

Java

interpretada
e compila.

OO
fortemente
tipada
garbage
colector

OO

abstração

acomplamento

atributos

classes

encapsulamento

polimorfismo

sobrecarga

herança

UML

triangulo

-A double

-B double

-C double

Area() double

reaproveitamento de código

anti DAD

Programa para ler os dados de um produto em estoque : nome, preço, quantidade no estoque

Em seguida:

--Mostrar dados do produto (nome , preço, quantidade de estoque

--Realizar uma entrada um entrada no estoque e mostrar novamente os dados do produto

--Realizar uma saída no estoque e mostrar novamente os dados do produto

EX

Entre os dados do produto

Nome: TV

Preço: 900.00

Quantidade: 10

Produto dados:

Tv, \$ 900.00 , 10 unidades , Total \$ 9000.00

Produto
- NOME - STRING
- PREÇO - double
- QUANT - int
TotalValueInStock() : double

AddProduto, RemoverProduto

Construtor

* É uma operação especial da classe que executa no momento de instanciação do programa

Usos comuns

--Iniciar valores e atributos

--Permitir ou obrigar que o obj receba dados / dependencia de instanciação(injeção de dependencia)

--Se um construtor não for especializado vai ser inicializado o construtor padrão

product p = new Product();

-É possível especificar mais de um construtor na mesma classe
sobrecarga

-nome
-preço

-
quantidade

|| ||
O
O

Exemplo de sobrecarga + soma e concatena

Passar o proprio objeto como argumento na chamada de um metodo ou construtor

```
public class ChessMatch{
```

constructor

```
.....
```

```
placeNewPiece('e', 1, new King(board, color :white, this));
```

obj

Encapsulamento

--É um princípio que busca esconder detalhes de implementação de uma classe, expondo apenas o que é seguro

--Boa prática

O objeto sempre deve estar em estado consistente e a própria classe deve sempre garantir isso

O objeto não deve expor nenhum atributo

Os atributos devem ser acessados por meio do get e set

Como implementar ?

```
private String name;  
private double preco;
```

```
public String getNome(){  
    return nome;  
}
```

```
public void setNome(String nome){  
    this.nome = nome;  
}
```

```
public String getPreco(){  
    return preco;  
}
```

```
public void setPreco(double preco){  
    this.preco = preco;  
}
```