Computação II

Java - Programação Orientada a Objetos

Prof. Danilo S. Carvalho[1, 2]

1. Departamento de Ciência da Computação - UFRJ
2. Centro de Desenvolvimento Tecnológico em Saúde - FIOCRUZ

Índice

1 Porque Java?

2 Estrutura básica de um programa Java

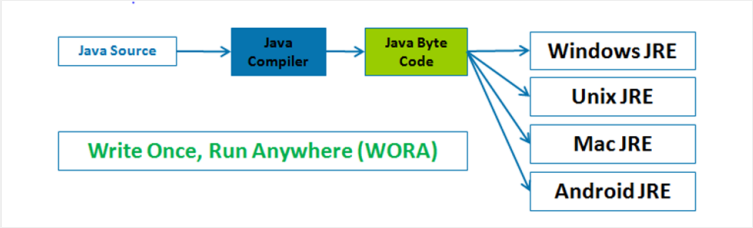
3 Pacotes

O custo do C/C++

- Linguagem de uso profissional mais popular antes do Java.
- Programas escritos em C/C++ precisam ser compilados para cada plataforma computacional onde serão executados.
- Compiladores são programas complexos e requerem grande quantidade de ajustes.
- Atualizações na linguagem implicam na mudança de toda a base instalada de compiladores.

Write once, run anywhere

- Java surgiu com uma proposta ambiciosa.
- Uma vez compilado, o código poderia ser executado em qualquer computador, através de um programa chamado Java Runtime Environment (JRE).
- O JRE, por sua vez, seria sesenvolvido para todas as plataformas de computação.



Linguagens Compiladas e Interpretadas

Linguagens compiladas:

- Produzem código para uma máquina específica (arquit. de computação).
- Código da máquina pode ser otimizado pelo compilador: maior desempenho.
- Permitem validar código (até certo ponto) durante a compilação.

Linguagens interpretadas:

- São executadas a partir de um programa, que é uma aplicação da máquina alvo.
- Não é necessário compilação, mas a interpretação leva mais tempo: menor desempenho.
- Código é validado no momento da execução.

Onde o Java é utilizado?

Como instalar o Java no meu computador?

- Linux: pacotes default-jdk/jre, openjdk-<versão>-jdk/jre
- Windows: pacote oficial Oracle JRE / JDK
- Mac OSX: *adoptopenjdk11* (homebrew), pacote oficial Oracle JRE / JDK

Índice

1 Porque Java?

2 Estrutura básica de um programa Java

3 Pacotes

Programa minimalista

Fonte

```
public class Minimo {

    public static void main(String[] args) {

    }
}
```

Classe principal

- Define o módulo principal do programa.

Fonte

```
public class Minimo { }
```

Método main

- Define o ponto de partida do programa.

Fonte

```
public static void main(String[] args) { }
```

Fonte

```
public class OlaMundo {

    public static void main(String[] args) {
        System.out.println("Olá mundo!");
    }
}
```

Sintaxe: Comentários

Fonte

```
// Comentários de uma linha são escritos assim.

/*
    Para múltiplas linhas,
    escrevemos assim.
*/
```

Sintaxe: Atribuição

Fonte

```
// Atribuição de variável (similar ao C).
int a = 1;
```

Sintaxe: Operadores aritméticos

Fonte

```
int w;
int x = 3;
int y = 7;
float z = x + y;  // - *
w = y / 2;  // % (módulo)
```

Fonte

```
boolean bool = false;
byte b = 0;  // [-128, +127]
char c = '\u0000';  // Unicode 16 bits, de '\u0000' até '\uffff'
short sInt = 0;  // [-32,768, +32,767]
int i = 0;  // [-2^31, +2^31 - 1]
long l = 0L;  // [-2^63, +2^63 - 1]
float f = 0.0f;  // 32-bit IEEE 754 (igual ao C)
double d = 0.0d;  // 64-bit IEEE 754 (igual ao C)
```

Sintaxe: Strings

Fonte

```
String str;
str = "Isso é uma string";
```

Sintaxe: Literais numéricos

Fonte

```
// O número 26, em decimal.
int decVal = 26;
//  O número 26, em hexadecimal.
int hexVal = 0x1a;
// O número 26, em binário.
int binVal = 0b11010;

double d1 = 123.4;
// O mesmo valor que acima, mas em notação científica.
double d2 = 1.234e2;

// Literal float
float f1  = 123.4f;
```

Fonte

```
int[] numeros = new int[30];

// Alocação dinâmica?
int n = 100;
float[] maisNumeros = new float[n];
```

Sintaxe: Método (Função)

Fonte

```
public static int soma(int x, int y) {
    // Tipos primitivos: int, long, float, double, char, boolean...
    return x + y;
}
```

Fonte

```
for(int i = 0; i <= 1000; i++) {
    if(i == 12) {
        continue;  // vai direto para a próxima iteração
    }

    if(i == 15) {
        break;  // sai completamente do loop
    }
}
```

Sintaxe: Loops (2)

Fonte

```
for(int i = 0; i <= 1000; i++) {
    // ... conteúdo do bloco "for" anterior
    // operadores lógicos/numéricos como em C:
    // &&, ||, ==, <, >, <=, >=, != ...
    if(i<5 && i%2 == 0) {
        numeros[i] = i + 100;
    }
    else {
        numeros[i] = i * i;
    }

    // também poderia ter escrito dessa forma:
    numeros[i] = (i<5 && i%2 == 0) ? i+100 : i*i;
    // o operador ternario é idêntico ao C
}
```

Sintaxe: Atribuição de booleano

Fonte

```
boolean test;
test = numeros[1] < numeros[5];  // um booleano "de verdade"
```

Sintaxe: Bloco switch

Fonte

```
int x = numeros[3];

switch(x) {
    case 9:
        System.out.println("Oi!");
        break;

    case 11:
        System.out.println("Tchau!");
        break;

    default:
        System.out.println(test ? "Sei lá!" : "Muito estranho");
}
```

Imprimindo divisores

Fonte

```
public static void imprimirDivisores(int numero) {

    for(int i=1; i <= numero; i++) {
        if(numero % i == 0) {
            System.out.println(i);
        }
    }
}
```

Índice

1 Porque Java?

2 Estrutura básica de um programa Java

3 Pacotes

Pacotes

- Pacotes permitem acessar dados e funcionalidade de módulos externos ao seu código.
- A linguagem Java conta com uma grande quantidade de pacotes em sua biblioteca padrão.
- Podemos obter pacotes de terceiros ou criar os nossos próprios (mais adiante...).

Importando a classe Scanner do pacote java.utils

Fonte

```
import java.util.Scanner;
```

Fonte

```
        Scanner scanner = new Scanner(System.in);  // declarando um scanner
        String linha = scanner.nextLine();  // lendo a próxima linha digita

        System.out.println(String.format("%s possui %d caracteres", linha, l
        /* A linha abaixo terá exatamente o mesmo efeito, usando printf
           (note que não precisamos do String.format() para passar parâmetro
           note também o \n no final). */
        System.out.printf("%s possui %d caracteres\n", linha, linha.length()

        System.out.println("Fim!");
    }
```

Perguntas?