

ECM251 – Linguagens de Programação I

Aula 04 – L1/1 e L2/1

Engenharia da Computação – 3ª série

Desvios Condicionais e Switch-case ***(L1/1 – L2/1)***

2024

Horário

Terça-feira: 2 x 2 aulas/semana

- L1/1 (07h40min-09h20min): *Prof. Calvetti*;
- L1/2 (09h30min-11h10min): *Prof. Calvetti*;
- L2/1 (07h40min-09h20min): *Prof. Igor Silveira*;
- L2/2 (11h20min-13h00min): *Prof. Calvetti*.

Desvios Condicionais e Switch-case

Tópico

- O desvio condicional *if*

Desvios Condicionais e Switch-case

O desvio condicional *if*

- Serve para mudar o fluxo de execução de um algoritmo baseado em uma ou mais condições;
- Considere o seguinte exemplo:
 - Se um número for maior que 0, imprimir que ele é positivo. Em Java, escreve-se:

```
if(x > 0)
{
    System.out.println("O número é positivo");
}
```

Desvios Condicionais e Switch-case

O desvio condicional *if*

- Para o caso dele não ser positivo, o número pode ser, ainda, negativo ou nulo, então:

```
if(x > 0)
{
    System.out.println("O número é positivo");
}
else
{
    System.out.println("O número é negativo ou nulo");
}
```

Desvios Condicionais e Switch-case

Tópico

- O desvio condicional aninhado com *if – else*

Desvios Condicionais e Switch-case

O desvio condicional aninhado com *if – else*

- Se x não for positivo, ou seja, for menor que 0, ou nulo, necessita-se de mais um desvio, utilizando os desvios aninhados:

```
if(x > 0)
{
    System.out.println("O número é positivo");
}
else
{
    if(x < 0)
    {
        System.out.println("O número é negativo");
    }
    else
    {
        System.out.println("O número é nulo");
    }
}
```

Desvios Condicionais e Switch-case

Tópico

- O desvio condicional *if – else if – else*

Desvios Condicionais e Switch-case

O desvio condicional *if – else if – else*

- Se x não for positivo, ou seja, for menor que 0, ou nulo, necessita-se de mais um desvio, não utilizando os desvios aninhados:

```
if(x > 0)
{ System.out.println("O número é positivo");
}
else if(x < 0)
{ System.out.println("O número é negativo");
}
else
{ System.out.println("O número é nulo");
}
```

ECM251 – Linguagens de Programação I

Desvios Condicionais e Switch-case

Tópico

- O desvio condicional múltiplo *if*

Desvios Condicionais e Switch-case

O desvio condicional múltiplo *if*

- Se *x* não for positivo, ou seja, for menor que 0, ou nulo, necessita-se de mais um desvio, utilizando os desvios condicionais múltiplos:

```
if(x > 0)
{ System.out.println("O número é positivo");
}
if(x < 0)
{ System.out.println("O número é negativo");
}
if(x == 0)
{ System.out.println("O número é nulo");
}
```

Tópico

- As Expressões Lógicas e Operadores Relacionais

Desvios Condicionais e Switch-case

As Expressões Lógicas e Operadores Relacionais

- O desvio é sempre baseado em uma expressão lógica, retornando somente **true** ou **false** (verdadeiro ou falso);
- Para isso tem-se os operadores relacionais, que são:

```
> maior que  
>= maior ou igual à  
< menor que  
<= menor ou igual à  
== igual à  
!= diferente de
```

As Expressões Lógicas e Operadores Relacionais

- Pode-se combinar expressões lógicas usando os operadores lógicos:

&& E

|| OU

! NÃO

- Quando usado o operador &&, as duas expressões lógicas devem ser verdadeiras para o resultado ser verdadeiro;
- Quando usado o operador ||, as duas expressões lógicas devem ser falsas para o resultado ser falso;
- Quando usado o operador !, inverte-se o resultado lógico da expressão, tornando o verdadeiro falso e vice-versa.

Desvios Condicionais e Switch-case

As Expressões Lógicas e Operadores Relacionais

- Exemplo 1: Dados 3 números inteiros, diferentes, a, b, e c, quer-se saber qual deles é o maior, utilizando-se operadores lógicos:

```
if( a > b && a > c)
{   System.out.println("O maior é o a");
}
else if( b > a && b > c)
{   System.out.println("O maior é o b");
}
else
{   System.out.println("O maior é o c");
}
```

As Expressões Lógicas e Operadores Relacionais

- Exemplo 2: Dados 3 números inteiros, a, b, e c, quer-se saber se, pelo menos, há dois iguais no grupo, utilizando-se operadores lógicos:

```
if( a == b || a == c || b == c)
{
    System.out.println("Há pelo menos 2 números iguais");
}
else
{
    System.out.println("Os números são diferentes");
}
```


As Expressões Lógicas e Operadores Relacionais

- Exemplo 3: Considerando-se o Exemplo 2, porém, utilizando-se o operador de negação e os operadores lógicos:

```
if( !(a == b || a == c || b == c))
{
    System.out.println("Os números são diferentes");
}
else
{
    System.out.println("Há pelo menos 2 números iguais");
}
```

Desvios Condicionais e Switch-case

Tópico

- Comparação de ***Strings***

Desvios Condicionais e Switch-case

Comparação de **Strings**

- Os operadores relacionais **==**, **!=**, **>**, **<**, **>=** e **<=**, só podem ser utilizados com variáveis nativas Java. Portanto, não utilize os relacionais para comparar **Strings**, nem outros **objetos**;
- Para isso tem-se o método **equals()**. Ex.:
 - "12345".equals("12345") retorna true
 - "12345".equals("123") retorna false
 - !"12345".equals("123") retorna true
- Para saber-se o tamanho de uma **String** e se uma é mais longa que a outra, pode utilizar-se:
 - "12345".length() > "678".length() retorna true

Desvios Condicionais e Switch-case

Comparação de *Strings*

- Exemplo: Dadas 2 *Strings*, para saber se são iguais, tem-se:

```
String s1 = "aluno";
String s2 = "aluno";

if( s1.equals("aluno"))
{
    System.out.println("São iguais");
}
else
{
    System.out.println("São diferentes");
}
```

Desvios Condicionais e Switch-case

Comparação de **Strings**

- Os operadores relacionais **==**, **!=**, **>**, **<**, **>=** e **<=**, só podem ser utilizados com variáveis nativas Java. Portanto, não utilize os relacionais para comparar **Strings**, nem outros **objetos**;
- Para isso tem-se o método **equals()**. Ex.:
 - "12345".equals("12345") retorna true
 - "12345".equals("123") retorna false
 - !"12345".equals("123") retorna true
- Para saber-se o tamanho de uma **String** e se uma é mais longa que a outra, pode utilizar-se:
 - "12345".length() > "678".length() retorna true

Desvios Condicionais e Switch-case

Tópico

- ***O switch-case***

Desvios Condicionais e Switch-case

O switch-case

- Escolhe uma opção dentre várias (**case**) a partir do valor de uma única variável (**switch**). Algumas regras de funcionamento:
 1. O primeiro case que for verdadeiro é o ponto de entrada, mais a direita; a partir daí todos os comandos à direita serão executados até que o **switch-case** termine ou um **break** seja encontrado;
 2. Somente variáveis primitivas Java são aceitas no **switch**; o tipo **String** passou a ser aceito a partir do **Java 7**;
 3. Se houver mais do que um caso que responda ao mesmo valor da variável, estes podem ser incluídos na mesma linha:

Ex. **case 1: case 2: case 3:**

Desvios Condicionais e Switch-case

O switch-case

- A cláusula **default** é executada quando nenhum **case** for aceito, sendo essa cláusula não é obrigatória;
- O **break** na última linha (**default**, no exemplo a seguir) é desnecessário, mas recomendado pois torna futuras manutenções no código menos sujeitas a erro.

ECM251 – Linguagens de Programação I

Desvios Condicionais e Switch-case

Execução de um *case* por vez

```
public class SwitchDemo {  
    public static void main(String[] args) {  
  
        int iMes = 8;  
        String sMes;  
        switch (iMes) {  
            case 1: sMes = "Janeiro";  
                break;  
            case 2: sMes = "Fevereiro";  
                break;  
            case 3: sMes = "Março";  
                break;  
            case 4: sMes = "Abril";  
                break;  
            case 5: sMes = "Maio";  
                break;  
            case 6: sMes = "Junho";  
                break;  
            case 7: sMes = "Julho";  
                break;
```

```
            case 8: sMes = "Agosto";  
                break;  
            case 9: sMes = "Setembro";  
                break;  
            case 10: sMes = "Outubro";  
                break;  
            case 11: sMes = "Novembro";  
                break;  
            case 12: sMes = "Dezembro";  
                break;  
            default: sMes = "Mês Inválido";  
                break;  
        }  
        System.out.println(sMes);  
    }  
}
```

Resultado: Agosto

ECM251 – Linguagens de Programação I

Desvios Condicionais e Switch-case

Execução de vários *cases*

```
public class SwitchDemoFallThrough {  
  
    public static void main(String[] args) {  
        java.util.ArrayList<String> mesesFuturos =  
            new java.util.ArrayList<String>();  
  
        int mes = 8;  
  
        switch (mes) {  
            case 1: mesesFuturos.add("Janeiro");  
            case 2: mesesFuturos.add("Fevereiro");  
            case 3: mesesFuturos.add("Março");  
            case 4: mesesFuturos.add("Abril");  
            case 5: mesesFuturos.add("Maio");  
            case 6: mesesFuturos.add("Junho");  
            case 7: mesesFuturos.add("Julho");  
            case 8: mesesFuturos.add("Agosto");  
            case 9: mesesFuturos.add("Setembro");  
            case 10: mesesFuturos.add("Outubro");  
            case 11: mesesFuturos.add("Novembro");
```

```
            case 12: mesesFuturos.add("Dezembro");  
                break;  
            default:  
                break;  
        }  
  
        if (mesesFuturos.isEmpty()) {  
            System.out.println(  
                "Número do mês inválido.");  
        }  
        else {  
            for (String mesName : mesesFuturos) {  
                System.out.println(mesName);  
            }  
        }  
    }  
}
```

Resultado:
Agosto
Setembro
Outubro
Novembro
Dezembro

ECM251 – Linguagens de Programação I

Desvios Condicionais e Switch-case

Múltiplos *cases* por linha

```
class SwitchDemo2 {  
    public static void main(String[] args) {  
  
        int mes = 2;  
        int ano = 2000;  
        int dias = 0;  
  
        switch (mes) {  
            case 1: case 3: case 5:  
            case 7: case 8: case 10:  
            case 12:  
                dias = 31;  
                break;  
            case 4: case 6:  
            case 9: case 11:  
                dias = 30;  
                break;  

```

```
        case 2:  
            if (((ano % 4 == 0) &&  
                !(ano % 100 == 0))  
                || (ano % 400 == 0))  
                dias = 29;  
            else  
                dias = 28;  
            break;  
        default:  
            System.out.println("Mês inválido.");  
            break;  
        }  
        System.out.println("Número de Dias = "  
                            + dias);  
    }  
}
```

Resultado:
Número de dias = 29

ECM251 – Linguagens de Programação I

Desvios Condicionais e Switch-case

Switch com Strings

```
public class StringSwitchDemo {  
    public static void main(String[] args) {  
        int iMes = 0;  
        String sMes = "Agosto";  
  
        switch (sMes.toLowerCase()) {  
            case "janeiro": iMes = 1;  
                break;  
            case "fevereiro": iMes = 2;  
                break;  
            case "março": iMes = 3;  
                break;  
            case "abril": iMes = 4;  
                break;  
            case "maio": iMes = 5;  
                break;  
            case "junho": iMes = 6;  
                break;  
            case "julho": iMes = 7;  
                break;  
        }  
    }  
}
```

```
        case "agosto": iMes = 8;  
            break;  
        case "setembro": iMes = 9;  
            break;  
        case "outubro": iMes = 10;  
            break;  
        case "novembro": iMes = 11;  
            break;  
        case "dezembro": iMes = 12;  
            break;  
        default: iMes = 0;  
            System.out.println("Mês inválido");  
            break;  
    }  
    System.out.println(iMes);  
}
```

Resultado: 8

Desvios Condicionais e Switch-case

Tópico

- Operador Ternário

Desvios Condicionais e Switch-case

Operador Ternário

Sintaxe:

(condição lógica ? retorno se verdadeiro : retorno se falso);

```
String retorno = (sexo == 'M' ? "masculino" : "feminino");
```

```
String par = (x%2 == 0 ? "par" : "ímpar");
```

```
int absoluto = (x >= 0 ? x : -x);
```

```
boolean nao = (true ? false : true);
```


ECM251 – Linguagens de Programação I

Desvios Condicionais e Switch-case

Exemplo

```
import javax.swing.JOptionPane;
public class ExemploTernario{
    public static void main(String[] args){
        String nome = JOptionPane.showInputDialog("Digite o nome:");
        int idade = Integer.parseInt(JOptionPane.showInputDialog(
            "Digite a idade:"));
        char sexo = JOptionPane.showInputDialog("Digite o sexo:").charAt(0);

        String saida = nome+", "+(idade >= 18?"maior de idade":
            "menor de idade")+", sexo "+(sexo == 'M'?"masculino":"feminino");

        JOptionPane.showMessageDialog(null, saida);
    }
}
```

Exercícios

1. Escrever, em Java, utilizando ***switch-case***, a classe ***ValorEmTexto***, cujos atributos únicos são:
 - a) ***valor***, que é uma variável ***double***, que armazenará um valor monetário positivo, menor que R\$1.000.000,00;
 - b) ***String textualiza(double valor)***, que receberá o conteúdo numérico de ***valor*** como parâmetro e retornará uma ***String*** equivalente ao valor textual da variável ***valor***.

Ex.: ***double valor = 99.00;***

String s = textualiza(valor);

s receberá “noventa e nove reais”

Exercícios

2. Escrever, em Java, utilizando operadores ternários apenas, a classe **ValorEmTexto**, cujos atributos únicos são:
- a) *valor***, que é uma variável ***double***, que armazenará um valor monetário positivo, menor que R\$1.000.000,00;
 - b) *String textualiza(double valor)***, que receberá o conteúdo numérico de ***valor*** como parâmetro e retornará uma ***String*** equivalente ao valor textual da variável ***valor***.

Ex.: ***double valor = 99.00;***

String s = textualiza(valor);

s receberá ***“noventa e nove reais”***

Exercícios

- Extras, propostos pelo professor em aula, utilizando os conceitos abordados neste material...



Bibliografia Básica

- MILETTO, Evandro M.; BERTAGNOLLI, Silvia de Castro. Desenvolvimento de software II: introdução ao desenvolvimento web com HTML, CSS, javascript e PHP (Tekne). Porto Alegre: Bookman, 2014. E-book. Referência Minha Biblioteca:
<https://integrada.minhabiblioteca.com.br/#/books/9788582601969>
- WINDER, Russel; GRAHAM, Roberts. Desenvolvendo Software em Java, 3ª edição. Rio de Janeiro: LTC, 2009. E-book. Referência Minha Biblioteca:
<https://integrada.minhabiblioteca.com.br/#/books/978-85-216-1994-9>
- DEITEL, Paul; DEITEL, Harvey. Java: how to program early objects. Hoboken, N. J: Pearson, c2018. 1234 p. ISBN 9780134743356.

Continua...

Bibliografia Básica (continuação)

- HORSTMANN, Cay S; CORNELL, Gary. Core Java. SCHAFRANSKI, Carlos (Trad.), FURMANKIEWICZ, Edson (Trad.). 8. ed. São Paulo: Pearson, 2010. v. 1. 383 p. ISBN 9788576053576.
- LIANG, Y. Daniel. Introduction to Java: programming and data structures comprehensive version. 11. ed. New York: Pearson, c2015. 1210 p. ISBN 9780134670942.
- TURINI, Rodrigo. Desbravando Java e orientação a objetos: um guia para o iniciante da linguagem. São Paulo: Casa do Código, [2017]. 222 p. (Caelum).

Bibliografia Complementar

- HORSTMANN, Cay. Conceitos de Computação com Java. Porto Alegre: Bookman, 2009. E-book. Referência Minha Biblioteca:
<https://integrada.minhabiblioteca.com.br/#/books/9788577804078>
- MACHADO, Rodrigo P.; FRANCO, Márcia H. I.; BERTAGNOLLI, Silvia de Castro. Desenvolvimento de software III: programação de sistemas web orientada a objetos em java (Tekne). Porto Alegre: Bookman, 2016. E-book. Referência Minha Biblioteca:
<https://integrada.minhabiblioteca.com.br/#/books/9788582603710>
- BARRY, Paul. Use a cabeça! Python. Rio de Janeiro: Alta Books, 2012. 458 p.
ISBN 9788576087434.

Continua...

ECM251 – Linguagens de Programação I

Desvios Condicionais e Switch-case

Bibliografia Complementar (continuação)

- LECHETA, Ricardo R. Web Services RESTful: aprenda a criar Web Services RESTfulem Java na nuvem do Google. São Paulo: Novatec, c2015. 431 p.
ISBN 9788575224540.
- SILVA, Maurício Samy. JQuery: a biblioteca do programador. 3. ed. rev. e ampl. São Paulo: Novatec, 2014. 544 p.
ISBN 9788575223871.
- SUMMERFIELD, Mark. Programação em Python 3: uma introdução completa à linguagem Phython. Rio de Janeiro: Alta Books, 2012. 506 p.
ISBN 9788576083849.

Continua...

Bibliografia Complementar (continuação)

- YING, Bai. Practical database programming with Java. New Jersey: John Wiley & Sons, c2011. 918 p.
- ZAKAS, Nicholas C. The principles of object-oriented JavaScript. San Francisco, CA: No Starch Press, c2014. 97 p. ISBN 9781593275402.

ECM251 – Linguagens de Programação I

Aula 04 – L1/1 e L2/1

FIM

ECM251 – Linguagens de Programação I

Aula 04 – L1/1 e L2/1

Engenharia da Computação – 3ª série

Desvios Condicionais e Switch-case ***(L1/1 – L2/1)***

2024

Horário

Terça-feira: 2 x 2 aulas/semana

- L1/1 (07h40min-09h20min): *Prof. Calvetti*;
- L1/2 (09h30min-11h10min): *Prof. Calvetti*;
- L2/1 (07h40min-09h20min): *Prof. Igor Silveira*;
- L2/2 (11h20min-13h00min): *Prof. Calvetti*.

ECM251 – Linguagens de Programação I

Desvios Condicionais e Switch-case

Problema

A prefeitura de Rio dos Grilos abriu uma linha de crédito para seus funcionários. O valor máximo da prestação não poderá ultrapassar a 30% do salário bruto de cada um.

Fazer um algoritmo que permita entrar com o salário bruto e o valor da prestação. Daí, informar se o empréstimo pode ou não ser concedido.



ECM251 – Linguagens de Programação I

Desvios Condicionais e Switch-case

Resolvido

```
1 //propositalmente sem acentos
2 import javax.swing.JOptionPane;
3 public class Emprestimo
4 {   public static void main(String[] args)
5     {   //ler os valores do salario e do emprestimo
6         String sSalario = JOptionPane.showInputDialog("Digite o valor do salario: ");
7         String sPrestacao = JOptionPane.showInputDialog("Digite o valor da prestacao: ");
8         //converter para real
9         Double salario = Double.parseDouble(sSalario);
10        Double prestacao = Double.parseDouble(sPrestacao);
11        //calcular relacao entre prestacao e salario e imprimir
12        Double relacao = 100.0*prestacao/salario;
13        System.out.println("Relacao entre parcela e salario: " + relacao + "%");
14        //testar a condicao e informar o resultado
15        if(relacao > 30)
16        {   System.out.println("O valor de parcela excede 30% do" + " salario");
17            System.out.println("Emprestimo negado.");
18        } else
19        {   System.out.println("Emprestimo concedido.");
20        }
21    }
22 }
```

Exercícios



1. Construir um algoritmo que leia dois valores numéricos inteiros e efetue a adição deles; caso o resultado seja maior que 10, apresentá-lo;
2. Construir um algoritmo que leia dois números e efetue a adição. Caso o valor somado seja menor ou igual a 20, este deverá ser apresentado subtraindo-se 5;
3. Entrar com um número e imprimir a raiz quadrada do número caso ele seja positivo e o quadrado do número caso seja negativo;

Exercícios



4. Ler três números e escrevê-los em ordem crescente (suponha que são números diferentes);
5. Construir um algoritmo que indique se o número digitado está compreendido entre 20 e 90 ou não;
6. Entrar com um número e imprimir uma das mensagens: maior do que 20, igual a 20 ou menor do que 20;
7. Entrar com o nome, sexo e idade de uma pessoa. Se a pessoa for do sexo feminino e tiver menos que 25 anos, imprimir nome e a mensagem: ACEITA. Caso contrário, imprimir nome e a mensagem: NÃO ACEITA. (Considerar f ou F);

Exercícios



8. Entrar com dois números e imprimir o maior número (suponha números diferentes);
9. Entrar com um número e imprimir uma das mensagens: é múltiplo de 3 ou não é;
10. Entrar com um número e informar se ele é ou não divisível por 5;
11. Entrar com um número e informar se ele é divisível por 3 e por 7;
12. Entrar com um número e informar se ele é divisível por 10, por 5, por 2 ou se não é divisível por nenhum destes;

Exercícios



13. Ler um número inteiro de 3 casas decimais e imprimir se o algarismo da casa das dezenas é par ou ímpar;
14. Ler um número inteiro de 4 casas e imprimir se é ou não múltiplo de quatro o número formado pelos algarismos que estão nas casas das unidades de milhar e das centenas;
15. Entrar com o ano de nascimento de uma pessoa e o ano atual. Imprimir a idade da pessoa. Não se esqueça de verificar se o ano de nascimento é um ano válido;
16. Crie um algoritmo que leia dois números e imprimir uma mensagem dizendo se são iguais ou diferentes;

Exercícios



17. Entrar com dois números e imprimir o menor número (suponha números diferentes);
18. Entrar com dois números e imprimi-los em ordem crescente (suponha números diferentes);
19. Entrar com dois números e imprimi-los em ordem decrescente (suponha números diferentes);
20. Criar um algoritmo que deixe entrar com dois números e imprimir o quadrado do menor número e a raiz quadrada do maior número, se for possível (suponha números diferentes);

Exercícios



21. Ler três números e escrever o maior número (suponha números diferentes);
22. Ler três números e armazenar o maior número na variável de nome maior (suponha números diferentes);
23. Ler três números e armazená-los em três variáveis com os seguintes nomes, de acordo com seus valores relativos: maior, intermediário e menor (suponha números diferentes);
24. Ler cinco números e identificar o maior e o menor de todos (NÃO suponha números diferentes).

Exercícios



25. Construir um algoritmo que leia três números e imprima se eles podem ou não ser lados de um triângulo. Lembrando que, para ser um triângulo, a soma de dois lados quaisquer deve ser sempre maior que o lado que ficou de fora da soma.
26. Construir um algoritmo que leia três números que representam os lados de um triângulo e que imprima se o triângulo é equilátero (três lados iguais), isósceles (dois lados iguais) ou escaleno (três lados diferentes). Antes, não se esqueça de testar se os três lados realmente formam um triângulo. Informe o usuário caso não formem;

Exercícios



27. Entrar com um verbo no infinitivo e imprimir se o verbo é da 1ª conjugação (terminados em ar), da 2ª conjugação (terminados em er), da 3ª conjugação (terminados em ir), se não está no infinitivo (não termina em r) ou se provavelmente nem é verbo no infinitivo (termina em or ou em ur);
28. Ajude uma escola a montar as divisões do laboratório de programação. Para isso, escreva um algoritmo que leia o nome do aluno e diga em qual divisão ele está respeitando a regra abaixo (dica: use o método ***charAt(posição)***):
- alunos cujo nome começa com as letras de A a K estão na D1;
 - alunos cujo nome começa com as letras de L a N estão na D2;
 - alunos cujo nome começa com as letras de O a Z estão na D3.

Atividade

- Individualmente, resolver os exercícios propostos e apresentar à sala, explicando-a, na próxima aula L1/2 e L2/2, a solução daquele solicitado pelo professor.

Bibliografia (apoio)

- LOPES, ANITA. GARCIA, GUTO. Introdução à Programação: 500 algoritmos resolvidos. Rio de Janeiro: Elsevier, 2002.
- DEITEL, P. DEITEL, H. Java: como programar. 8 Ed. São Paulo: Prentice-Hall (Pearson), 2010.

ECM251 – Linguagens de Programação I

Aula 04 – L1/2 e L2/2

FIM