Módulo 1

Introdução

Programação Orientada a Objetos I

Java

(Rone Ilídio)

Dados Importantes

- E-mail: rone@ufsj.edu.br
- Página: www.ronepage.com
- Deitel, H. M., Deitel, T. J., “Java”, editora Bookman, 6ª edição – 2003 (ou mais nova)

- Downloads:
- Java: https://www.java.com/pt_BR/download/
- Eclipse: https://eclipse.org/downloads/

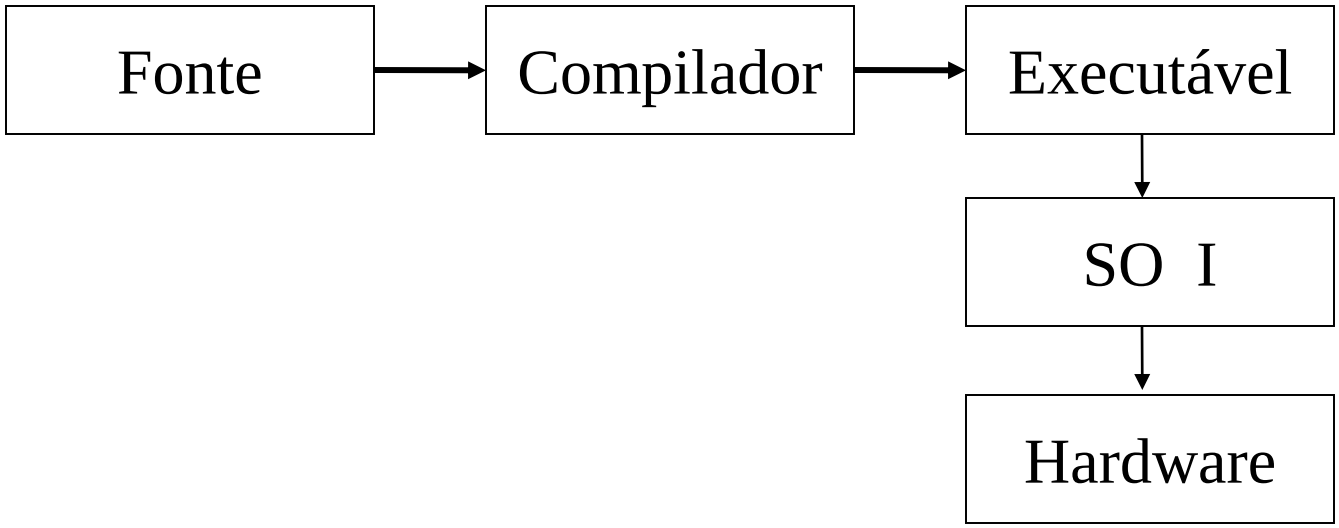
- IMPORTANTE: instale primeiro o Java depois o Eclipse

Java - Vantagens

- Derivado do C e do C++, a mesma sintaxe, por isso programadores desta linguagens têm facilidade para migração.
- FREEWARE : pode ser baixado do site da Sun (www.sun.com)
- Interface gráfica fácil de ser utilizada (GUI)
- Bibliotecas bastante desenvolvidas (API)
- Obs: Java < > Java Script

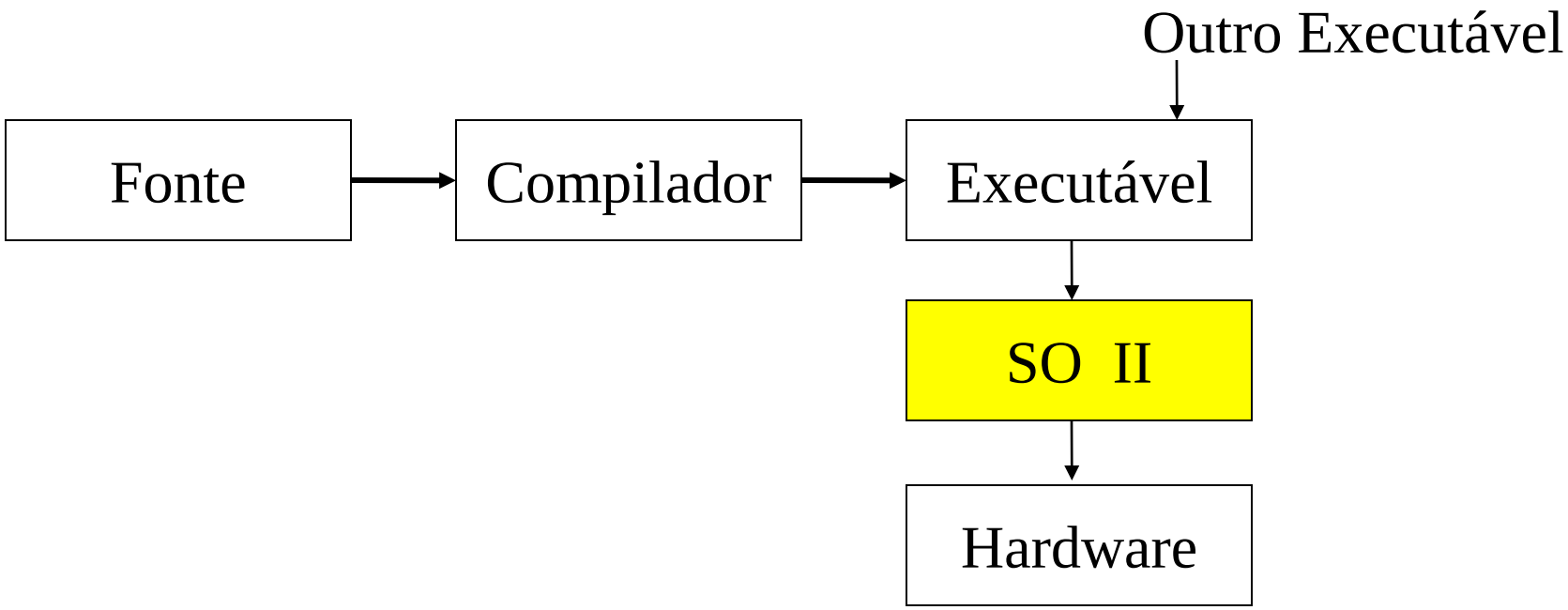
Java - Vantagens

Outras linguagens funcionam da seguite forma:



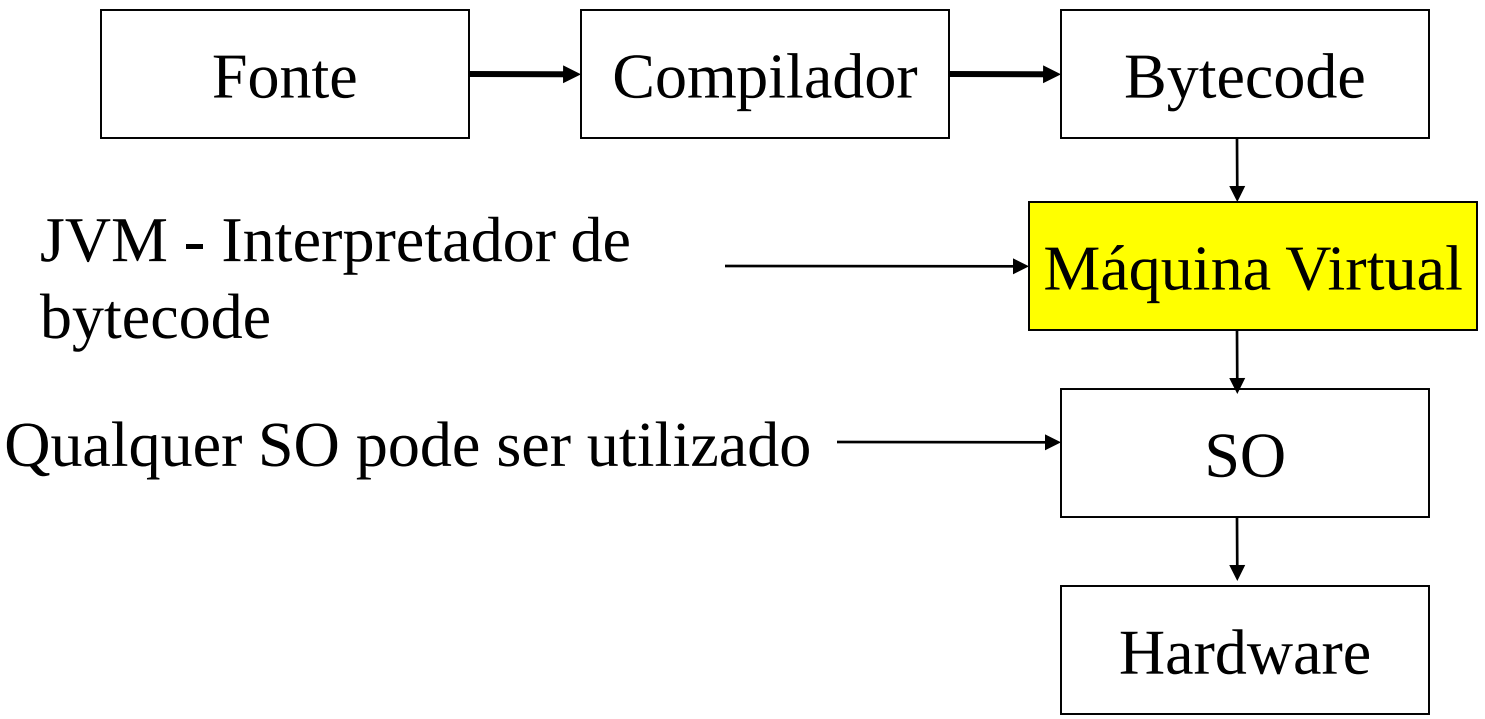
Java - Vantagens

Outras linguagens funcionam da seguite forma:



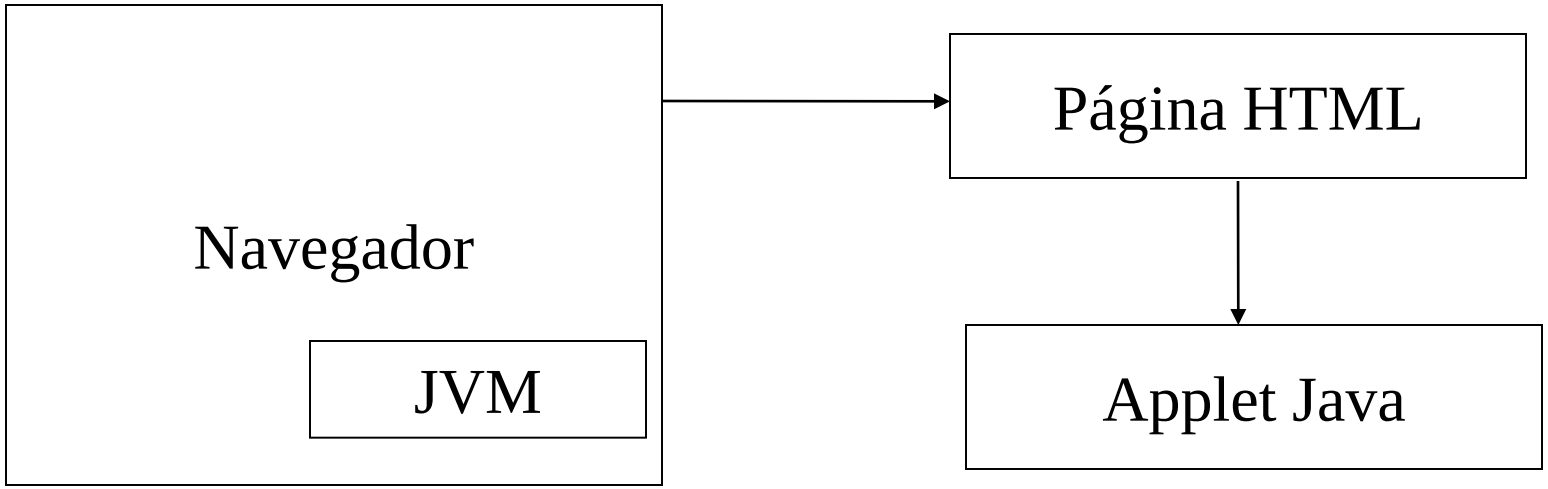
Java - Vantagens

Java é multiplataforma



Java - Vantagens

- É executado por navegadores WEB (*browsers*,como Internet Explorer e Netscape)
- Exemplo: site do Banco do Brasil



Programação Orientada a Objetos

- Java é puramente orientada a objetos pois não há forma de criar um programa, um procedimento ou uma função sem criar pelo menos uma classe.
- Em C++ existe essa possibilidade

Java – Aplicativos e Applets

- Aplicativos são executados localmente como um programa.
- Applets são chamada de dentro de uma página HTML e executados por um navegador WEB.

Java – Aplicativos

Primeiro aplicativo – Hello World!

```
public class Welcome1
{
public static void main (String args [])
{
    System.out.println("Hello Word!");
}
};
```

Java - Aplicativos

- O arquivo de ser salvo com o mesmo nome da classe mais a extensão .java, no caso welcome1.java
- Obs: Java é case sensitive, ou seja, diferencia maiúsculas de minúsculas
- Para executar, pelo prompt do DOS vá até a pasta onde o Programa foi salvo e digite:
  - javac Welcome1.java → para gerar o bytecode

  Obs: este comando gera o arquivo welcome1.class
  - java Welcome1 → para a JVM interpretar welcome1.class

Java - Aplicativos

- **public class Welcome1{}** – declaração da classe
- **public static void main (String args []) {}** – declaração do método principal, primeiro método a ser executado.
- **System.out.println("Hello World!"); -** Comando para imprimir na tela

Java - Aplicativos

Segundo aplicativo – Hello World! Versão 2.0

```
import javax.swing.*;
public class welcome4 {
    public static void main (String args []){
        JOptionPane.showMessageDialog(null,"Hello
                                    World!\nHello World");
    }
}
```

Java - Aplicativos

- **import javax.swing.JOptionPane** – busca na API do Java a classe JOptionPane, onde são implementadas caixas de diálogos, java.lang é o único que não precisa importar.
- **public class welcome4** – declaração da classe welcome4
- **public static void main (String args [])** – método principal, obrigatório senão o aplicativo não é interpretado.
- **JOptionPane.showMessageDialog(null,"Hello World!")** – exibe uma caixa de diálogo com a string “Hello World”. A palavra null é usada para informar o posicionamento padrão (centro da tela).
- **System.exit(0)** – finaliza o aplicativo. Dever ser sempre utilizada em aplicativos que utilizam interface gráfica com o usuário.

Java - Aplicativos

- Terceiro Aplicativo – Addition
- Recebe dois valores, converte para inteiro, soma os dois inteiros e exibe o resultado da soma.
- Este aplicativo trabalha com entrada de dados pelo usuário.

```
import javax.swing.*;
public class Addition{
public static void main( String args [] ){
  String firstnumber, secondnumber;
  int number1, number2, sum;
  firstnumber = JOptionPane.showInputDialog("Entre
                                                com o primeiro número:");
  secondnumber  =
      JOptionPane.showInputDialog("Entre com o segundo número:");
  number1 = Integer.parseInt(firstnumber);
  number2 = Integer.parseInt(secondnumber);
  sum = number1 + number2;
  JOptionPane.showMessageDialog(null,"A soma é : " + sum);

System.exit(0);
}
};
```

Java Aplicativos

- **import javax.swing.JOptionPane** – importa a classe JOptionPane
- public class Addition – declaração da classe Addition
- **public static void main(String args [])** – declaração do método principal
- **String firstnumber, secondnumber** – cria duas variáveis do tipo String
- **int number1, number2, sum** – cria três variáveis do tipo inteiro

Java - Aplicativos

- **firstnumber = JOptionPane.showInputDialog("Entre com o primeiro número:")**
  – A variável recebe a primeira String passada pelo usuário.
- **secondnumber =JOptionPane.showInputDialog("Entre com o segundo número:");**
  – A variável recebe a segunda String passada pelo usuário.
- **number1 = Integer.parseInt(firstnumber);**
- **number2 = Integer.parseInt(secondnumber);**
  – As strings são convertidas em inteiro
- **sum = number1 + number2;** – soma dos valores inteiros

Java Aplicativo

- **JOptionPane.showMessageDialog(null,"A soma é : " +sum, "Resultado", JOptionPane.PLAIN_MESSAGE);**

  - exibe o resultado final.
- **System.exit(0); -** sai do aplicativo

Java - Applets

- É um aplicativo que interpretado por um contêiner, que na maioria das vezes é um navegador WEB, como o Internet Explores ou Netscape.
- Pode ser interpretado pelo aplicativo appletviewer do pacote Java.
- É sempre chamado através de um arquivo HTML

Java - Applets

- **WelcomeLines.java**

```
import javax.swing.*;
public class WelcomeLines extends JApplet{
    public void init(){
        JOptionPane.showMessageDialog(null,”Hello”);
    }
};
```

Java - Applets

- **import java.awt.Graphics; -** pacote que permite a um applet desenhar
- **import javax.swing.JApplet; -** todo applet é filho desta classe. A classe Applet pode ser utilizada, mas não trataremos isso.
- **public class WelcomeLines extends JApplet{** : declaração da classe WelcomeLines, que é filha de Japplet (o conceito de herança será explicado posteriormente)

Java - Applet

- **super.paint(g);** - chama o metódo paint da classe JApplet
- **g.drawLine(15,10,210,10);**
- **g.drawLine(15,30,210,30);** - desenham linhas na tela. Os argumentos são X inicial e Y inicial, X final e Y final.
- g.drawString("Linhas acima e abaixo!",25,25); - desenha strings na tela. Os argumentos são: a string, posição X e posição Y inciais.

Java - Applets

- **WelcomeLines.html**

```
<HTML>
<HEAD>
<TITLE> New Document </TITLE>
</HEAD>
<BODY>
<applet code="WelcomeLines.class"
   width="300" height="45"></applet>
</BODY>
</HTML>
```

Java - Applets

- Para interpretar um applet vá até a pasta onde o arquivo .java foi salvo
- Compile este arquivo:
  - javac WelcomeLines.java
- Agora execute o arquivo WelcomeLines.html:
  - appletviewer WelcomeLines.html
- O appletviewer desconsidera o código html, ou seja, só interpreta applet.

Java - Applets

- A classe Japplet possui três métodos que sempre são chamados pelo contêiner, são eles:
- init: sempre chamado uma vez no início da execução do applet
- start: chamado depois do init e toda vez que o usuário do navegador retornar para a página html em que o código reside
- paint: chamado no início e sempre que a janela do applet for coberta por outra janela

```
import java.awt.Graphics;
import javax.swing.*;
public class SwitchTest extends JApplet{
int choice;
public void init(){
    String input;
    input = JOptionPane.showInputDialog("Digite 1 para desenhar
                10 linhas\n" + "Digite 2 para desenhar 10 retângulos\n" +
                "Digite 3 para desenhar 10 elipses");
    choice = Integer.parseInt(input);
}

// Continua no próximo slide;
```

```
public void paint(Graphics g){
    super.paint(g);
    for (int i=1; i <= 10 ; i++){
        switch(choice){
            case 1:
                g.drawLine(10,10,250,i*10);
                break;
            case 2:
                g.drawRect(10*1,10*i, 50 + 10 * i,50 + 10 * i);
                break;
            case 3:
                g.drawOval(10*1,10*i, 50 + 10 * i,50 + 10 * i);
                break;
            default:
                g.drawString("Comando Inválido!",10, 20 + i * 10);
                break;
        }
    }
}
}
```