
PROGRAMAÇÃO ORIENTADA A OBJETOS

POO

*Todos os conceitos fazem parte do livro: Conceitos de Computação Com o Essencial de
JAVA. Cay Horstmann. 2005

Por que utilizar P00?

- Fácil representação da vida real;
- Reutilização de código;
- Fácil manutenção futura ou da equipe;
- Organização mais clara do código;
- Segurança.

O que é um objeto?

Um objeto é uma entidade que você pode manipular em seu programa.

Deve-se pensar em um objeto como uma “caixa-preta”. Com uma interface pública e uma implementação oculta.

Normalmente, o objeto usa de seus *métodos para manipular o programa*.

O que é uma classe?

A classe define os métodos para os objetos.

A classe é responsável por todas propriedades (variáveis) e métodos (funções) que um objeto instanciará.

O que é um método?

Método é uma funcionalidade que o objeto tem.

Exemplo: um carro pode *acelerar*. Acelerar é uma funcionalidade que todo carro deve ter.

Métodos podem ou não receber argumentos (parâmetros) .

Retornos de um método

Um método pode ou não retornar uma resposta.

Exemplo: Eu posso ter um método para adicionar um item no pedido. Este método não precisa de retorno.

Mas também posso ter um método para ver o valor do pedido. Este método precisa de um retorno.

Quando um método não tem retorno, usamos a palavra "void" na sua construção. Já quando ele tem retorno, precisamos definir o tipo que esse método retornará

Diferença entre variáveis e objetos

Variáveis de tipos primitivos são criados já na declaração.

“Variáveis” de objeto são do tipo composto e precisam ser instanciados para iniciarem.

“Variáveis” de objeto fazem referência ao endereço de memória do objeto e não ao valor em si.

Construtor

É usado no momento que instanciamos um novo objeto. O padrão é sem argumentos.

Com ele podemos definir como deve ser a inicialização do objeto

A palavra this

É usada quando queremos referenciar a propriedade da nossa classe.

Comumente usamos os argumentos com os mesmos nomes que a propriedade do campo. A palavra this indica ao compilador que estamos acessando a propriedade da nossa classe e não o argumento.

Sobrecarga

Sobrecarga é o nome utilizado para métodos que possuem a mesma assinatura (nome), porém o retorno ou argumentos são diferentes.

Um exemplo muito usado de sobrecarga é método `.println`, que pode receber tipos como `string`, `double`, `int`.

Exercício 1

Criar uma classe Jogador que tenha as propriedades nome, gols, assistências. E os métodos adicionarGol e adicionarAssistencia que receba como argumento as quantidades de gols/assistências.

Também criar dois métodos para ver o número de gols e assistências

Exercício 2

Criar uma classe aluno que receba um nome, nota1, nota2 e nota3, onde o somatório das notas podem chegar no máximo 100.

nota1 (max 30)

nota2 (max 30)

nota3 (max 40)

Cria um método que retorne o nome do aluno, sua média, a maior e menor nota