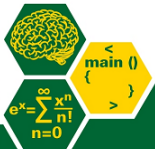




Universidade Federal do ABC

CMCC

Centro de Matemática, Computação e Cognição



Processamento da Informação

Estruturas de seleção em Java

Mirtha Lina Fernández Venero

mirtha.lina@ufabc.edu.br

Sala 506-2, Bloco A

4 de março de 2018



Outline

Introdução

Entrada de dados

Exercícios

Bibliografia



Agenda

Introdução

Entrada de dados

Exercícios

Bibliografia

Estruturas de seleção

Em muitas situações precisamos escolher entre tomar um caminho ou outro



Exemplo: Escrever um programa Java que dado um número inteiro determine se é par ou ímpar

Entrada: Número inteiro

Saída: Mensagem que depende de se o número é par ou ímpar



Agenda

Introdução

Entrada de dados

Exercícios

Bibliografia



Entrada de dados em Java

Para realizar a leitura dum dado de entrada é preciso usar uma variável para armazenar o valor após lido do dispositivo. Em Java existem várias formas, todas pouco intuitivas para um iniciante em programação. A mais simples é a seguinte, e.g.

```
22 // Código para usar a entrada standard (teclado)
23 // Deve ser colocado uma única vez no início
24 // A variável ler será usada para qualquer entrada
25 Scanner ler = new Scanner(System.in);
26
27 // Mensagem para o usuário saber o que deve entrar
28 System.out.print("Digite um número inteiro: ");
29
30 // Leitura dum inteiro
31 int num = ler.nextInt(); // Ler o próximo inteiro do teclado
32
```



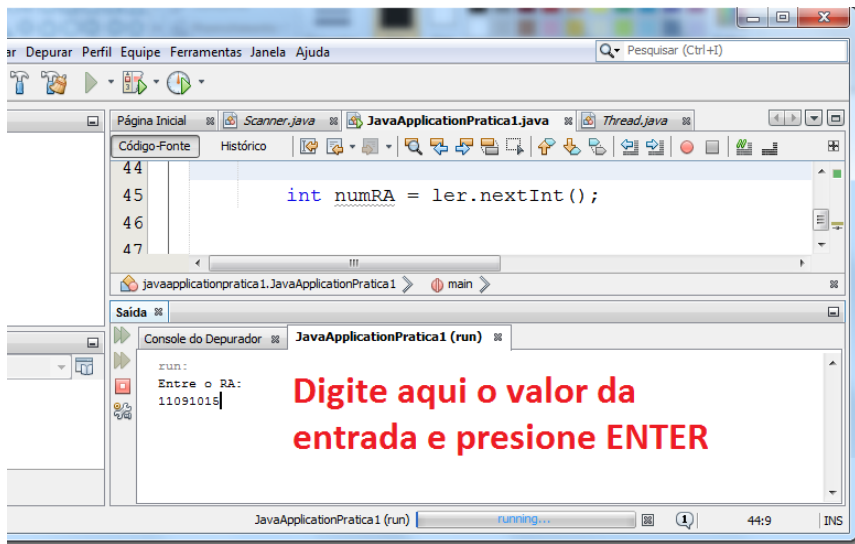
Entrada de dados em Java

Para realizar a leitura dum dado de entrada é preciso usar uma variável para armazenar o valor após lido do dispositivo. Em Java existem várias formas, todas pouco intuitivas para um iniciante em programação. A mais simples é a seguinte, e.g.

```
... ler.  
}
```

<input type="radio"/> next(String pattern)	String
<input type="radio"/> nextBigDecimal()	BigDecimal
<input type="radio"/> nextBigInteger()	BigInteger
<input type="radio"/> nextBigInteger(int radix)	BigInteger
<input type="radio"/> nextBoolean()	boolean
<input type="radio"/> nextByte()	byte
<input type="radio"/> nextByte(int radix)	byte
<input type="radio"/> nextDouble()	double
<input type="radio"/> nextFloat()	float
<input checked="" type="radio"/> nextInt()	int

Entrada de dados durante a execução



Ar Depurar Perfil Equipe Ferramentas Janela Ajuda

Página Inicial Scanner.java JavaApplicationPratica1.java Thread.java

```
44  
45     int numRA = ler.nextInt();  
46  
47
```

Javaapplicationpratica1.JavaApplicationPratica1 main

Saída

Console do Depurador JavaApplicationPratica1 (run)

```
run:  
Entre o RA:  
11091015
```

Digite aqui o valor da entrada e presione ENTER

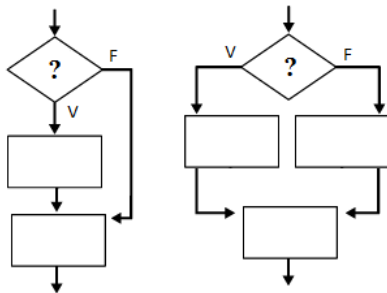
JavaApplicationPratica1 (run) running... 44:9 INS

Estruturas de seleção básicas

Permitem tomar decisões, i.e. executar diferentes passos dependendo de uma pergunta ou condição



Estruturas Condicionais

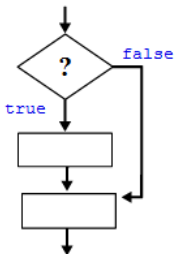




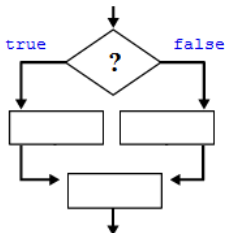
Estruturas de seleção básicas em Java

Permitem tomar decisões, i.e. executar diferentes passos dependendo de uma pergunta ou condição

```
if ( /* boolean exp */ ) {  
    /* inst exp = true */  
}
```



```
if ( /* boolean exp */ b ) {  
    /* inst exp = true */  
}  
else {  
    /* inst exp = false */  
}
```





Expressões booleanas em Java

- ▶ Variáveis, constantes e funções de tipo **boolean**
- ▶ **Expressões relacionais**: Os operadores binários `<`, `<=`, `>`, `>=`, `==` (igualdade), `!=` (desigualdade) são aplicados aos tipos básicos exceto `boolean` e retornam um valor booleano.

Portanto: `x < y < z` (**não!**), `'a' < 'b'` (**ok!**),
`"a" < "b"` (**não!**)

- ▶ **Expressões lógicas**: Os operadores `!` (negação), `&&` (conjunção), `||` (disjunção) operam sobre expressões booleanas e têm menor precedência que os relacionais.

Portanto: `x < y && y < z` (**ok!**), `car == 'a' || 'b'` (**não!**)

São associativos à esquerda e avaliados em curto-circuito.

Portanto: `p || q || r` é avaliado como `((p || q) || r)`



Precedência dos operadores em Java

postfix	<code>expr++ expr--</code>
unary	<code>++expr --expr +expr -expr !</code>
multiplicative	<code>* / %</code>
additive	<code>+ -</code>
relational	<code>< > <= >=</code>
equality	<code>== !=</code>
logical AND	<code>&&</code>
logical OR	<code> </code>
assignment	<code>= += -= *= /= %=</code>



Atenção: Na dúvida use parênteses.



Exemplo: par ou ímpar?

```
public static void main(String[] args) {
    // TODO code application logic here
    Scanner ler = new Scanner(System.in);

    System.out.print("Digite um número inteiro: ");

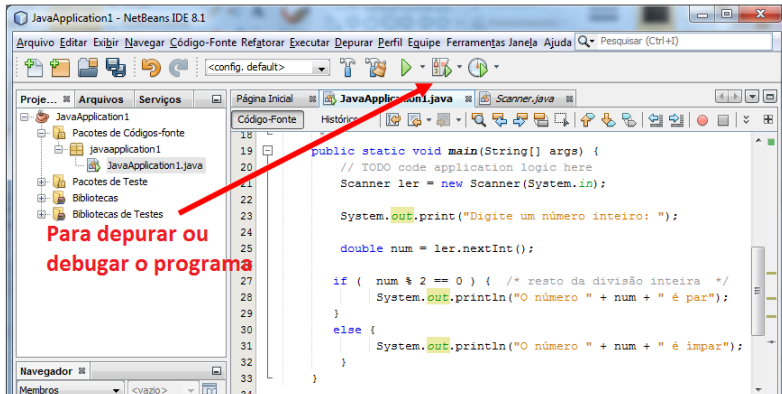
    int num = ler.nextInt();    // Ler o próximo inteiro

    if ( num % 2 == 0 ) { /* resto da divisão inteira */
        System.out.println("O número " + num + " é par");
    }
    else {
        System.out.println("O número " + num + " é ímpar");
    }
}
```

Lembre-te: em Java o operador de igualdade é `==`; o símbolo `=` é usado na instrução de atribuição, nunca numa condição.

Depurando (testando passo a passo) programas em Java

Antes devemos colocar pontos de interrupção nas instruções desejadas clicando no número de linha correspondente

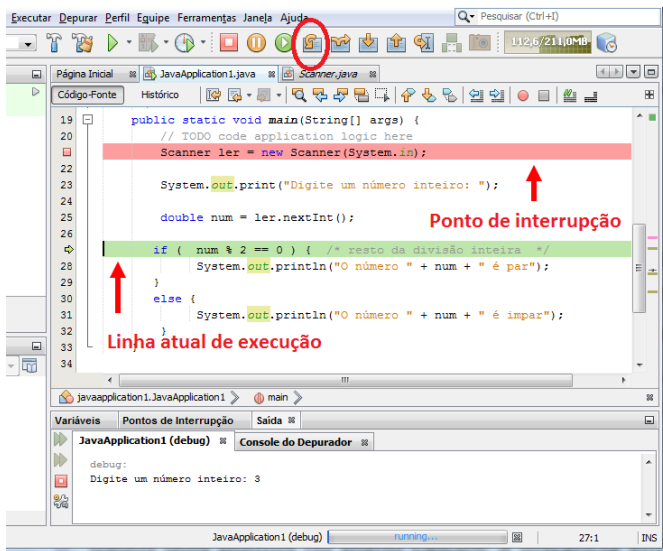


The screenshot shows the NetBeans IDE interface. The main editor displays the source code of `JavaApplication1.java`. A red arrow points to the line number '21' in the left margin, where a breakpoint has been set. The code in the editor is as follows:

```
18  
19 public static void main(String[] args) {  
20     // TODO code application logic here  
21     Scanner ler = new Scanner(System.in);  
22  
23     System.out.print("Digite um número inteiro: ");  
24  
25     double num = ler.nextInt();  
26  
27     if ( num % 2 == 0 ) { /* resto da divisão inteira */  
28         System.out.println("O número " + num + " é par");  
29     }  
30     else {  
31         System.out.println("O número " + num + " é impar");  
32     }  
33 }  
34
```

Below the code editor, there is a red text overlay that reads: **Para depurar ou debugar o programa**. The IDE's toolbar at the top includes icons for running and debugging, with a red arrow pointing to the 'Run with Debugger' icon.

Depurando (testando passo a passo) programas em Java



Executar Depurar Perfil Equipe Ferramentas Janela Ajuda

Pesquisar (Ctrl+F)

112,6/211,0MB

Página Inicial JavaApplication1.java Scanner.java

Código-Fonte Histórico

```
19 public static void main(String[] args) {
20     // TODO code application logic here
21     Scanner ler = new Scanner(System.in);
22
23     System.out.print("Digite um número inteiro: ");
24
25     double num = ler.nextInt();
26
27     if ( num % 2 == 0 ) { /* resto da divisão inteira */
28         System.out.println("O número " + num + " é par");
29     }
30     else {
31         System.out.println("O número " + num + " é impar");
32     }
33 }
34
```

Ponto de interrupção

Linha atual de execução

javaapplication1.JavaApplication1 > main

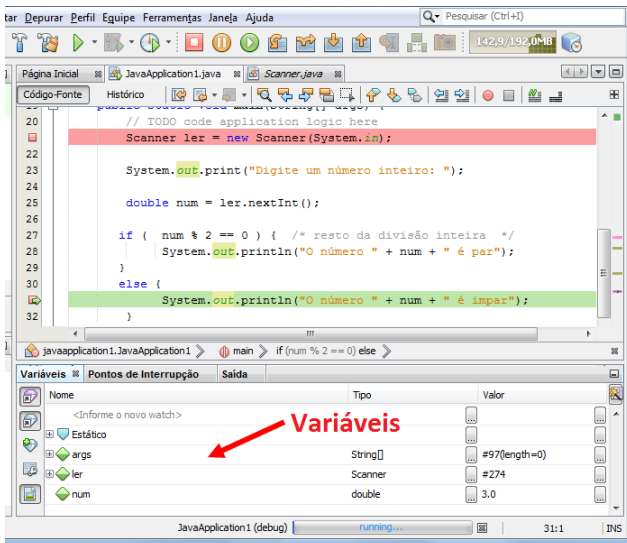
Variáveis Pontos de Interrupção Saída

JavaApplication1 (debug) Console do Depurador

debug:
Digite um número inteiro: 3

JavaApplication1 (debug) running... 27:1 INS

Depurando (testando passo a passo) programas em Java



```
20 // TODO code application logic here
21 Scanner ler = new Scanner(System.in);
22
23 System.out.print("Digite um número inteiro: ");
24
25 double num = ler.nextInt();
26
27 if ( num % 2 == 0 ) { /* resto da divisão inteira */
28     System.out.println("O número " + num + " é par");
29 }
30 else {
31     System.out.println("O número " + num + " é impar");
32 }
```

Variáveis

Nome	Tipo	Valor
<Informe o novo watch>		
Estático		
args	String[]	#97(length=0)
ler	Scanner	#274
num	double	3.0

JavaApplication1 (debug) | running... | 31:1 | INS



Agenda

Introdução

Entrada de dados

Exercícios

Bibliografia



Exercícios em Sala

Escreva um programa Java que leia um valor real x e determine se o resultado da avaliação do polinômio $x^4 - 3x^3 + 4x^2 - 9$ em x é positivo, negativo ou zero.

Lembre-te: Sempre que possível, use variáveis para evitar que seu programa calcule a mesma expressão mais duma vez.



Exercício a submeter no tidia: Atividade1 MassaCorporal

- ▶ Escreva um programa que classifique o estado duma pessoa usando o índice de massa corporal segundo a tabela abaixo

<http://www.calculoimc.com.br/tabela-de-imc/>

$$\text{IMC} = \frac{\text{massa}}{(\text{altura} \cdot \text{altura})}$$

Resultado	Situação
Abaixo de 17	Muito abaixo do peso
Entre 17 e 18,49	Abaixo do peso
Entre 18,5 e 24,99	Peso normal
Entre 25 e 29,99	Acima do peso
Entre 30 e 34,99	Obesidade I
Entre 35 e 39,99	Obesidade II (severa)
Acima de 40	Obesidade III (mórbida)

Data de submissão até 2/3, 14h; Pontos de Atividades: 0.2

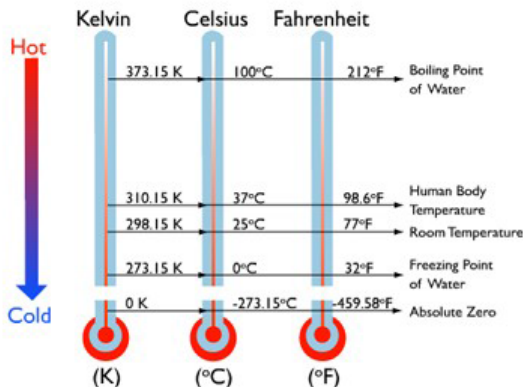


Lembre-te

- ▶ O compilador não vai rodar seu programa com erros. Estude a linguagem e suas estruturas de programação (ver e.g. <http://bcc.ufabc.edu.br/~rogerio.neves/pi/>)
- ▶ **Atenção** com as chaves e parênteses: todo o que é aberto deve ser fechado
- ▶ As expressões em Java não são instruções, **não** coloque ponto e vírgula após a expressão do `if(<expr>)`
- ▶ Use o operador `+` (**não a vírgula**) para juntar várias saídas numa mesma instrução `System.out.print` ou `println`
- ▶ Antes de resolver um exercício leia com atenção o enunciado, pense bem qual deve ser a entrada e a saída do programa, defina variáveis para armazenar essas entradas com nomes e tipos apropriados, pense bem nos passos necessários para o processamento, escreva o programa Java e faça vários testes

Exercícios para estudar em casa

- ▶ Escreva um programa para converter uma temperatura em graus Fahrenheit para graus Celsius e dizer se é quente, fria ou morna (17°-27°). Use a fórmula $C = (F - 32) \times \frac{5}{9}$





Exercícios

- ▶ Escreva um programa que leia três inteiros i , j e k e determine se i é múltiplo somente de j ou somente de k ou de ambos.
- ▶ Dados dois pontos no plano determine qual está mais afastado do origem. Imprima mensagem apropriada se ambos estão à mesma distância (www.mathsisfun.com/algebra/distance-2-points.html)
- ▶ Escreva um programa que classifique um triângulo dependendo dos lados, i.e. equilátero, escaleno ou isósceles
- ▶ Dados os conceitos de um estudante na teoria e prática de PI determine o conceito final e se deve fazer REC ou não



Agenda

Introdução

Entrada de dados

Exercícios

Bibliografia



Bibliografia

Toda a bibliografia recomendada na teoria

Notas de curso: disponibilizadas dia após cada aula em <https://tidia4.ufabc.edu.br/portal>, site **PI-Q1-2017-Mirtha**

Ambientes Integrados de Desenvolvimento para Java

- ▶ NetBeans (<http://www.oracle.com/technetwork/java/javase/downloads/jdk-netbeans-jsp-142931.html>)
Ver video de instalação
<https://www.youtube.com/watch?v=3qn0KAlJgtI>
- ▶ Eclipse (<http://www.eclipse.org/>)
- ▶ Online editors: <https://repl.it/>,
<http://pythontutor.com/java.html#mode=edit>,
http://www.tutorialspoint.com/compile_java_online.php,
<https://www.compilejava.net/>, ...