UNIP
UNIVERSIDADE PAULISTA

LINGUAGEM DE PROGRAMAÇÃO ORIENTADA À OBJETOS – LPOO

Aula 07 – Making Decisions (Review)

TÓPICOS DA AULA

- Make accurate and efficient decisions
- if and if…else statements
- Multiple statements in if and if…else clauses
- Use AND and OR operators and the conditional and NOT operators
- Add decisions and constructors to instance methods

Java Programming – Joyce Farrell (Chapter 5)
Java Programming for Beginners (paginas 100 a 125)

EXERCICIOS DE FIXAÇÃO – AULA ANTERIOR

1. Suppose that you have created a program with only the following variables.

```
int a = 5;
int b = 6;
```

Suppose that you also have a method with the following header:

```
public static void mathMethod(int a)
```

Which of the following method calls are legal?

a. `mathMethod(a);`

b. `mathMethod(b);`

c. `mathMethod(a + b);`

d. `mathMethod(a, b);`

e. `mathMethod(2361);`

f. `mathMethod(12.78);`

g. `mathMethod(29987L);`

h. `mathMethod();`

i. `mathMethod(x);`

j. `mathMethod(a / b);`

EXERCICIOS DE FIXAÇÃO – AULA ANTERIOR

2. Suppose that you have created a program with only the following variables.

```
int age = 34;
int weight = 180;
double height = 5.9;
```

Suppose that you also have a method with the following header:

```
public static void calculate(int age, double size)
```

Which of the following method calls are legal?

a. `calculate(age, weight);`

b. `calculate(age, height);`

c. `calculate(weight, height);`

d. `calculate(height, age);`

e. `calculate(45.5, 120);`

f. `calculate(12, 120.2);`

g. `calculate(age, size);`

h. `calculate(2, 3);`

i. `calculate(age);`

j. `calculate(weight, weight);`

EXERCICIOS DE FIXAÇÃO – AULA ANTERIOR

6. Which of the following is a correct call to a method declared as `public static void aMethod(char code)`?

a. `void aMethod();`

b. `void aMethod('V');`

c. `aMethod(char 'M');`

d. `aMethod('Q');`

7. A method is declared as `public static void showResults(double d, int i)`. Which of the following is a correct method call?

a. `showResults(double d, int i);`

b. `showResults(12.2, 67);`

c. `showResults(4, 99.7);`

d. Two of these are correct.

EXERCICIOS DE FIXAÇÃO – AULA ANTERIOR

1) Crie uma classe chamada **FormLetterWrited** que contem 2 métodos sobrecarregado (overloading) chamados **displaySalutation**():

- *O primeiro método usa um parâmetro String que representa o sobrenome de um cliente e exibe a saudação "Prezado Sr. ou Sra.“ seguido pelo sobrenome.*
- *O segundo método aceita dois parâmetros String que representam um nome e sobrenome, e exibe a saudação "Caro" seguido do primeiro nome, um espaço e o sobrenome.*

Após cada saudação, exiba o resto de um breve carta comercial: “Obrigado por seu pedido recente.”

Escreva uma classe outra classe contendo o método main () que testa cada método sobrecarregado da classe **FormLetterWrited**

EXERCICIOS DE FIXAÇÃO – AULA ANTERIOR

2) Crie uma classe chamada **Calculadora** que contém 4 métodos chamados:

- ***add*** -> recebe como parâmetros 2 números inteiros e retorna a soma dos 2 números:
- ***multiply*** -> recebe como parâmetros 2 números inteiros e retorna a multiplicação dos 2 números :
- ***division*** -> recebe como parâmetros 2 números inteiros e retorna a divisão dos 2 números :
- ***sub*** -> recebe como parâmetros 2 números inteiros:

Para testar os métodos da classe Calculadora, crie uma uma classe com o método main ( ), chamada **CalculadoraTeste**.

EXERCICIOS DE FIXAÇÃO – AULA ANTERIOR

3) Modifique a classe **Calculadora** (exercicio 2) para que ela aceite operações aritméticas com números decimais (double) e não apenas com números inteiros (atualmente)

Obs: Não será permitido modificar os métodos atuais da classe.

Para testar os métodos da classe Calculadora, com as operações artimeticas com numero inteiros e números decimais use a classe **CalculadoraTeste**.

IF and IF.. ELSE

- Em Java, quando você deseja realizar uma ação se uma **expressão booleana for verdadeira, você usa uma declaração if.**

- Se você deseja realizar uma ação quando uma expressão booleana for verdadeira, **mas realizar uma ação diferente ação quando a expressão é falsa, você usa uma instrução if… else.**

- Em outras palavras, as declarações if e else, possibilitam **desviar e executar diferentes trechos do programa**, de acordo com certa condição.

LISTA DE OPERADORES EM JAVA

OPERADOR	FUNÇÃO	OPERADOR	FUNÇÃO
+	Adição	~	Complemento
-	Subtração	<<	Deslocamento à esquerda
*	Multiplicação	>>	Deslocamento à direita
/	Divisão	>>>	Desloc. a direita com zeros
%	Resto	=	Atribuição
++	Incremento	+=	Atribuição com adição
--	Decremento	-=	Atribuição com subtração
>	Maior que	*=	Atribuição com multiplicação
>=	Maior ou igual	/=	Atribuição com divisão
<	Menor que	%=	Atribuição com resto
<=	Menor ou igual	&=	Atribuição com AND
==	Igual	|=	Atribuição com OR
!=	Não igual	^=	Atribuição com XOR
!	NÃO lógico	<<=	Atribuição com desl. esquerdo
&&	E lógico	>>=	Atribuição com desloc. direito
||	OU lógico	>>>=	Atrib. C/ desloc. a dir. c/ zeros
&	AND	? :	Operador ternário
^	XOR	(*tipo*)	Conversão de tipos (cast)
|	OR	instanceof	Comparação de tipos

ESTRUTURAS CONDICIONAIS

- Na linguagem Java temos três recursos para criação de estruturas de decisão:

if - else

operador ternário

switch - case

ESTRUTURAS CONDICIONAIS – IF - ELSE

- **SINTAXE**

```
if  (expressão booleana) {
   <instruções para condição verdadeira>
}
```

```
if  (expressão booleana) {
   <instruções para condição verdadeira>
}  else  {
   <instruções para condição falsa>
}
```

```
if  (expressão booleana) {
    // bloco de código
}  else if (expressão booleana) {
      // bloco de código
}  else {
        // bloco de código
}
```

ESTRUTURAS CONDICIONAIS – OPERADOR TERNARIO

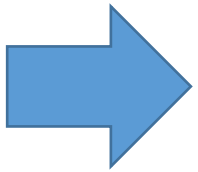
- Similar ao if e else, porém é codificado em apenas uma linha;
  - **SINTAXE**

```
B = (expressão booleana) ? ValorTRUE : ValorFALSE;
```

Exemplos:

```
int numeroDias = // valor entre 1 e 30
System.out.println((numeroDias <= 15) ? "Primeira quinzena" : "Segunda quinzena");

float nota =  // nota informada pelo usuario;
boolean notaValida = (nota > 0 and ≤ 10) ? true: false;
```



Exemplos:

```
int numeroDias = // valor entre 1 e 30
if(numeroDias <= 15) {
    System.out.println("Primeira quinzena")
} else {
    System.out.println("Segunda quinzena")
}
```

ESTRUTURAS CONDICIONAIS – SWITCH - CASE

- Utilizada em programas onde uma variável ou expressão pode assumir uma grande quantidade de valores e há uma ação (ou bloco de ações) para cada valor possível.

- **SINTAXE**

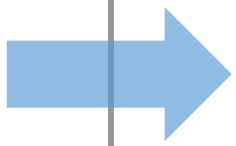
```
switch (<expressão>)  {
        case 1: instruções; break;
        case 2: instruções; break;
        case 3: instruções; break;
        case 4: instruções; break;
        default: instruções;
}
```

A instrução **default** é opcional.

ESTRUTURAS CONDICIONAIS – SWITCH - CASE

- Por exemplo, suponha que você queira exibir o nome da aula de um aluno com base em um número. **Os valores possíveis para o ano é entre 1-4;**

```
if(year == 1)
   System.out.println("Freshman");
else
   if(year == 2)
      System.out.println("Sophomore");
   else
      if(year == 3)
         System.out.println("Junior");
      else
         if(year == 4)
            System.out.println("Senior");
         else
            System.out.println("Invalid year");
```



```
switch(year)
{
   case 1:
      System.out.println("Freshman");
      break;
   case 2:
      System.out.println("Sophomore");
      break;
   case 3:
      System.out.println("Junior");
      break;
   case 4:
      System.out.println("Senior");
      break;
   default:
      System.out.println("Invalid year");
}
```

ESTRUTURAS CONDICIONAIS – SWITCH – CASE

- Multiplos cases:

```
public static boolean isValidSupervisor(String name)
{
   boolean isValid;
   switch(name)
   {
      case "Jones":
      case "Staples":
      case "Tejano":
         isValid = true;
         break;
      default:
         isValid = false;
   }
   return isValid;
}
```

EXERCICIOS DE FIXAÇÃO

1) Crie uma classe em Java, que com as seguintes instruções:
   - **Nome da classe**: InformacoesNutricionais
   - **Método**: *calculaPesoIdeal ( )* : Esse método deverá receber como parâmetro o **peso** e **altura** do paciente e calcular o peso ideal, acordo com o IMC (Indice de Massa Corporal) na tabela abaixo. Após o calculo, o método deverá exibir uma String contendo a descrição do calculo.

Índice IMC	Descrição
Menor que 18,5	Peso abaixo do normal
Entre 18,5 e 24,4	Peso ideal
Entre 24,5 e 29,9	Pré-obesidade
Entre 30 e 34,9	Obesidade classe I
Entre 35 e 39,9	Obesidade classe II (severa)
Maior que 39,9	Obesidade classe III (móbida)

→ Para cálculo do IMC use a fórmula: **imc = peso / altura^2.**

EXERCICIOS DE FIXAÇÃO

Para validar o método, crie uma classe chamada **TesteIMC**, que contém o método main ( )

a) Instanciar a classe InformacoesNutricionais
b) Chamar o método calculaPesoIdeal, com diferentes pacientes.
c) Para obter os valores peso e altura de cada paciente, usar a classe Scanner ou JOptionPane.

Dica: Utiliza uma calculadora de IMC online =)
https://drauziovarella.uol.com.br/obesidade/calculadora-de-imc/

EXERCICIOS DE FIXAÇÃO

2) Crie uma classe em Java, que com as seguintes instruções:

- **Nome da classe**: Universidade
- **Método**: *calculaMediaSemestral( )* : Esse método deverá receber como parâmetro 4 notas do aluno para efetuar a media final do aluno. O retorno do método será uma String que com a descrição da situação do aluno, se acordo com a regra abaixo:

MÉDIA FINAL	SITUAÇÃO
média ≥ 7.0	Aprovado
média ≥ 5.0 e < 7	Exame
média < 5	Reprovado

EXERCICIOS DE FIXAÇÃO

Para validar o método, crie uma classe chamada **TesteCalculoMediaSemestral**, que contém o método main ( )

a) Instanciar a classe Universidade
b) Chamar o método *calculaMediaSemestral*, com diferentes alunos.
c) Para obter os valores das notas de cada paciente, usar a classe Scanner ou JOptionPane.

Dica: Validar se as notas informadas são validas, antes de realizar o calculo da média =)

Ex: notas invalidas: -1 ou maior que 10

PRÓXIMA AULA

- Estrutura de repetição (*Looping*)

Leitura:

- *Java Programming – Joyce Farrell (Chapter 6 – Looping)*