Aula 2

“Olá Mundo” no Portugol

Lembra como imprimir um texto simples no Portugol Studio?

```
programa
{
	funcao inicio ()
	{
		escreva("Olá Mundo!!”)
	}
}
```

“Olá Mundo” no Java

No Java, a estrutura não é tão diferente assim

```
public class Hello_world {

	public static void main(String[] args) {

		System.out.println("Olá Mundo!!");
	}
}
```

Revisando

Já aprendemos:

- O que é Java
- Eclipse IDE
- Nosso primeiro código em Java : “Olá Mundo!”

Novidade, nem tanto!

O que iremos aprender

- Variáveis Primitivas e Controle de Fluxo:
  - Declarando e usando variáveis;
  - Tipos primitivos e valores;
  - Casting;
  - O if e else (se, senão);
  - O while (enquanto);
  - O for (para... até... faça);
  - Escopo de variáveis;

Declarando e usando variáveis

No Portugol Studio, tínhamos os seguintes tipos de variáveis:

- inteiro : Número inteiros -> 1 ; 2 ; 3;
- real : Números de ponto flutuante -> 1.1 ; 3.14 ; 10.3;
- cadeia : Cadeia de caracteres -> “Adoro estudar programação”;
- caracter : Apenas um caractere -> “A”;
- logico : Caractere booleano : verdadeiro, falso;

Declarando e usando variáveis

No Java apenas a notação para cada tipo de variável muda:

- int ou long: Número inteiros -> 1 ; 2 ; 3;
- float ou double : Números de ponto flutuante -> 1.1 ; 3.14 ; 10.3;
- String: Conjunto de caracteres -> “Adoro estudar programação”;
- char : Apenas um caractere -> “A”;
- boolean : Caractere booleano : verdadeiro, falso;

Atribuição de variáveis

A atribuição de valores não é muito diferente do que já vimos no nivelamento

Exemplos:

- int a = 3; // a recebe o valor 3;
- int b = 5; // b recebe o valor 5;
- b = a + b ; // b vira 8 e a continua 3;
- String nome = “Marcelo Collares”; //nome recebe Marcelo Collares;
- double pi = 3,1415; //pi recebe 3,1415;

Vamos praticar a abstração

Alguns conceitos antes de testarmos outro exercício.

- O Java é uma linguagem Orientada a Objetos;
- A programação é feita a partir de classes;
- Cada classe representa o projeto de um objeto;
- A função main() é obrigatória e estará contida dentro de uma classe;
- O Java agrupa as classes em pacotes;
- Para uma classeA usar uma classeB é preciso “importar” a classeB na classeA;

Exercício 2

Em nossa empresa, há tabelas com o quanto foi gasto em cada mês. Para fechar o balanço do primeiro trimestre, precisamos somar o gasto total. Sabendo que, em Janeiro, foram gastos R$ 15000, em Fevereiro, R$ 23000, e em Março, R$ 17000, faça um programa que calcule e imprima o gasto total no trimestre. Siga os passos :

- Crie uma classe chamada BalancoTrimestral com um bloco main, como nos exemplos anteriores;
- Dentro do main (o miolo do programa), declare uma variável inteira chamada gastosJaneiro e inicialize-a com 15000;
- Crie também as variáveis gastosFevereiro e gastosMarco, inicializando-as com 23000 e 17000, respectivamente e utiliza uma linha para cada declaração;
- Crie uma variável chamada gastosTrimestre e inicialize-a com a soma das outras 3 variáveis;
- Imprima a variável gastosTrimestre.

Type Casting

O que acontece com o seguinte trecho de código?

```
double d = 3.1415;
int i = d;
System.out.println("O valor de i é = " + i);
```

Exception in thread "main" java.lang.Error: Unresolved compilation problem:
Type mismatch: cannot convert from double to int

Type Casting

O que acontece com o seguinte trecho de código?

```
double d = 3.1415;
int i = (int)d;
System.out.println("O valor de i é = " + i);
```

```
Saida : O valor de i é = 3
```

Type Casting

Casting possíveis

- A indicação impl. Quer dizer que o cast é implícito e automático, ou seja, você não precisa indicar o cast explicitamente.
- O tipo boolean não pode ser convertido para outro tipo.

PARA: / DE:	byte	short	char	int	long	float	double
byte	----	*Impl.*	(char)	*Impl.*	*Impl.*	*Impl.*	*Impl.*
short	(byte)	----	(char)	*Impl.*	*Impl.*	*Impl.*	*Impl.*
char	(byte)	(short)	----	*Impl.*	*Impl.*	*Impl.*	*Impl.*
int	(byte)	(short)	(char)	----	*Impl.*	*Impl.*	*Impl.*
long	(byte)	(short)	(char)	(int)	----	*Impl.*	*Impl.*
float	(byte)	(short)	(char)	(int)	(long)	----	*Impl.*
double	(byte)	(short)	(char)	(int)	(long)	(float)	----

Exercícios 3

Testando castings

- Declare duas variáveis do tipo int e realize sua soma. Em seguida, realize o casting destes dois inteiros para double para realizar sua divisão.
- Declare dois caracteres : “A” e “Z”, depois realize sua soma e armazene em uma variável do tipo int.
  - Qual é o resultado apresentado?
  - Por que você acha que esse foi o resultado apresentado?

Recordar é viver... e aprender

Desvios condicionais

- No nivelamento vimos que podemos desviar a execução de nosso código utilizando se... senao.

```
se (condicao) {
    // Execute uma parte de código
}
senao {
    // Execute outra parte de código
}
```

Recordar é viver... e aprender

Desvios condicionais

- Em Java para desviar a execução basicamente trocamos o se por if e o senao por else.

```
if (condicao) {
    // Execute uma parte de código
}
else {
    // Execute outra parte de código
}
```

Só isso !?

Recordar é viver... e aprender

Operadores lógicos

- A notação dos operadores lógicos que aprendemos anteriormente também muda:
- O operador E é representado por &&;
- O operador OU é representado por ||;
- O operador NAO é representado por !;

```
if ( (A || B)  && (C ==D)  && !E ) {
    // Execute uma parte de código
}
```

Recordar é viver... e aprender

Laços de repetição - faca enquanto

- Vimos a estrutura do faca... enquanto no nivelamento.

```
faca {
    // Execute uma parte de código
}
enquanto (condicao)
```

Recordar é viver... e aprender

Laços de repetição - do while

- Em Java basicamente trocamos o faca por do e o enquanto por while.

```
do {
    // Execute uma parte de código
}
while (condicao)
```

Recordar é viver... e aprender

Laços de repetição - enquanto

- Vimos a estrutura do enquanto no nivelamento.

```
enquanto (condicao) {
    // Execute uma parte de código
}
```

Recordar é viver... e aprender

Laços de repetição - enquanto

- Em Java basicamente trocamos o enquanto por while.

```
while (condicao) {
    // Execute uma parte de código
}
```

Recordar é viver... e aprender

Laços de repetição - para

- Vimos a estrutura do para no nivelamento.

```
para (inicializacao; condicao; incremento) {
    // Execute uma parte de código
}
```

Recordar é viver... e aprender

Laços de repetição - for

- Em Java basicamente trocamos o para por for.

```
for (inicializacao; condicao; incremento) {
    // Execute uma parte de código
}
```

Não é possível!
Só isso mesmo !?

Escopo de variáveis

- O que define um escopo?
  - Basicamente o uso de chaves { }
- No Java, podemos declarar variáveis a qualquer momento. Porém, dependendo de onde você as declarou, ela vai valer de um determinado ponto a outro;
- Escopo da variável é o nome dado ao trecho de código em que aquela variável existe e onde é possível acessá-la;
- Quando abrimos um novo bloco com as chaves, as variáveis declaradas ou inicializadas ali dentro só valem até o fim daquele bloco;

Exercícios

Vamos pegar alguns exercícios do antigo Portugol e refaze-los: em Java:

- NumeroRepeticoes;
- VerificaTriangulo;
- VerificaTrianguloEspecial;
- SomaNumerosSequenciais;
- TabuadaEspecial;

Revisando

Já aprendemos:

- O que é Java;
- Eclipse IDE;
- Nosso primeiro código em Java : “Olá Mundo!”;
- Variáveis e controle de fluxo;
- Declarando e usando variáveis;
- Tipos primitivos, valores e casting;
- O if e else (se, senão);
- O while e for (enquanto, para… até);
- Escopo de variáveis;