Aspectos Fundamentais

Instituto Federal do Espirito Santo – Ifes
Campus Colatina – Programação Orientada a Objetos I
Professor: Giovany Frossard Teixeira

Tipos de Dados Primitivos

Tipo	Quantidade de Bits	Exemplo
char	16	'a'
byte	8	00000001
int	32	1
short	16	1
long	64	1
float	32	2.99
double	64	2.99
boolean	8	true

Palavras reservadas

- Como já sabemos palavras reservadas não podem ser nomes de variáveis, Java possui diversas palavras reservadas.

abstract	continue	for	new	switch
assert	default	goto	package	synchronized
boolean	do	if	private	this
break	double	implements	protected	throw
byte	else	import	public	throws
case	enum	instanceof	return	transient
catch	extends	int	short	try
char	final	interface	static	void
class	finally	long	strictfp	volatile
const	float	native	super	while

Declaração de variáveis

- É possível declarar variáveis de quaisquer tipos primitivos: char, byte, int, long, float, double ou boolean.
- Assim como em C programas feitos em Java devem ter suas variáveis inicializadas antes de serem utilizadas.
- No próximo slide veremos um exemplo de declaração de variáveis.

Exemplo:

```
package aspectos_fundamentais;

/**
 *
 * @author Giovany
 */
public class AspectosFundamentais01 {
    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        // TODO code application logic here
        int x = 1, y = 2;
        double z = 2.99;
        System.out.println(x);
        System.out.println(y);
        System.out.println(z);
    }
}
```

Declaração de constantes

- Não existem constantes em Java, mas existe um tipo de variável com comportamento semelhante ao de outras linguagens.
  - Em C podemos declarar uma constante da seguinte forma:
    - define pi 3.14;
  - Em Java devemos fazer:
    - final double pi = 3.14;

- Variáveis declaradas com a palavra final não podem ser alteradas, se comportando como constantes.

Comentários

- Java aceita 3 tipos de comentários:
  - Comentando uma única linha: //
  - Para comentar várias linhas: /* */
  - Para comentar várias linhas e gerar documentação com Javadoc: /** */
- Javadoc é um programa gerador de documentação em HTML, fornecido pela Sun junto com o SDK.

Exemplo – Comentários e variáveis final

```
package aspectos_fundamentais;
/**
 *
 * @author Giovany
 */
public class AspectosFundamentais02 {
    /**
     * Exemplo de Declaração de variável final
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        // TODO code application logic here
        final double z = 2.99;
        int x = 1, y = 2;
        System.out.println(x);
        System.out.println(y);
        System.out.println(z);
    }
}
```

Operadores

- Os operadores:
  - Aritméticos
  - Relacionais
  - Lógicos

Operador	Ação
+	Soma (inteira e ponto flutuante)
-	Subtração ou Troca de sinal (inteira e ponto flutuante)
*	Multiplicação (inteira e ponto flutuante)
/	Divisão (inteira e ponto flutuante)
%	Resto de divisão (de inteiros)
++	Incremento (inteiro e ponto flutuante)
--	Decremento (inteiro e ponto flutuante)

São os mesmos da linguagem C.

Operador	Ação
>	Maior do que
>=	Maior ou igual a
<	Menor do que
<=	Menor ou igual a
==	Igual a
!=	Diferente de

Operador	Ação
&&	AND (E)
||	OR (OU)
!	NOT (NÃO)

Exemplo - Operadores

```
package aspectos_fundamentais;

public class AspectosFundamentais03 {

   public static void main(String[] args) {

      int x = 10; int y = 3;

      System.out.println("X = " + x);

      System.out.println("Y = " + y);

      System.out.println("-X = " + (-x));

      System.out.println("X/Y = " + (x/y));

      System.out.println("O resto de X por Y = " + (x%y));

      System.out.println("Inteiro de X por Y = " + (int)(x/y));

      System.out.println("X + 1 = " + (x++));

   }

}
```

Passagem de Parâmetros

- Uma aplicação em Java, assim como em C, pode receber valores a partir da linha de comando (prompt).

- Para executar um programa Java com parâmetros basta utilizar o seguinte comando no prompt:

  java –jar <nome_do_.jar> <argumentos>

  Ex:

  java –jar Java2_Ensino_Didatico.jar aprendendo Java

Exemplo Passagem de Parâmetros

```
package aspectos_fundamentais;


public class AspectosFundamentais04 {


        public static void main(String[] args) {
        // TODO code application logic here
        System.out.println(args[0]);
        System.out.println(args[1]);
    }
}
```

É importante ressaltar que no ambiente NetBeans é necessário setar a classe que contem o método Main do projeto.

O “tipo” String

- Em Java existe um “tipo” String, na realidade String, em Java, é uma classe, mas devido ao sua facilidade de uso começaremos a tratá-la antes mesmo de falar de classes.

- Por enquanto podemos visualizar uma String como um vetor de caracteres, assim como havíamos aprendido em C.

Conversão de tipos

- É muito comum a conversão de tipos. Por exemplo se quiséssemos receber números e não String[] como foi passado no parâmetro args. Isso não seria possível pois os parâmetros de qualquer método em Java são tipados e somos obrigados a fornecer String[] como parâmetro.

- Podemos então para cada String em String[] fazer a conversão para o tipo numérico desejado.

Exemplo – Conversão de tipos

```
package aspectos_fundamentais;

public class AspectosFundamentais05 {

   public static void main(String[] args) {


      double nota1, nota2, trabalho1, trabalho2, media;

      nota1 = Double.parseDouble(args[0]);

      nota2 = Double.parseDouble(args[1]);

      trabalho1 = Double.parseDouble(args[2]);

      trabalho2 = Double.parseDouble(args[3]);

      media = ( nota1 + nota2 + trabalho1 + trabalho2 ) / 4;

      System.out.println("Media = " + media);

   }

}
```

Tabela de conversão de tipos

Declaração da variável a	Conversão para	b recebe a
int a = 20;	float	float b = (float) a;
int a = 20;	double	double b = (double) a;
int a = 20;	String	String b = String.valueOf(a);
float a = 20.1;	int	int b = (int) a;
float a = 20.1;	double	double b = (double) a;
float a = 20.1;	String	String b = String.valueOf(a);
double a = 20.1;	int	int b = (int) a;
double a = 20.1;	float	float b = (float) a;
double a = 20.1;	String	String b = String.valueOf(a);
String a = “23”;	int	int b = Integer.parseInt(a);
String a = “23.3”;	float	float b = Float.parseFloat(a);
String a = “23.3”;	double	double b = Double.parseDouble(a);

Usando o Teclado para Entrada de Dados

```
package aspectos_fundamentais;


import java.io.*;

public class AspectosFundamentais06 {

    public static void main(String[] args) {

        // TODO code application logic here

        String s = "";

        float nota1, nota2, media;

        DataInputStream dado;

        try

        {

            System.out.println("Entre com a nota 1");
            dado = new DataInputStream(System.in);
            s = dado.readLine();
            nota1 = Float.parseFloat(s);

            System.out.println("Entre com a nota 2");
            dado = new DataInputStream(System.in);
            s = dado.readLine();
            nota2= Float.parseFloat(s);

            media = ( nota1 + nota2 ) / 2;
            System.out.println("Media: " + media);
        }
        catch (IOException erro)
        {
            System.out.println("Erro na entrada de dados");
        }
        catch ( NumberFormatException erro )
        {
            System.out.println("Houve erro na conversão,
            digite apenas caracteres numericos");
        }
    }
}
```

Usando o Teclado para Entrada de Dados

Analisando o código:

import java.io.* : Determina que o pacote java.io seja carregado no momento da compilação.

String s = "" : Declarando e inicializando uma String como um tipo primitivo.

DataInputStream dado : Classe para leitura de dados via teclado.

try : palavra reservada utilizada para tratamento de exceções. Se a execução de todo bloco try for executada com sucesso nenhum dos blocos catch será acionado, se houver algum erro um dos catch será acionado e o bloco try não terminará de ser executado.

dado = new DataInputStream(System.in) : criação do objeto dado associando-o a entrada de dados System.in.

s = dado.readLine() : Lê uma linha de dados do teclado.

Usando o Teclado para Entrada de Dados – outra forma

```
package aspectos_fundamentais;

import java.io.*;

public class AspectosFundamentais07 {

    public static void main(String[] args) {

        // TODO code application logic here

        String s = "";

        float nota1, nota2, media;

        BufferedReader dado;

        try

        {

System.out.println("Entre com a nota 1");

dado = new BufferedReader(new
 InputStreamReader(System.in));

s = dado.readLine();

nota1 = Float.parseFloat(s);

System.out.println("Entre com a nota 2");

dado = new BufferedReader(new
 InputStreamReader(System.in));

s = dado.readLine();

nota2= Float.parseFloat(s);

media = ( nota1 + nota2 ) / 2;

System.out.println("Media: " + media);
```

Usando o Teclado para Entrada de Dados – outra forma

```
        }
        catch (IOException erro)
        {
           System.out.println("Erro na entrada de dados");
        }
        catch ( NumberFormatException erro )
        {
           System.out.println("Houve erro na conversão, digite
            apenas caracteres numericos");
        }
    }
}
```

Usando o Teclado para Entrada de Dados – outra forma

- Nessa outra forma de ler dados via teclado utilizamos a classe BufferReader.
- Os códigos são bastante semelhantes.
- É preferível o uso do BufferReader, pois o método readline() da classe DataInputStream é deprecated (obsoleto), no NetBeans métodos deprecated são representados por um traço cortando o nome do método.

Comandos condicionais e repetição

- Os comandos condicionais:
  - if
  - switch
- E os comandos de repetição:
  - for
  - while
  - do-while

São os mesmos da linguagem C e funcionam da mesma forma. Portanto não os estudaremos novamente.

Dúvidas?