**Introdução à Linguagem Java Através de Exemplos**

* Queremos introduzir a linguagem Java básica (sem uso de orientação a objeto) através de alguns exemplos simples.
* O aluno deve entender, neste momento, que Java é apenas uma outra linguagem de programação, mas possuindo de sintaxe diferente da linguagem vista em Programação I (Python, Pascal, ...)
* Mais tarde, veremos que Java é, na realidade, substancialmente diferente de Python
* Queremos ver como Java trata
  + Tipos básicos
  + Constantes
  + Decisões
  + Laços
  + Compilação e execução de programas
  + IDE - Integrated Development Environment (Eclipse)
* Observe que *não* veremos chamadas de sub-programas (subrotinas ou funções) aqui
  + Este assunto é bem diferente em Java, devido à orientação a objeto, e será visto à frente
* Importante! *Você* é responsável por aprender detalhes adicionais da linguagem Java. O que faremos em aula não é uma cobertura completa da linguagem.

**Por que Java?**

* Linguagem *muito* poderosa embora simples
* Permite portabilidade
* Muito usada em programação em rede
  + Server Programming
  + Aplicações de e-commerce, e-business, etc.
  + Aplicações para acesso via Internet, intranet, etc.

**Um primeiro programa: Hello, World!**

* O programa está no arquivo [Hello.java](http://www.dsc.ufcg.edu.br/~jacques/cursos/p2/html/src/p2/exemplos/Hello.java)

**package p2.exemplos;**

**/\***

**\* O primeiro programa em Java: Hello World**

**\* Autor:**

**\*/**

**// Todo programa tem um ponto de entrada: o "método" main de alguma "classe"**

**public class Hello {**

**public static void main(String[] args) {**

**System.out.println("Hello, world!");**

**}**

**}**

* Observações segue ...
* A linha ...

**package p2.exemplos;**

* ... é usada para indicar que este programa faz parte de um "pacote" que pode conter vários programas
  + É uma forma de organizar vários programas, da mesma forma que "pastas" ou "diretórios" são usados para organizar arquivos num sistema de arquivos
* Dois tipos de comentários (há mais um tipo a ser visto adiante)
* Esqueça, por enquanto, o que significam "public", "classe", "static", "void" e "método"
  + "Método" é semelhante a "module" em Python
  + Outros nomes: sub-rotina, função, procedimento, procedure, ...
* O nome do programa é Hello
  + Por convenção, deve iniciar com uma letra maiúscula
  + Observe que o programa Hello está obrigatoriamente armazenado no arquivo Hello.java
  + Java é "case-sensitive" (reconhece diferença de caixa)
* Até entender detalhes, sempre use as primeiras duas linhas de código do exemplo acima, trocando apenas o nome do programa (Hello)
* "{" significa "BEGIN" e "}" significa "END"
* Observe a forma de imprimir
* Observe a formação de uma constante do tipo string ("Hello, world!")
* Vamos compilar o programa (no Windows, UNIX, etc.)
  + Isso é desnecessário em Python que é interpretado

**javac Hello.java**

* O comando javac é o compilador Java
  + O resultado deve estar no arquivo Hello.class (verifique)
* Agora, vamos executar o programa:

**java Hello**

* O comando java é a "Java Virtual Machine" (JVM) que sabe executar um programa Java compilado com o comando java e presente no arquivo Hello.class
  + Isso é diferente de outras linguagens (C, C++) que são diretamente executáveis após a compilação
  + Motivo: independência de plataforma: Um programa em Java executa em qualquer lugar onde houver uma JVM, sem recompilação
* A saída é:

**Hello, world!**

* Observe que a saída é "a caractere"
  + É possível fazer interfaces gráficas com Java mas não falaremos disso
  + A disciplina de laboratórios tratará do assunto
* É possível usar ambientes integrados de desenvolvimento (IDE) para programar e depurar em Java, o que facilita o desenvolvimento dos projetos.
  + Eclipse
  + NetBeans

**Um segundo programa: entrada de dados, tipos básicos, variáveis, decisões simples**

* Ler 3 números inteiros da entrada, imprimir o menor e o maior.
* A primeira solução está no arquivo [MinMax1.java](http://www.dsc.ufcg.edu.br/~jacques/cursos/p2/html/src/p2/exemplos/MinMax1.java)

**package p2.exemplos;**

**import java.util.Scanner;**

**/\***

**\* "Ler 3 números inteiros da entrada, imprimir o menor e o maior"**

**\***

**\* Autor:**

**\*/**

**public class MinMax1 {**

**public static void main(String[] args) {**

**Scanner sc = new Scanner(System.in);**

**int n1, n2, n3;**

**System.out.print("Entre com o primeiro inteiro: ");**

**n1 = sc.nextInt();**

**System.out.print("Entre com o segundo inteiro: ");**

**n2 = sc.nextInt();**

**System.out.print("Entre com o terceiro inteiro: ");**

**n3 = sc.nextInt();**

**if (n1 > n2) {**

**if (n1 > n3) {**

**if (n2 < n3) {**

**System.out.println("O menor numero eh: " + n2);**

**} else {**

**System.out.println("O menor numero eh: " + n3);**

**}**

**System.out.println("O maior numero eh: " + n1);**

**} else {**

**if (n1 < n2) {**

**System.out.println("O menor numero eh: " + n1);**

**} else {**

**System.out.println("O menor numero eh: " + n2);**

**}**

**System.out.println("O maior numero eh: " + n3);**

**}**

**} else {**

**if (n2 > n3) {**

**if (n1 < n3) {**

**System.out.println("O menor numero eh: " + n1);**

**} else {**

**System.out.println("O menor numero eh: " + n3);**

**}**

**System.out.println("O maior numero eh: " + n2);**

**} else {**

**if (n1 < n2) {**

**System.out.println("O menor numero eh: " + n1);**

**} else {**

**System.out.println("O menor numero eh: " + n2);**

**}**

**System.out.println("O maior numero eh: " + n3);**

**}**

**}**

**}**

**}**

* Como compilar e rodar o programa

**javac MinMax1.java**

**java MinMax1**

* Uma saída típica:

**Entre com o primeiro inteiro: 3**

**Entre com o segundo inteiro: 9**

**Entre com o terceiro inteiro: 123**

**O menor numero eh: 3**

**O maior numero eh: 123**

* A linha ...

**import java.util.Scanner;**

* ... é usada para dizer ao Java que usaremos alguma coisa externa ao nosso programa (a "classe" [Scanner](http://java.sun.com/j2se/1.5.0/docs/api/java/util/Scanner.html))
* A linha ...

**Scanner sc = new Scanner(System.in);**

* ... cria um "Scanner" que é usado para ler dados da entrada
  + Os detalhes (o que significa "new", etc. serão explicados adiante)
* A linha ...

**int n1, n2, n3;**

* ... declara três variáveis inteiras para uso posterior
  + Por convenção, variáveis iniciam com letra minúscula
* A linha ...

**n1 = sc.nextInt();**

* ... lê um inteiro da entrada
* Também poderíamos ter feito assim:

**int n1 = sc.nextInt();**

* As linhas ...

**if(n2 < n3) {**

**System.out.println("O menor numero eh: " + n2);**

**} else {**

**System.out.println("O menor numero eh: " + n3);**

**}**

* ... mostram uma decisão
* Duas dessas linhas também mostram a concatenação de strings com o operador +
  + Falamos que, em Java, o operador "+" está overloaded porque ele significa adição de números e também concatenação de strings, dependendo dos seus operandos
  + É o único operador que sofre overload
* Segue uma segunda versão do programa
  + Esta solução está no arquivo [MinMax2.java](http://www.dsc.ufcg.edu.br/~jacques/cursos/p2/html/src/p2/exemplos/MinMax2.java)

**package p2.exemplos;**

**import java.util.Scanner;**

**/\***

**\* "Ler 3 números inteiros da entrada, imprimir o menor e o maior"**

**\***

**\* Autor:**

**\*/**

**public class MinMax2 {**

**public static void main(String[] args) {**

**Scanner sc = new Scanner(System.in);**

**int n1, n2, n3;**

**System.out.print("Entre com o primeiro inteiro: ");**

**n1 = sc.nextInt();**

**System.out.print("Entre com o segundo inteiro: ");**

**n2 = sc.nextInt();**

**System.out.print("Entre com o terceiro inteiro: ");**

**n3 = sc.nextInt();**

**int mínimo;**

**int máximo;**

**if (n1 > n2) {**

**if (n1 > n3) {**

**if (n2 < n3) {**

**mínimo = n2;**

**} else {**

**mínimo = n3;**

**}**

**máximo = n1;**

**} else {**

**if (n1 < n2) {**

**mínimo = n1;**

**} else {**

**mínimo = n2;**

**}**

**máximo = n3;**

**}**

**} else {**

**if (n2 > n3) {**

**if (n1 < n3) {**

**mínimo = n1;**

**} else {**

**mínimo = n3;**

**}**

**máximo = n2;**

**} else {**

**if (n1 < n2) {**

**mínimo = n1;**

**} else {**

**mínimo = n3;**

**}**

**máximo = n3;**

**}**

**}**

**System.out.println("O menor numero eh: " + mínimo);**

**System.out.println("O maior numero eh: " + máximo);**

**}**

**}**

* Observe que estamos usando acentuação variáveis com acentuação
  + Isso é possível porque Java usa Unicode como código de caracteres
* Perguntas sobre o programa acima
  + Você achou o programa "bem escrito"?
  + É fácil de entender?
  + É fácil trocar as mensagenms de saída por outras?
  + É fácil assegurar-se de que não há bug?
    - Na realidade, um dos programas acima tem um bug: ache o!
  + É fácil estender para 4 números lidos na entrada?
* Que tal o seguinte programa que resolve o mesmo problema:
  + Esta solução está no arquivo [MinMax3.java](http://www.dsc.ufcg.edu.br/~jacques/cursos/p2/html/src/p2/exemplos/MinMax3.java)

**package p2.exemplos;**

**import java.util.Scanner;**

**/\***

**\* "Ler 3 números inteiros da entrada, imprimir o menor e o maior"**

**\***

**\* Autor:**

**\*/**

**public class MinMax3 {**

**public static void main(String[] args) {**

**Scanner sc = new Scanner(System.in);**

**int num;**

**int mínimo = Integer.MAX\_VALUE;**

**int máximo = Integer.MIN\_VALUE;**

**System.out.print("Entre com o primeiro inteiro: ");**

**num = sc.nextInt();**

**if (num < mínimo) {**

**mínimo = num;**

**}**

**if (num > máximo) {**

**máximo = num;**

**}**

**System.out.print("Entre com o segundo inteiro: ");**

**num = sc.nextInt();**

**if (num < mínimo) {**

**mínimo = num;**

**}**

**if (num > máximo) {**

**máximo = num;**

**}**

**System.out.print("Entre com o terceiro inteiro: ");**

**num = sc.nextInt();**

**if (num < mínimo) {**

**mínimo = num;**

**}**

**if (num > máximo) {**

**máximo = num;**

**}**

**System.out.println("O menor numero eh: " + mínimo);**

**System.out.println("O maior numero eh: " + máximo);**

**}**

**}**

* É muito mais limpo, não é?
  + Explique por quê
* Agora, vamos resolver a mesma coisa com um laço simples:
  + Esta solução está no arquivo [MinMax4.java](http://www.dsc.ufcg.edu.br/~jacques/cursos/p2/html/src/p2/exemplos/MinMax4.java)

**package p2.exemplos;**

**import java.util.Scanner;**

**/\***

**\* "Ler 3 números inteiros da entrada, imprimir o menor e o maior"**

**\***

**\* Autor:**

**\*/**

**public class MinMax4 {**

**public static void main(String[] args) {**

**final int NÚMEROS\_A\_LER = 3;**

**Scanner sc = new Scanner(System.in);**

**int mínimo = Integer.MAX\_VALUE;**

**int máximo = Integer.MIN\_VALUE;**

**for (int i = 0; i < NÚMEROS\_A\_LER; i++) {**

**System.out.print("Entre com o proximo inteiro: ");**

**int num = sc.nextInt();**

**if (num < mínimo) {**

**mínimo = num;**

**}**

**if (num > máximo) {**

**máximo = num;**

**}**

**}**

**System.out.println("O menor numero eh: " + mínimo);**

**System.out.println("O maior numero eh: " + máximo);**

**}**

**}**

* NÚMEROS\_A\_LER é declarado como "final" para indicar que é uma constante
  + Como "const" em Pascal ou C
  + Por convenção, usam-se letras maiúsculas para constantes (como em Python, C, C++)
  + Melhor usar constantes simbólicas do que contantes numéricas
    - Programa fica mais simples de entender
* Observe como fazer um laço "for" na linha:

**for(int i = 0; i < NÚMEROS\_A\_LER; i++) {**

* A expressão ...

**i++**

* ... significa ...

**i = i + 1**

* Observe também onde a variável "num" foi declarada:

**int num = sc.nextInt();**

* De forma geral, é bom declarar uma variável perto de onde ela é usada
  + Fizemos a mesma coisa com a declaração da variável do laço (i)

**Um terceiro programa: tipos primitivos, limites de representação, constantes, operadores e expressões**

* Primeiro, um pouco de teoria ...

**Tipos primitivos, limites de representação e constantes**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Tipo primitivo** | **Tamanho** | **Mínimo** | **Máximo** | **Exemplos de Constantes** |
| boolean | 1 bit | - | - | true, false |
| char | 16 bits | Unicode 0 | Unicode 65.535 | 'a' (letra a) |
| byte | 8 bits | -128 | +127 | 97, -23 0x65 (hexadecimal) |
| short | 16 bits | -215 (-32.768) | +215-1(32.767) | 17569, -21875 |
| int | 32 bits | -231 (uns -2 bi) | +231-1 (uns 2 bi) | 1876345240, -2000000000 |
| long | 64 bits | -263  (uns -9 quintilhões) | +263-1 (uns 9 quintilhões) | 123981723971982318273L, -12381726387613678688L, 97L, -23L, 0L (Observe o 'L' final) |
| float | 32 bits | aprox -1038  (6-7 dígitos significativos) | aprox +1038  (6-7 dígitos significativos) | -3.4F 45.78E+23F (Observe o 'F' final) |
| double | 64 bits | aprox -10308  (15 dígitos significativos) | aprox +10308  (15 dígitos significativos) | -3.4 45.78E+23 |
| void | - | - | - | indica ausência de tipo (quando um "método" nada retorna) |

* A conversão entre tipos (quando possível) é feita com cast:

**double x = 8.89;**

**int n = (int)x; // n terá valor 8**

**Operadores**

* Operadores matemáticos
  + + (soma)
  + - (subtração)
  + \* (multiplicação)
  + / (divisão)
  + % (módulo)
  + Há operadores unários - e +
  + Operadores binários podem ser seguido de =, tal como na seguinte expressão:

**soma += nota\*peso; // equivalente a soma = soma + nota\*peso**

* Operador de String
  + + (overload do operador par concatenar strings)
* Operadores de auto-incremento e auto-decremento

**númeroDeAlunos++; // equivalente a númeroDeAlunos = númeroDeAlunos + 1**

**númeroDeAlunos--; // equivalente a númeroDeAlunos = númeroDeAlunos - 1**

**if(númeroDeAlunos-- > 0) // equivalente a testar númeroDeAlunos e depois decrementar**

**if(--númeroDeAlunos > 0) // equivalente a decrementar e depois testar númeroDeAlunos**

* Operadores relacionais
  + < (menor)
  + <= (menor ou igual)
  + > (maior)
  + >= (maior ou igual)
  + == (igual)
  + != (não igual)
* Operadores lógicos
  + && (AND)
  + || (OR)
  + ! (NOT)
  + Exemplos:

**if(númeroDeAlunos > MAX\_ALUNOS || númeroDeProfessores > MAX\_PROFS) ...**

**if(ano % 4 == 0 && ano % 100 != 0 || ano % 400 == 0) ... // ano bissexto**

* Operadores de bits e de deslocamento
  + &, &=, |, |=, ^, ^=, ~
  + Não falaremos deles aqui
* Operador ternário
  + Para escrever uma operação condicional sem usar if-else
  + Exemplo segue

**// a linha seguinte**

**média = númeroDeNotas == 0 ? 0.0 : soma/númeroDeNotas;**

**// é equivalente às linhas seguintes**

**if(númeroDeNotas == 0) {**

**média = 0.0;**

**} else {**

**média = soma/númeroDeNotas;**

**}**

* Expressões complexas e precedência de operadores
  + A precedência dos operadores pode ser vista na tabela abaixo, a qual *não* deve se decorada
  + Observe que a tabela contém operadores que ainda não mencionamos

|  |  |  |
| --- | --- | --- |
| Operadores | Precedência | Associatividade |
| [] . () | Mais alta | dir.gif (503 bytes) |
| ! ~ ++ -- + (unário) - (unário) (cast) new |  | esq.gif (933 bytes) |
| \* / % |  | dir.gif (503 bytes) |
| + - |  | dir.gif (503 bytes) |
| << >> >>> |  | dir.gif (503 bytes) |
| < <= > >= instanceof |  | dir.gif (503 bytes) |
| == != |  | dir.gif (503 bytes) |
| & |  | dir.gif (503 bytes) |
| ^ |  | dir.gif (503 bytes) |
| | |  | dir.gif (503 bytes) |
| && |  | dir.gif (503 bytes) |
| || |  | dir.gif (503 bytes) |
| ?: |  | dir.gif (503 bytes) |
| = += -= \*= /= %= &= |= ^= <<= >>= >>>= | Mais baixa | esq.gif (933 bytes) |

* De forma geral, as precedências "esperadas" funcionam, como mostra o exemplo abaixo

**// a linha seguinte**

**if(númeroDeAlunos > MAX\_ALUNOS || númeroDeProfessores > MAX\_PROFS) ...**

**// não precisa de parênteses, pois é equivalente a**

**if((númeroDeAlunos > MAX\_ALUNOS) || (númeroDeProfessores > MAX\_PROFS)) ...**

* Na dúvida sobre precedências, use parênteses

**Um programa exemplo**

* Segue um programa que ajuda a planejar sua aposentadoria (será que jovens de 18 anos pensam nisso???)
* Esta solução está no arquivo [Aposentadoria.java](http://www.dsc.ufcg.edu.br/~jacques/cursos/p2/html/src/p2/exemplos/Aposentadoria.java)

**package p2.exemplos;**

**import java.util.Scanner;**

**/\***

**\* Planejamento de aposentadoria**

**\***

**\* Autor:**

**\*/**

**public class Aposentadoria {**

**public static void main(String[] args) {**

**Scanner sc = new Scanner(System.in);**

**double alvo;**

**double juros;**

**double contribuição;**

**int anos = 0;**

**double saldo = 0;**

**// exercicio: verifique valores informados abaixo são aceitáveis**

**System.out.print("Quanto dinheiro voce quer para se aposentar? ");**

**alvo = sc.nextDouble();**

**System.out.print("Quanto dinheiro voce vai contribuir todo ano?");**

**contribuição = sc.nextDouble();**

**System.out**

**.print("Taxa de juros em % (exemplo: digite 7.5 para 7.5%): ");**

**juros = sc.nextDouble() / 100;**

**while (saldo < alvo) {**

**saldo = (saldo + contribuição) \* (1 + juros);**

**anos++;**

**}**

**System.out.println("Voce pode se aposentar em " + anos + " anos.");**

**}**

**}**