

Fatec Mococa

Linguagens de Programação

- Classes
- Objetos e instâncias
- Referências

Sandra Cristina Costa Prado

Classes Java

Exemplo: Escrever um programa que defina a classe Produto com os **campos** descricao, qtdEstoque e precoUnit.

```
class Produto {  
    String descricao;  
    int qtdEstoque;  
    double precoUnit;  
}
```

Campos são variáveis que

- são declaradas como membros da classe ou
- são declaradas fora de qualquer método ou construtor dentro da classe.

Classes Java

Exemplo: Escrever um programa que defina a classe Produto com os campos descricao, qtdEstoque e precoUnit.

```
class Produto {  
    String descricao;  
    int qtdEstoque;  
    double precoUnit;  
}
```

Para armazenar os dados do produto é preciso criar uma **instância** da classe Produto. Esta **instanciação** é feita em uma outra classe.

Classes Java

Exemplo: Escrever um programa que defina a classe Produto com os campos descricao, qtdEstoque e precoUnit.

```
class Produto {  
    String descricao;  
    int qtdEstoque;  
    double precoUnit;  
}
```

Para armazenar os dados do produto é preciso criar uma **instância** da classe Produto. Esta **instanciação** é feita em uma outra classe.

```
public class PapelariaClasse {  
    public static void main(String[] args) throws FileNotFoundException {  
  
        Produto prod = new Produto(); //instanciação da classe Produto  
  
        prod.descricao = "caneta";  
        prod.qtdEstoque = 10;  
        prod.precoUnit = 5.5;  
  
        System.out.printf("%10s %10d %10.2f\n", prod.descricao,  
                           prod.qtdEstoque, prod.precoUnit);  
  
    }  
}
```

Classes Java

Portanto, é preciso escrever, pelo menos, duas classes no programa.

Nos exemplos, os objetos serão instanciados na classe que contém o método `main()`.

Declaração

Produto prod;

A variável **prod** não armazenará os dados propriamente, mas sim uma referência de onde os dados serão armazenados.

prod armazena null.

Instanciação

```
Produto prod = new Produto();
```

O operador **new** retorna uma referência (endereço de memória). A referência é a indicação de onde o **objeto** está armazenado

prod não é mais null, armazena um endereço de memória

Onde estão os objetos?

Quando uma variável é do tipo de uma classe, diz-se que ela referencia um objeto. Informalmente, ela é chamada de objeto, embora não armazene o objeto de fato.

Uma razão para este comportamento é que um objeto pode ser muito grande. É mais eficiente armazenar sua localização do que o próprio objeto.

A referência **prod** fica numa região chamada stack e o objeto referenciado por **prod** fica numa região chamada heap.

Construtor

Produto prod = new Produto();



Classe

Define os objetos.
É um arquivo de texto.

Construtor

Tem o mesmo nome da classe.
Executa (realiza, torna real)
o que está especificado na classe.

Referência para o objeto

O objeto é o ator principal, pois é ele que

- 1) Armazena os dados
- 2) Realiza as operações sobre os dados

Classes e objetos

Classe é um conceito estático: uma classe é um elemento reconhecido num texto de programa.

Objeto é um conceito puramente dinâmico: não pertence ao texto do programa, mas à memória do computador, um local que ele ocupa durante a execução do programa.

As características do objeto (dados que deve armazenar, operações que deve realizar, como deve interagir com outros objetos, etc.) são definidas na classe a que ele pertence.


Classes e Objetos

Objeto é uma unidade básica da Programação Orientada a Objetos e representa entidades da vida real.

Um Objeto é uma instância de uma Classe.

Quando uma classe é definida, nenhuma memória é alocada, mas quando é instanciada (ou seja, um objeto é criado), memória é alocada.

Um objeto possui identidade, estado e comportamento. Cada objeto contém dados e código para manipular os dados.



Objetos podem interagir sem precisar conhecer detalhes dos dados ou código um do outro, é suficiente conhecer o tipo de mensagem aceita e o tipo de resposta retornada pelos objetos.

Um objeto que é criado usando uma classe é chamado de instância dessa classe. Às vezes, dizemos que o objeto pertence à classe. As variáveis que o objeto contém são chamadas de variáveis de instância. Os métodos que o objeto contém são chamados de métodos de instância.

Objeto vs Instância vs Referência

Um objeto é uma instância de uma classe e ocupa um espaço na memória do computador, armazenando seus dados e seus métodos (chamados de comportamentos).

Tecnicamente, "instância" refere-se à ideia conceitual ou lógica de um objeto específico, enquanto "objeto" refere-se à implementação física desse conceito, ocupando espaço na memória do computador.

Embora eles não sejam estritamente a mesma coisa do ponto de vista técnico, na prática cotidiana da programação, é comum usar "instância" e "objeto" de forma intercambiável para descrever um objeto específico criado a partir de uma classe.

Referência é uma variável que aponta para um objeto. Significa que o valor que ela armazena é o endereço de memória onde está alocado o objeto apontado por ela.

O objeto existe quando a classe está em uso, ou seja, quando foi alocada memória para armazenar o que está definido ou especificado nela. Para alocar memória para os seus dados (atributos) e seus métodos (comportamentos) e para poder acessá-los, criamos uma referência usando a palavra-chave new. O new cria uma referência para um objeto de uma classe e aloca memória para ele.