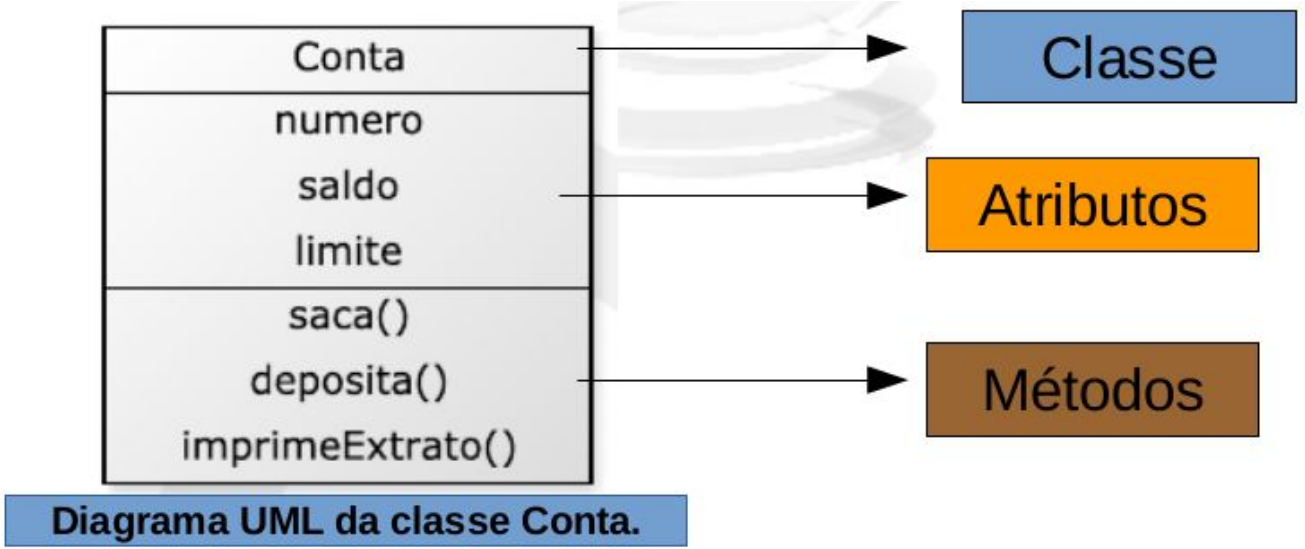


Diagrama UML da classe Conta.

Classes

- Declaração de uma classe em java:

```
[<modificadores_da_classe>] class <nome_da_classe>
        [extends <nome_superclasse>]
        [implements <interface_1>, <interface_2>, ...] {

        // variáveis e métodos da classe
}
```

Objetos

Criando objetos em Java:

- Após definir a classe, podemos criar objetos a partir dela. Esses objetos devem ser alocados na memória RAM do computador.

- Todo o processo de alocação do objeto na memória é gerenciado pela máquina virtual.

- O gerenciamento da memória é um dos recursos mais importantes oferecidos pela máquina virtual.

Objetos

Criando objetos em Java:

- Do ponto de vista da aplicação, basta utilizar um comando especial para criar objetos e a máquina virtual se encarrega do resto.
- O comando para criar objetos é o new.
- O operador new cria um novo objeto a partir de uma classe especificada.
- Retorna uma referência (endereço de memória) para esse objeto.

```
new <tipo_classe> ([parametro_1, parametro_2, ...]);
```

```
new Scanner(System.in);
```

Construtores

O que são

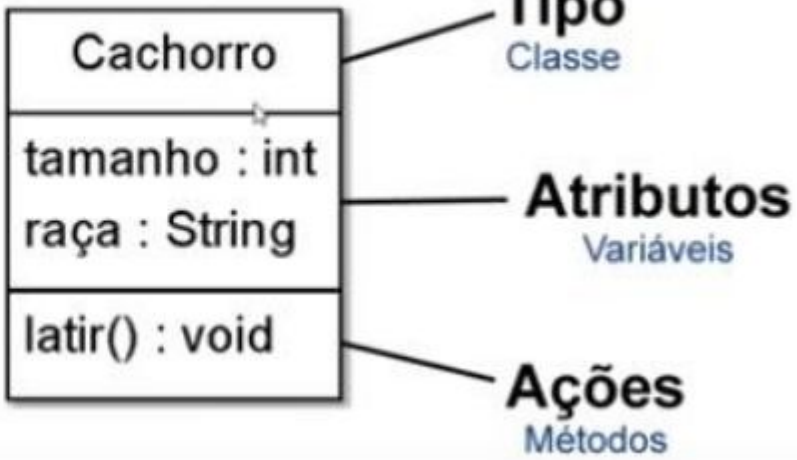
- Construtores são utilizados basicamente para inicializar as variáveis de uma classe
- O Java cria automaticamente o construtor padrão Classe(), que não recebe nenhum parâmetro e inicializa as variáveis com seus valores padrão como 0 e NULL
    - **Variáveis numéricas recebem zero, lógicas recebem false e objetos recebem null.**
- Para inicializar as variáveis com valores diferentes, precisamos criar nossos próprios construtores

Construtores

- O Construtor é um método:
  - Obrigatório para toda classe
  - Utilizado para inicializar as variáveis da classe
  - Que pode chamar outros métodos
  - Sem tipo de retorno
  - Que aceita parâmetros
  - Que aceita sobrecarga

OBS.: ao criar qualquer construtor, o Java não criará o construtor padrão Classe()

Atividade prática 1



Atividade prática 2

Crie uma classe que possua pelo menos 3 atributos de tipos diferentes

Crie pelo menos 4 construtores:

- Um que receba cada um dos 3 parâmetros separadamente
- Um que recebe os 3 parâmetros juntos
- Crie uma classe de teste que demonstre a utilização dos 4 construtores
    - OBS.: não esqueça de utilizar os SET e GET

Atividade prática 3

Faça um programa que solicite os dados de 3 pessoas, configure através de métodos SET e depois mostre na saída do console através de System.out.println() e métodos GET.

A sua classe pessoa deve ter pelo menos 5 atributos, e pelo menos três tipos diferentes de dados.

Bibliografia

Cadenhead, Rogers; LEMAY, Laura, Aprenda em 21 dias Java 2. 4. ed. São Paulo: Campus, 2005.

DEITEL, Harvey H .; DEITEL, Paul J. Java: como Programar. 8. Ed. São Paulo: Pearson / Prentice-Hall, 2010.

Perguntas?

Obrigado!