

Construtor

- É uma operação especial da classe, que executa no momento da instanciação do objeto
- Usos comuns:
 - Iniciar valores dos atributos
 - Permitir ou obrigar que o objeto receba dados / dependências no momento de sua instanciação (injeção de dependência)
- Se um construtor customizado não for especificado, a classe disponibiliza o construtor padrão:

```
Product p = new Product();
```
- É possível especificar mais de um construtor na mesma classe (sobrecarga)

Problema exemplo

Enter product data:

Name: **TV**

Price: **900.00**

Quantity in stock: **10**

Product data: TV, \$ 900.00, 10 units, Total: \$ 9000.00

Enter the number of products to be added in stock: **5**

Updated data: TV, \$ 900.00, 15 units, Total: \$ 13500.00

Enter the number of products to be removed from stock: **3**

Updated data: TV, \$ 900.00, 12 units, Total: \$ 10800.00

Product
- Name : string
- Price : double
- Quantity : int
+ TotalValueInStock() : double
+ AddProducts(quantity : int) : void
+ RemoveProducts(quantity : int) : void

```

package application;

import java.util.Locale;
import java.util.Scanner;

import entities.Product;

public class Program {
    public static void main(String[] args) {
        Locale.setDefault(Locale.US);
        Scanner sc = new Scanner(System.in);

        Product product = new Product();
        System.out.println("Enter product data: ");
        System.out.print("Name: ");
        product.name = sc.nextLine();
        System.out.print("Price: ");
        product.price = sc.nextDouble();
        System.out.print("Quantity in stock: ");
        product.quantity = sc.nextInt();

        System.out.println();
        System.out.println("Product data: " + product);

        System.out.println();
        System.out.print("Enter the number of products to be added in stock: ");
        int quantity = sc.nextInt();
        product.addProducts(quantity);

        System.out.println();
        System.out.println("Updated data: " + product);

        System.out.println();
        System.out.print("Enter the number of products to be removed from stock: ");
        quantity = sc.nextInt();
        product.removeProducts(quantity);

        System.out.println();
        System.out.println("Updated data: " + product);

        sc.close();
    }
}

```

```

package entities;

public class Product {

    public String name;
    public double price;
    public int quantity;

    public double totalValueInStock() {
        return price * quantity;
    }

    public void addProducts(int quantity) {
        this.quantity += quantity;
    }

    public void removeProducts(int quantity) {
        this.quantity -= quantity;
    }

    public String toString() {
        return name
            + ", $ "
            + String.format("%.2f", price)
            + ", "
            + quantity
            + " units, Total: $ "
            + String.format("%.2f", totalValueInStock());
    }
}

```