Programar em Linguagem Orientada a Objetos Básica

Prof. Arthur M. Araújo

UNIFACISA
CENTRO UNIVERSITÁRIO

01

Conteúdo

- Introdução a Java
- Estrutura de uma aplicação Java
- Sugestões
  - Instalar o Java
  - IDE
- Primeiro código
- Revisar estruturas de controle
- Desafios

UNIFACISA
CENTRO UNIVERSITÁRIO

02

Introdução a Java

O que é Java?

- Uma linguagem de programação versátil e orientada a objetos, conhecida por sua portabilidade, segurança e capacidade de executar em diferentes plataformas. É amplamente utilizado no desenvolvimento de aplicativos empresariais, móveis e outros aplicativos a fins.

Introdução a Java

Versões do Java?

- http://www.oracle.com/technetwork/java/javase

Edições

- **Java ME** - Java Micro Edition - dispositivos embarcados e móveis - IoT
  - http://www.oracle.com/technetwork/java/javame
- **Java SE** - Java Standard Edition - core - desktop e servidores
  - http://www.oracle.com/technetwork/java/javase
- **Java EE** - Java Enterprise Edition - aplicações corporativas
  - http://www.oracle.com/technetwork/java/javaee

04

Introdução a Java

Visões Importantes

- Código compilado para bytecode e executado em máquina virtual (JVM)
- Segura, robusta
- Roda em vários tipos de dispositivos
- Domina o mercado corporativo desde o fim do século 20
- Padrão Android por muitos anos

Introdução a Java

JVM - Java Virtual Machine

- Máquina virtual do Java (necessário para executar aplicações Java)

O que Máquina Virtual?

É um software que simula uma máquina física e executa programas em um formato intermediário.

06

Introdução a Java

Tipos de Linguagens

Compiladas: C e C++

São linguagens em que o código-fonte é traduzido para linguagem de máquina (código binário) antes da execução. O compilador converte todo o código de uma vez, gerando um arquivo executável que pode ser executado diretamente pelo sistema operacional.

Interpretadas: PHP e JavaScript

São linguagens em que o código-fonte é executado linha por linha por um interpretador em tempo real. Não há uma etapa de compilação que gere um arquivo executável separado.

Pré-Compiladas + Máquina Virtual: Java e C#

São linguagens em que o código-fonte é primeiro compilado para um código intermediário (bytecode) e, em seguida, interpretado por uma máquina virtual específica. Isso oferece portabilidade e eficiência.

07

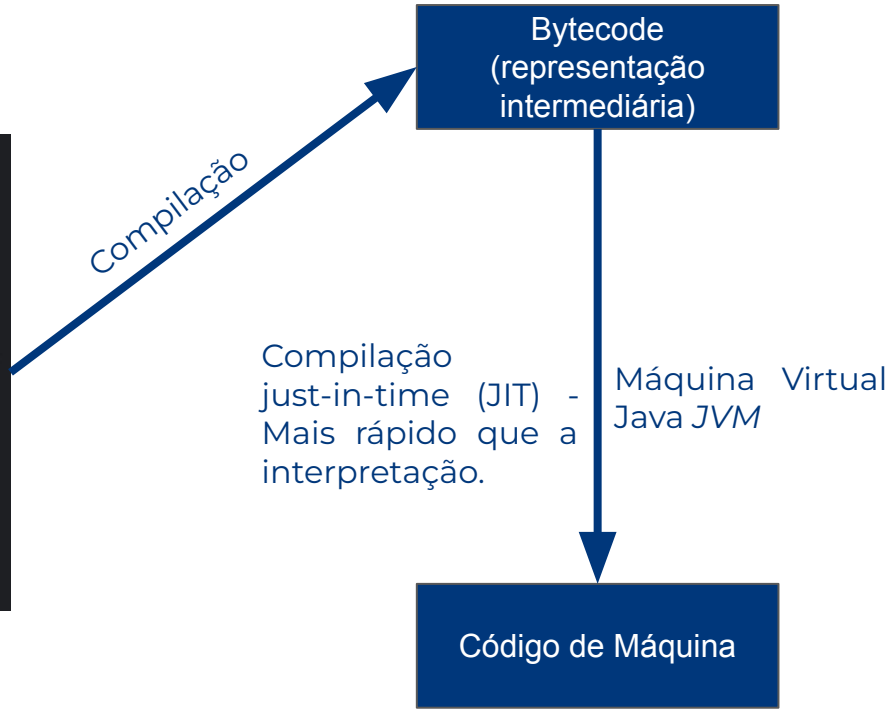
Introdução a Java

JAVA - Linguagem Pré-Compiladas + Máquina Virtual

```java
package org.example;

import java.util.Scanner;

public class Main {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        double x, y, soma;
        System.out.println("Digite o primeiro número: ");
        x = sc.nextInt();
        System.out.println("Digite o segundo número: ");
        y = sc.nextInt();
        soma = x + y;
        System.out.println("Resultado: " + soma);
        sc.close();
    }
}
```

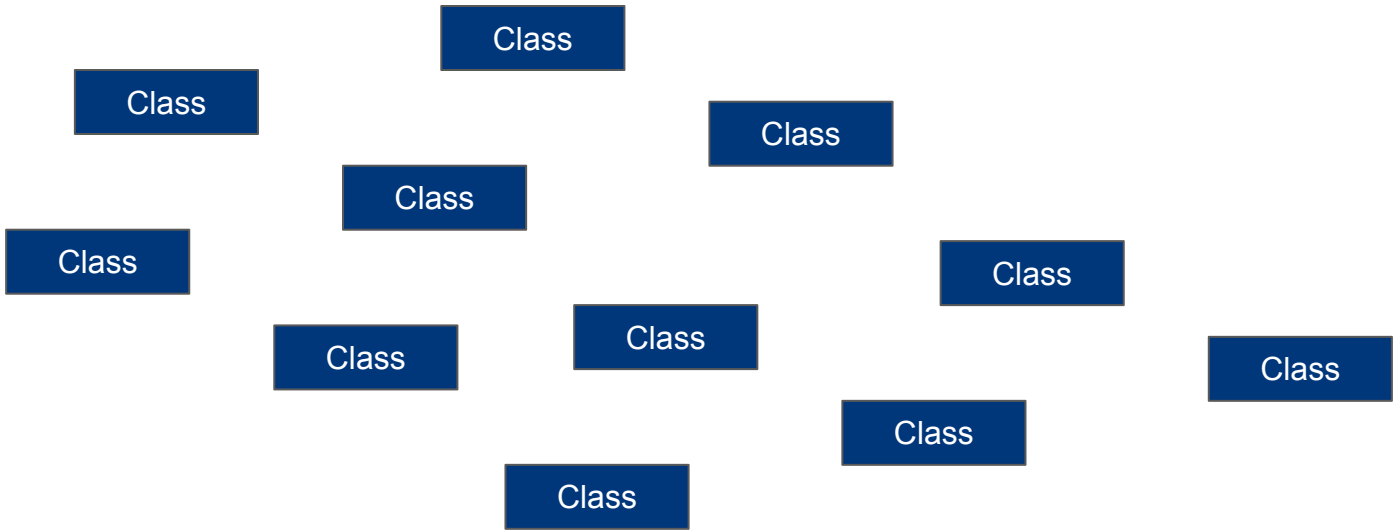


UNIFACISA CENTRO UNIVERSITÁRIO

08

Estrutura de uma Aplicação Java

JAVA é uma linguagem orientada a objetos. Então suas aplicações são compostas por classes

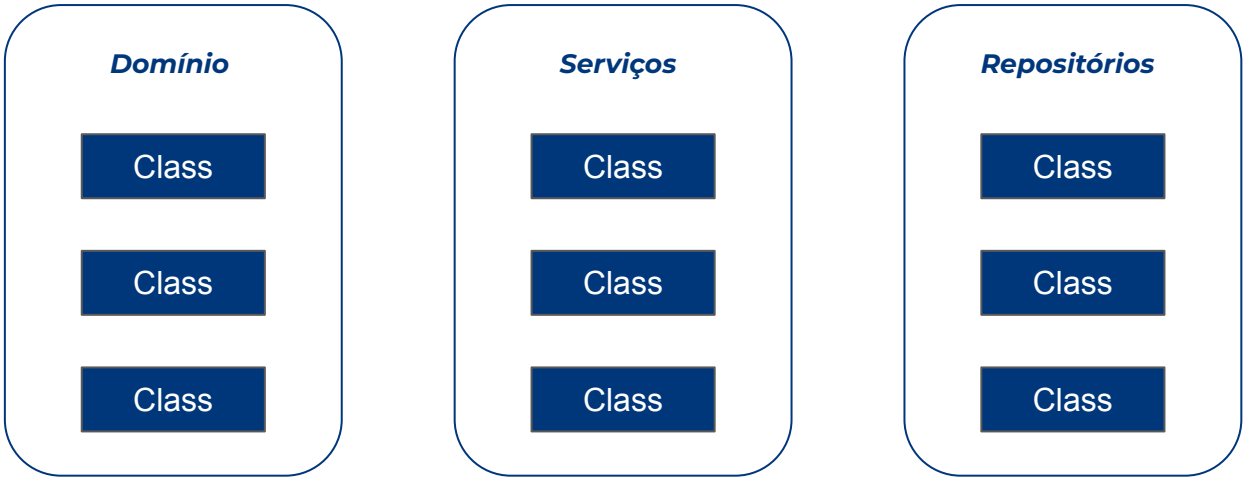


UNIFACISA
CENTRO UNIVERSITÁRIO

09

Estrutura de uma Aplicação Java

Package, nada mais que um agrupamento **lógico** de classes relacionadas.

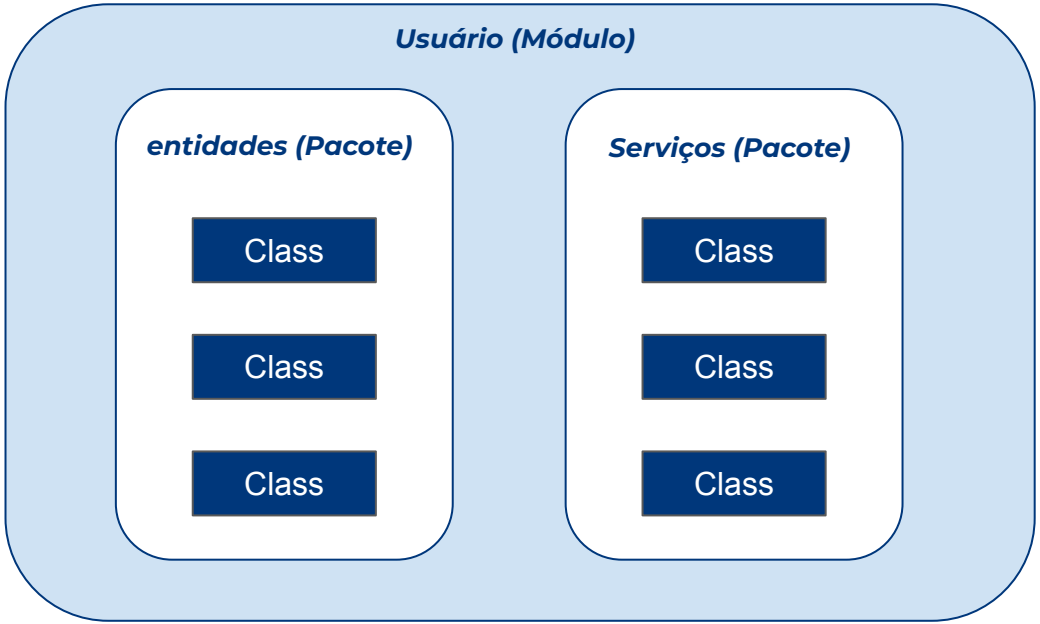


UNIFACISA
CENTRO UNIVERSITÁRIO

09

Estrutura de uma Aplicação Java

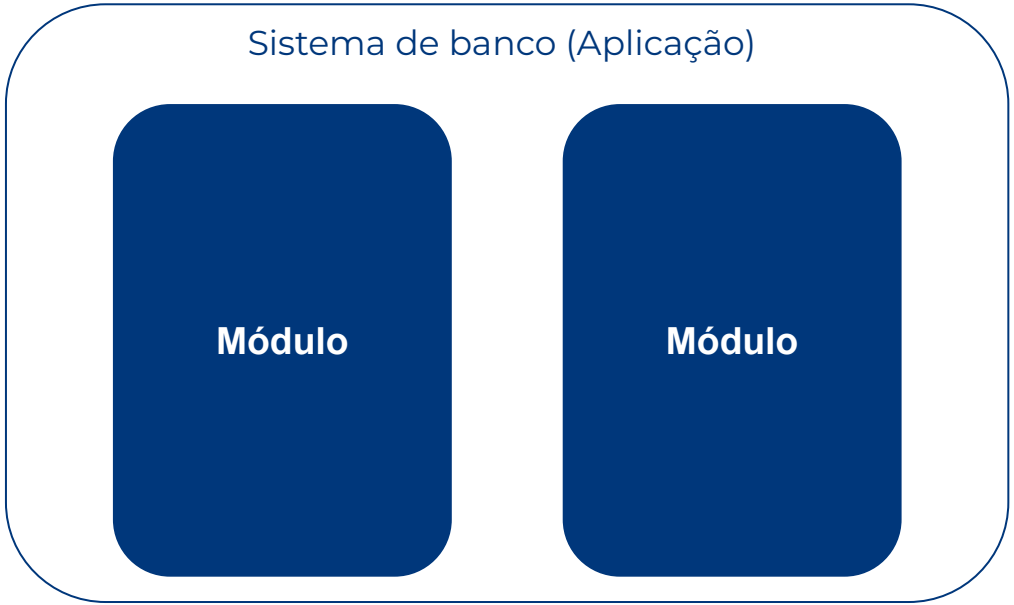
Módulos (Java 9+), nada mais que um agrupamento lógico de pacotes relacionados.



UNIFACISA
CENTRO UNIVERSITÁRIO

Estrutura de uma Aplicação Java

Aplicação, nada mais que um agrupamento de módulos ou pacotes relacionados.



Instalando o Java

Sugestão de links:

Windows:

https://www.youtube.com/watch?v=QekeJBShCy4

Linux:

https://www.youtube.com/watch?v=Sv0EwYPLw8w&list=PLNuUvBZGBA8mcAF-YX7RJhA26TBLdG5yk&index=3

Integrated Development Environment (IDE)

O que é IDE?

IDE (Ambiente de Desenvolvimento Integrado) refere-se a um software que oferece um conjunto abrangente de ferramentas para facilitar o desenvolvimento de software. Uma IDE geralmente inclui um editor de código-fonte, um compilador/intérprete, ferramentas de depuração, um navegador de código, e muitas vezes recursos como controle de versão, gerenciamento de projetos e suporte a plugins.

Sugestão de IDE;

- intellij idea
- eclipse

Primeiro código

```java
package org.example;

import java.util.Scanner;

public class Main {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        double x, y, soma;
        System.out.println("Digite o primeiro número: ");
        x = sc.nextInt();
        System.out.println("Digite o segundo número: ");
        y = sc.nextInt();
        soma = x + y;
        System.out.println("Resultado: " + soma);
        sc.close();
    }
}
```

Nesse código iremos pedir que o usuário nos forneça dois números e iremos imprimir no terminal a soma deles.

Scanner sc = new Scanner(System.in);
sc.nextInt()
Obs.: esse trecho de código que fizemos, faz a mesma coisa que fizemos com o **input()** em python.

System.out.println()
Obs.: esse trecho de código que fizemos, faz a mesma coisa que fizemos com o **print()** em python.

Revisando Estruturas de Controle

Estrutura de condição

- **If e else:** Usada para tomar decisões com base em uma condição. O bloco de código dentro do if é executado se a condição for verdadeira; caso contrário, o bloco dentro do else é executado (se presente).

```
if (condicao) {
    // Código a ser executado se a condição for verdadeira
} else {
    // Código a ser executado se a condição for falsa
}
```

Revisando Estruturas de Controle

Estrutura de condição

- **Switch:** Usada para criar uma estrutura de decisão com múltiplas opções com base no valor de uma expressão. Cada opção é representada por um caso (case).

```
switch (expressao) {
    case valor1:
        // Código para valor1
        break;
    case valor2:
        // Código para valor2
        break;
    // ...
    default:
        // Código padrão
}
```

Revisando Estruturas de Controle

Estrutura de repetição

- **for:** Usada para criar loops com um número específico de iterações.

```
for (int i = 0; i < 5; i++) {
    // Código a ser repetido
}
```

Revisando Estruturas de Controle

Estrutura de repetição

- **while:** Usada para criar loops enquanto uma condição é verdadeira.

```
while (condicao) {
    // Código a ser repetido enquanto a condição for verdadeira
}
```

Revisando Estruturas de Controle

Estrutura de repetição

- **Do-while:** Usada para criar loops onde o bloco de código é executado pelo menos uma vez e, em seguida, repetido enquanto a condição é verdadeira.

```
do {
    // Código a ser executado pelo menos uma vez e repetido enquanto a condição for verdadeira
} while (condicao);
```

Exercício - Gerenciador de Tarefas

Agora vamos aplicar o que aprendemos sobre **if, else, switch, for, while e do while**. Iremos resolver três desafios que são:

1. Um de if e else
2. Um de switch
3. Um de for, while e do while.

Link da atividade: Desafios - if, else, switch, for, while, do while

Obrigado!

Contatos

Prof. Arthur M. Araújo

arthur.araujo@maisunifacisa.com.br

UNIFACISA
CENTRO UNIVERSITÁRIO