

Introdução ao ambiente Java

Programação Orientada a Objetos

Prof. Emerson Ribeiro de Mello

Instituto Federal de Santa Catarina – IFSC
campus São José
mello@ifsc.edu.br

12 de fevereiro de 2014



História da linguagem Java

- Em 1991 um pequeno grupo de engenheiros da Sun Microsystems, chamado de “Green Team”, acreditava que a nova onda computacional seria a união dos dispositivos eletrônicos portáteis com os computadores
 - Nessa época o MS-DOS era o sistema operacional dominante e um telefone celular pesava meio kilo



História da linguagem Java

- Em 1991 um pequeno grupo de engenheiros da Sun Microsystems, chamado de “Green Team”, acreditava que a nova onda computacional seria a união dos dispositivos eletrônicos portáteis com os computadores
 - Nessa época o MS-DOS era o sistema operacional dominante e um telefone celular pesava meio kilo
- Surge a linguagem Java, nome inspirado durante uma reunião do grupo em uma cafeteria
 - O primeiro nome foi Oak (homenagem a árvore que o criador da linguagem via por sua janela), porém descobriu-se mais tarde que já existira uma linguagem com este nome





História da linguagem Java

- O foco inicial do Green Team estava em produtos para entretenimento ligados a indústria de TV digital
 - O conceito estava muito avançado para a época e o projeto começava a ruir



- O foco inicial do Green Team estava em produtos para entretenimento ligados a indústria de TV digital
 - O conceito estava muito avançado para a época e o projeto começava a ruir

A Internet

Em 1993 o projeto toma um novo rumo com o surgimento do primeiro navegador gráfico para web, o Mosaic 1.0 da NCSA. Em 1995 a Sun lança oficialmente o ambiente Java e sua incorporação no Netscape Navigator trouxe vida as páginas web, antes estáticas



O foco inicial: StartSeven - *7



- O Java continua a estar presente em páginas web
 - Em sua forma original (applets) ou em uma forma mais moderna (Java Server Pages - JSP)
- Aplicações para computadores de mesa – IRPF
- Aplicações servidoras – Apache Tomcat
- Dispositivos móveis (telefone celular, GPS, videogame)
- Em sistemas embarcados – Ginga (SBTVD interativa), SmartTVs



- **Orientada a objetos**

- Paradigma que surgiu na década de 60 que tem como foco dados, ou objetos, e suas interfaces
- Recursos de OO do Java são comparáveis aos recursos do C++

- **Robustez**

- Ênfase na verificação preliminar de possíveis problemas, verificação dinâmica (em tempo de execução) e eliminação de situações propensas a erros
- Apresenta solução elegante para os principais pontos fracos do C++
 - Alocação dinâmica de memória e ponteiros



Características da linguagem Java

- **Neutro em relação à arquitetura**

- O compilador Java gera um código intermediário, chamado de *bytecode*, que pode ser executado em qualquer arquitetura de máquina e S.O que tenha um ambiente de execução Java (Máquina virtual Java)

- **Portável**

- Na especificação da linguagem não existem pontos como “dependente de implementação”, como ocorre em C e C++
- Em Java o tipo primitivo `int` sempre consiste de um número inteiro de 32 bits

```
1 #include<stdio.h>
2 #include<limits.h>
3
4 int main(void){
5     printf("%d\n", INT_MAX); /* Qual o valor? Depende da tua
6         arquitetura, 32bits, 64bits...*/
7     return 0;
8 }
```



- **Independente de plataforma**
 - Escreva uma única vez e rode em qualquer lugar que possua uma máquina virtual Java (JVM)
- **Biblioteca completa para concepção de aplicações complexas**
 - Programação concorrente – *Multi-thread*
 - Programação distribuída
- **Alto desempenho**
 - Os *bytecode* são interpretados pela JVM resultando em um desempenho inferior quando comparado com códigos compilados para um CPU específico
 - Os compiladores de *bytecode* “*just-in-time*” surgem como uma solução para este problema, impondo em alguns casos um desempenho superior



- **O Java é interpretado, portanto é muito mais lento**
 - Compiladores *just-in-time* permitem que códigos Java sejam executados com tanta rapidez como códigos C++
 - A inicialização da JVM e as interfaces gráficas em Java (GUI) são sim lentas
- **C# é uma linguagem mais nova, deixando o Java obsoleto**
 - C# incorporou muitas boas ideias do Java, como máquina virtual, linguagem limpa, coleta de lixo
 - Mas deixou para trás a segurança e independência de plataforma (feita para Windows, apesar de haver máquinas virtuais de terceiros para outros S.O, ex: Mono)
- **Javascript é uma versão simplificada do Java**
 - Javascript foi criada pela Netscape para criação de scripts que podem ser usada em páginas Web



- O kit de desenvolvimento Java (*Java Development Kit* – JDK) é distribuído gratuitamente pela Oracle
 - Compilador, máquina virtual Java, código fonte, documentação das APIs
- Para executar uma aplicação Java é necessário possuir uma máquina virtual Java
 - A Oracle disponibiliza gratuitamente o “ambiente de execução Java” (Java Runtime Environment – JRE) para diversos sistemas operacionais e arquiteturas de máquina



Algumas tecnologias Java

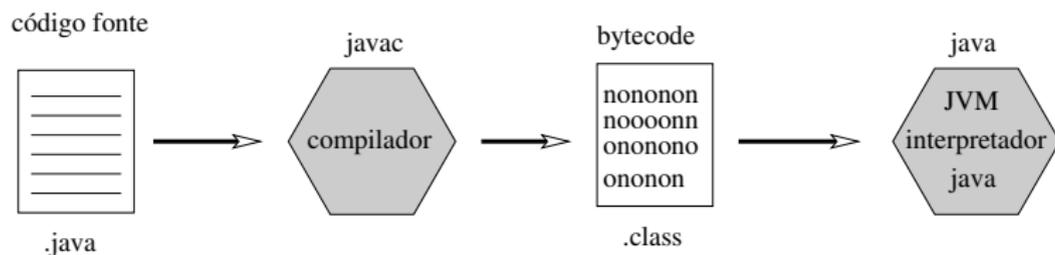
- Java SE – Standard Edition
 - Para o desenvolvimento de aplicações *desktop*
- Java EE – Enterprise Edition
 - Para o desenvolvimento de aplicações empresariais
- Java ME – Mobile Edition
 - Para o desenvolvimento de aplicações para dispositivos móveis
- Java FX
 - Para o desenvolvimento de ricos aplicativos clientes (áudio, vídeo)
 - Aplicações JavaFX podem ser para navegador, *desktop* e dispositivos móveis



- Os ambientes integrados de desenvolvimento (AID) (ou *Integrated Development Environment* – IDE) tornam mais agradável e produtivo o desenvolvimento de aplicações
 - Netbeans – Apoiado pela Oracle
 - <http://www.netbeans.org>
 - Eclipse – Bem usado pela comunidade
 - <http://www.eclipse.org>



Criando e executando um aplicativo Java



- Compilando

8

```
javac Arquivo.java
```

- Executando

9

```
java Arquivo
```



- Um programa em Java consiste em uma coleção de classes
- Geralmente cada classe possui seu respectivo arquivo `.java`
- O nome do arquivo deve ser idêntico ao nome da classe
- O conteúdo do método `main` é a primeira parte de uma classe a ser executada, quando pretende-se que esta seja executada como um aplicativo



Primeiro código em Java – OlaMundo.java

```
10 public class OlaMundo{
11
12
13     public static void main(String[] args){
14
15         // imprimindo a mensagem na tela
16         System.out.println("Ola mundo!");
17
18     }
19
20 }
```

- Compilando e executando

```
21 $ javac OlaMundo.java
22
23 $ java OlaMundo
```



Referências sobre a linguagem

```
24 double d;
25 // algumas funcoes matematicas
26 d = Math.sqrt(25); // obtem a raiz quadrada
27 d = Math.pow(4,2); // 4 elevado a 2
28 d = Math.sin(45); // Math.cos(45), Math.tan(45)...
29
30 // obtendo numeros pseudo-aleatorios de 0 a 9
31 Random r = new Random();
32 int i = r.nextInt(10);
33
34 // formatando a saida
35 // largura de campo de 8 caracteres e precisao de 2 caracteres
36 System.out.printf("%8.2f", d);
37
38 // criando um vetor de inteiros com 10 posicoes
39 int[] vet = new int[10];
40 vet[0] = 5;
41 vet[9] = 4;
```



Referências sobre a linguagem – Lendo informações pelo teclado

```
43 import java.util.Scanner;
44
45 public class Segundo{
46
47     public static void main(String[] args){
48
49         Scanner teclado = new Scanner(System.in);
50
51         int i = teclado.nextInt(); // lendo inteiro
52         double r = teclado.nextDouble(); // lendo real
53         String s = teclado.nextLine(); // lendo cadeia de caracteres
54
55         System.out.println("inteiro: " + i + ", real: " + r);
56         System.out.println("Frase: " + s);
57
58     }
59 }
```



Referências sobre a linguagem – Lendo informações de outra forma

```
60 import javax.swing.JOptionPane;
61
62 public class Terceiro{
63
64     public static void main(String[] args){
65         String texto = JOptionPane.showInputDialog("Entre com um
66             numero");
67         int numero = Integer.parseInt(texto); //convertendo String
68             para int
69         JOptionPane.showMessageDialog(null, numero);
70     }
71 }
```





Java Code Conventions

<http://www.sj.ifsc.edu.br/~mello/livros/java/java-codeconventions.pdf>



Caelum Ensino e Soluções em Java

Apostila Caelum FJ-11 Java e Orientação a Objetos

<http://www.sj.ifsc.edu.br/~mello/livros/java>

- Capítulo 3 – Leitura obrigatória
- Capítulo 2 – Leitura recomendada

