

POO

Msc. Édimo Sousa Silva.

Sistemas de Informação
Faculdade Paraíso (FAP)



Aula 02 - Introdução a Java

- **JVM, JRE, JDK**
- **Compilação e interpretação**
- **Configuração de Ambiente**
- **Hello World**

JVM, JRE, JDK

- **JVM (Java virtual Machine)**
 - É a máquina virtual do java, ela é a interface que permite que um programa feito em Java seja executado em qualquer Sistema Operacional.
- **JRE (Java Runtime Environment)**
 - Composta pela (JVM), nas classes centrais e bibliotecas de suporte da plataforma Java.
- **JDK (Java Development kit)**
 - Conjunto de ferramentas necessárias para desenvolver uma aplicação Java, contém itens como compilador javac e o empacotador jar.

Compilação e interpretação

- **Compilação**

- Os arquivos .java são compilados para .class.

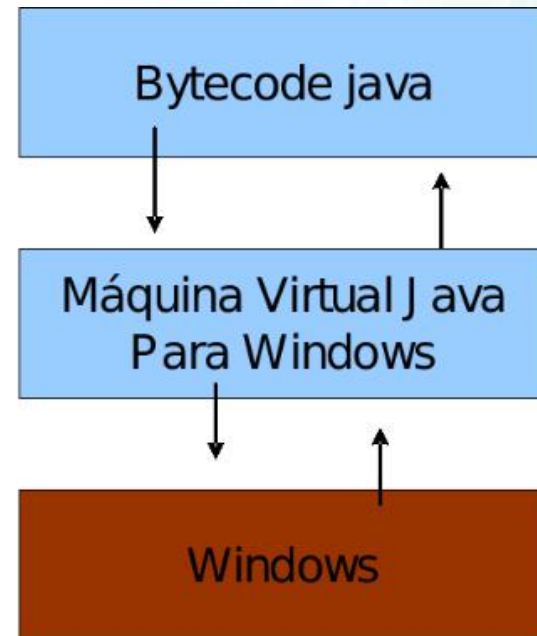
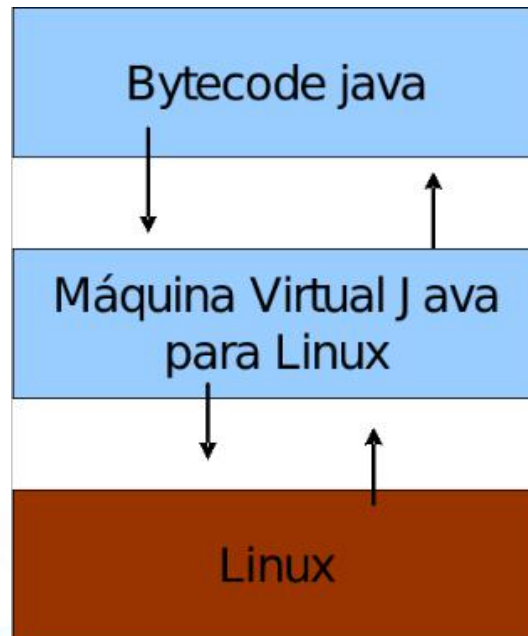
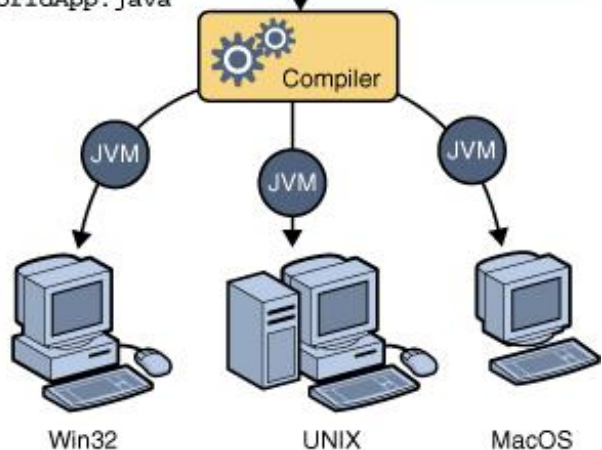
- **Interpretação**

- Os arquivos .class são interpretados pela JVM.

Java Program

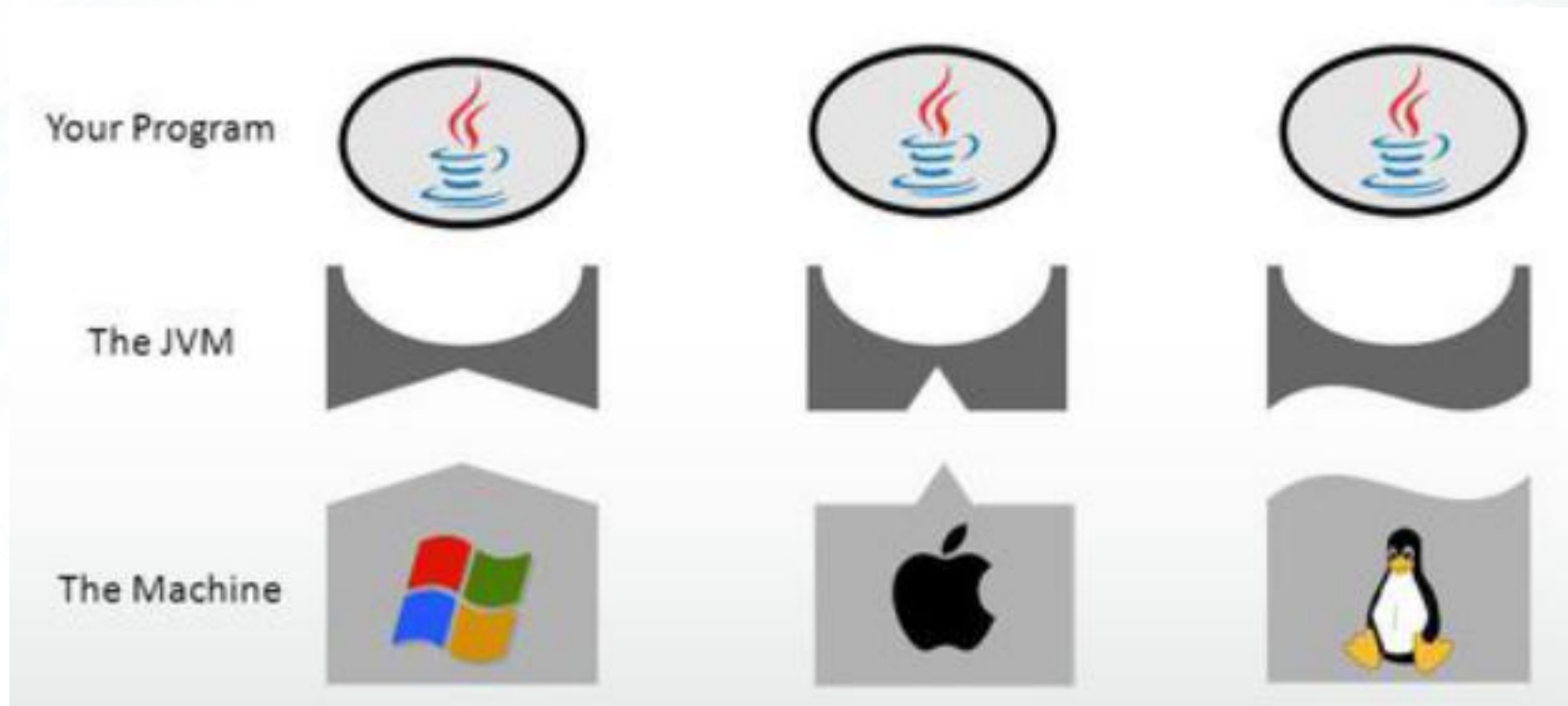
```
class HelloWorldApp {  
    public static void main(String[] args) {  
        System.out.println("Hello World!");  
    }  
}
```

HelloWorldApp.java



Compilação e interpretação

- O código java é compilado para bytecodes que são interpretados pela JVM.



Configuração de ambiente

1. Download JDK
2. Instalar o JDK
3. Configurar variáveis de ambiente
4. Instalar IDE (Eclipse ou Netbeans)

Hello World

- Criar um projeto com o nome "HelloWorld";
- Criar um pacote com o nome "logica";
- Criar uma classe dentro do pacote logica com o nome "Principal";
- Criar o método main;

Principal.java

```
package logica;  
  
public class Principal {  
  
    public static void main(String[] args) {  
        System.out.println("Hello World");  
    }  
  
}
```


Método main

- Uma classe pode possuir vários métodos, mas o método main é o método que é chamado quando a classe é executada.
- É um método estático
- Recebe como parâmetro um vetor de caracteres.
- Possui assinatura "public static void main(String[] args)"

Classe - parte 1

- **Classe:** Define as características e o comportamento de um grupo de objetos, que serão as instâncias dessa classe.
- **Características:** são as características que compõem um objeto. Também conhecidas por atributos, propriedades, variáveis.
- **ex:** Nome, Altura, Peso, idade .
- **Comportamento:** São ações que o objeto pode executar quando solicitado. Também conhecidas por método, função.
- **ex:** correr, pular, falar, calcularIMC.

Classe - part 2

- Atributos:

- São dados que caracterizam objetos da classe.
- São armazenados em variáveis.
- Constituem o estado do objeto.

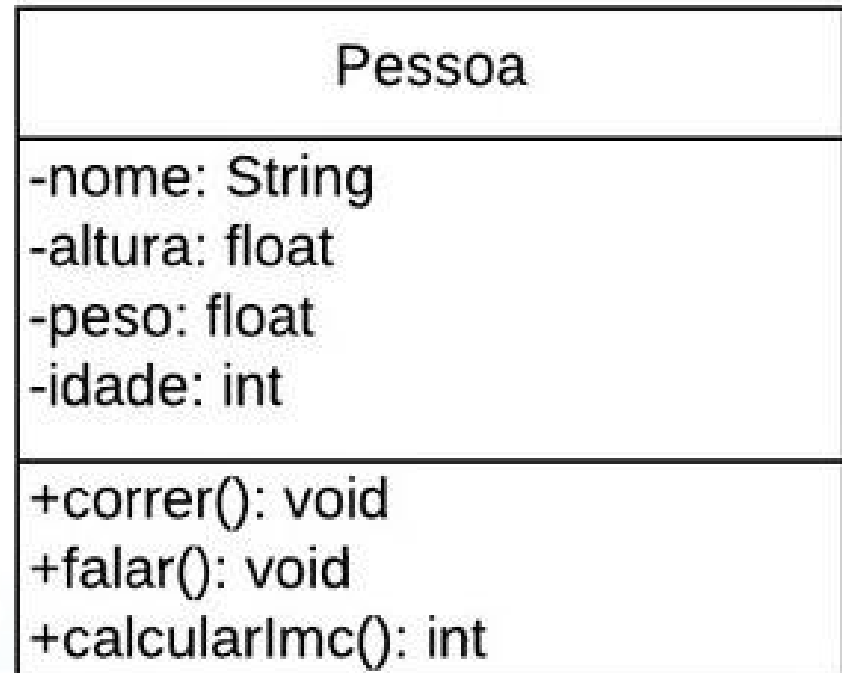
- Operações:

- São os métodos (procedimentos ou funções) que manipulam os dados.

Classe - part 3

```
public class Pessoa {  
    String nome;  
    float altura, peso;  
    int idade;  
  
    void correr () {  
        ...  
    }  
    void falar() {  
        ...  
    }  
    int calcularIMC () {  
        ...  
    }  
}
```

Representação UML



Objeto

- **Objeto:** Uma instância criada a partir de uma classe.

- **Exemplo:**

- A classe Pessoa possui os atributos "nome, altura, peso, idade".
- O objeto pessoa1 possui nome = "Pedro", altura = 1.87 e peso = 91.30.
- O objeto pessoa2 possui nome = "Marta", altura = 1.68 e peso = 60.00.

Objeto - instanciando objetos

- **Dentro da classe principal e do método main:**

```
Pessoa pessoaA = new Pessoa();
```

```
pessoaA.nome = "Pedro";
```

```
pessoaA.idade = 18;
```

```
pessoaA.altura = 1.87f;
```

```
pessoaA.peso = 84f;
```

```
System.out.println("nome = " + pessoaA.nome);
```

```
System.out.println("idade = " + pessoaA.idade);
```

Classe (Exercício)

1) Criar a classe Androide

- **propriedades:** altura, peso, modelo.
- **métodos:** falar, calcular.

Androide
-modelo: String -altura: float -peso: float
+falar(): void +calcular(): void

Bibliografia

- P. Boratti. Programação Orientada a Objetos em Java. 2007.
- P. Deitel, H. Deitel. Java Como Programar. 2010.