

Programação Orientada a Objetos

Módulo 1

Vaux Gomes ¹

¹Instituto Federal de Educação, Ciência e Tecnologia do Ceará
Campus Jaguaribe

7 de Agosto de 2022

Sumário

Sumário

Introdução

- Paradigma de programação

História

Tecnologia de objetos

- Vantagens

- Desvantagens

Fundamentos da Linguagem Java

- Variáveis

- Entrada e Saída de Dados

Tente isso

Introdução

Paradigmas de programação

Paradigma de programação

Um **paradigma de programação** é um estilo, maneira, modelo, de programar.

Alguns paradigmas importantes

- ▶ Programação funcional
- ▶ Programação estruturada
- ▶ Programação orientada a objetos

Introdução

Paradigmas de programação

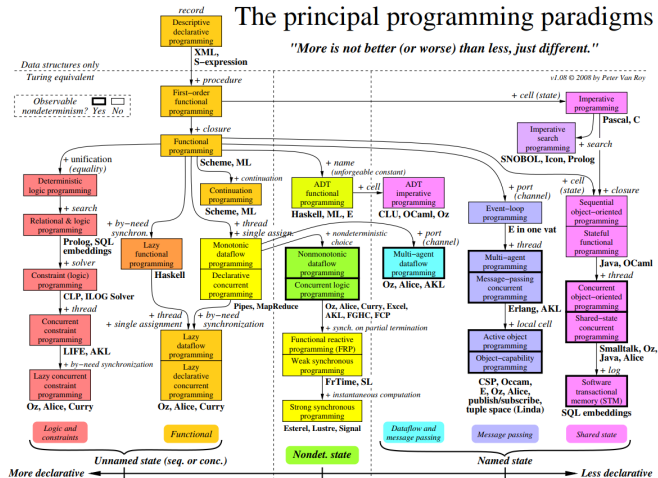


Figura 1: Fluxograma dos principais paradigmas de programação.

História

História dos paradigmas

1. Programação de baixo nível
 - ▶ Linguagem de máquina
 - ▶ Varia de computador para computador
2. Programação linear
3. Programação estruturada
 - ▶ Primeiros sistemas
4. Programação modular
5. **Programação orientada a objeto (POO)**

História

História dos paradigmas



História

Alan Kay

Alan Kay

- ▶ Nascimento: 1940
- ▶ Graduado em **matemática e biologia molecular**
- ▶ Pesquisador Xerox (1971)
 - ▶ Ethernet
 - ▶ Impressora laser
 - ▶ Modelo cliente-servidor
- ▶ Dynabook (laptop para crianças)
 - ▶ Smalltalk

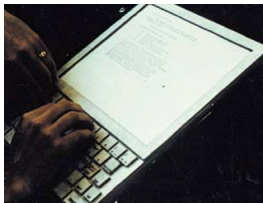
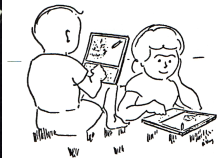


FIGURE 11.27 Children with Dynabooks from "A Personal Computer for Children of All Ages" [Kay 1972]



Tecnologia de objetos

Por que utilizar?

- ▶ A técnica de programação orientada a objetos (POO) é **utilizada por programadores para desenvolver a maioria das aplicações que utilizamos hoje em dia.**
- ▶ Programas orientados a objetos são muitas vezes **mais fáceis de entender, corrigir e modificar.**

Tecnologia de objetos

Vantagens

- ▶ São fáceis de entender.
- ▶ Uma vez que tudo é tratado como objeto, nós podemos modelar conceitos do mundo real usando POO.
- ▶ A POO oferece o **reuso de classes**.
 - ▶ Nós podemos reutilizar classes criadas anteriormente (por nós ou por outros) sem a necessidade de reescrevê-las novamente.
- ▶ A **programação pode ocorrer em paralelo**. Como resultado é possível desenvolver programas completos rapidamente.
- ▶ Os programas escritos em POO são mais **fáceis de testar, gerenciar e atualizar**.
- ▶ É uma **técnica de desenvolvimento segura**, uma vez que os dados são escondidos e não podem ser acessados por classes externas.

Tecnologia de objetos

Desvantagens

- ▶ O **relacionamento entre as classes** pode se tornar artificial por natureza.
- ▶ O programador deve **planejar o design do programa** antes mesmo de começar a programar.
 - ▶ Em POO tudo é tratado como objeto, então o programador precisa **desenvolver habilidades de desenho de solução em termos de objetos** além de desenvolver suas habilidades de programação.
- ▶ Em geral, **o tamanho dos programas são maiores**, por exemplo, que os de uma abordagem estruturada.
 - ▶ Programas maiores implicam em mais instruções a serem executadas e **isto reflete na velocidade de execução dos programas**.

Fundamentos da Linguagem Java

Variáveis

- ▶ Variáveis são *containers* para armazenar dados/valores.
- ▶ Os tipos são importantes porque informam ao compilador quanto de memória ele deve separar para cada dado

Fundamentos da Linguagem Java

Variáveis

- ▶ Em java existem diferentes tipos de variáveis:
 - ▶ **String**: Textos como "Hello World"
 - ▶ No Java usamos sempre **aspas duplas** para declarar Strings e
 - ▶ **Aspas simples** para declarar caracteres.
 - ▶ **int**: Números inteiros como 123 ou -123
 - ▶ **float**: Números fracionários (pontos flutuantes) e reais como 19.99 ou -19.99
 - ▶ Lembrando que a escrita é feita em inglês, logo **os números possuem um ponto para separar as casas decimais**
 - ▶ **char**: Armazena um caractere como 'a' ou 'B'
 - ▶ **boolean**: Armazena um Booleano cujo valor pode ser true ou false

Fundamentos da Linguagem Java

Variáveis

- ▶ A **declaração** de variáveis implica na **especificação do tipo** da variável

```
// Classe principal
public class Main {

    // Função principal
    public static void main(String[] args) {
        String nome = "Bob";
        int idade = 20;
        float saldo = 11.59;
        char primeiraLetraNome = 'B';
        boolean casado = false;
    }
}
```

Fundamentos da Linguagem Java

Variáveis

Atenção

O Java exige ponto e vírgula no fim dos comandos ';' e também um gerenciamento de chaves '{}' que não existem em linguagens como Python.

Fundamentos da Linguagem Java

Entrada e Saída de Dados

- ▶ As funções de leitura e impressão de dados no console são particular de cada Sistema
- ▶ O java disponibiliza uma classe System presente em qualquer parte do código para que possamos imprimir nossas variáveis no console

```
public class Main {  
    public static void main(String[] args) {  
        String nome = "Bob";  
        int idade = 20;  
  
        // Olá, meu nome é: Bob. Eu tenho: 20 anos.  
        System.out.println("Olá, meu nome é: " + nome +  
            ". Eu tenho: " + idade + " anos.");  
    }  
}
```

Fundamentos da Linguagem Java

Entrada e Saída de Dados

- ▶ A entrada de dados não é tão trivial quanto em outras linguagens de programação
- ▶ Seguem duas formas de pegar dados do usuário. Para ambos exemplos é necessário fazer o `import` de algumas bibliotecas

Fundamentos da Linguagem Java

Entrada e Saída de Dados

```
import java.util.Scanner;

public class Main {
    public static void main(String[] args) {
        Scanner in = new Scanner(System.in);

        System.out.print("Digite seu nome: ");
        String nome = in.nextLine();

        System.out.print("Digite sua idade: ");
        int idade = in.nextInt();

        System.out.println("Olá, meu nome é: " + nome +
            ". Eu tenho: " + idade + " anos.");
    }
}
```

```
Digite seu nome: Bob
Digite sua idade: 20
Olá, meu nome é: Bob. Eu tenho: 20 anos.
```

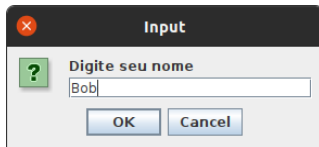
Fundamentos da Linguagem Java

Entrada e Saída de Dados

```
import javax.swing.*;

public class Main {
    public static void main(String[] args) {
        String nome = JOptionPane.showInputDialog("Digite seu nome");
        int idade = Integer.parseInt(JOptionPane.showInputDialog("Digite sua idade"));

        System.out.println("Olá, meu nome é: " + nome +
            ". Eu tenho: " + idade + " anos.");
    }
}
```



Tente isso

O som viaja a aproximadamente 335 metros por segundos. Então, se você viu o relâmpago e escutou o trovão depois 7.2 segundos, é possível saber a distância entre você e o local onde o relâmpago aconteceu.

1. Crie um arquivo/classe chamado **Sound.java**
2. Adicione o método *main* na classe
3. Crie uma variável **dist** do tipo **double** para armazenar a distância entre o relâmpago e você
4. Calcule o valor da distância assumindo que você escutou o trovão 7.2 segundos após ver o relâmpago
5. Imprima o valor na tela