

JAVA

Entendendo a estrutura do programa Java.

Todo programa Java, deve conter ao menos uma declaração da forma.

```
public class [nome] {  
  
    public static void main(String args[]) {  
        ...  
    }  
}
```

EXEMPLO

```
public class AloMundo {  
  
    public static void main(String args[]) {  
        System.out.print("Alo Mundo !!!");  
    }  
}
```

System.out.println

É o nome de uma função que serve para escrever informações textuais na tela. Os dados a serem escritos, devem estar delimitados entre os parênteses "(" e ")". "Alo pessoal!" é uma frase. Em computação, uma palavra ou uma frase que tenha função literal é denominada **string**. Em Java, a representação de uma string constante se dá colocando os caracteres entre aspas, por exemplo: "Imagem", "Rio de Janeiro", "Brasília", etc, são strings constantes.

Podemos concatenar dois ou mais strings usando o operador "+". Por exemplo, "Alo " + "pessoal!"

```
public class Numero {  
    public static void main(String args[]) {  
        System.out.println("O valor é " + 29);  
    }  
}
```

Usando variáveis

Uma variável é simplesmente um espaço vago, reservado e rotulado para armazenar dados. Toda variável tem um nome que a identifica univocamente e um valor, que corresponde à informação a ela atribuída. Por exemplo, **int n**; especifica que **n** é o nome de uma variável que pode armazenar um número inteiro como valor. Em geral, num contexto onde aparece o nome de uma variável ocorre a substituição por seu valor. O valor de uma variável pode mudar muitas vezes durante a execução de um programa, por meio de atribuições de valor.

Há diversos tipos de variáveis em Java, correspondendo aos vários tipos de dados aceitos. A discussão sobre esses [tipos de dados](#) será feita daqui a pouco. Antes porém, vamos fazer um pequeno programa que declara uma variável inteira, atribui a ela uma constante, e imprime seu valor na tela.

```
public class Numero {  
    public static void main(String args[]) {  
        int n;  
        n = 17+21;  
        System.out.println("O valor numérico é " + n);  
    }  
}
```

O local onde uma variável está declarada é extremamente importante. Uma variável é conhecida apenas dentro de algum escopo. Por exemplo, uma variável declarada no escopo de uma classe (fora de um método) é conhecida por qualquer método que esteja declarado dentro dessa mesma classe, enquanto uma variável declarada no escopo de um procedimento é conhecida apenas por esse procedimento. Há ainda outros tipos de escopo, como veremos mais adiante.

O sinal "=" é um operador, utilizado para atribuir um valor a uma variável. Por exemplo, **n = 1**; faz com que o valor **1** seja armazenado na variável **n**. Há também os operadores usuais de adição, subtração, multiplicação e divisão de números. Estes são representados pelos símbolos "+", "-", "*" e "/", respectivamente.

Ao executar o programa acima ele escreve:

O valor numérico é 38

Introduzindo comentários no código

Um comentário é uma porção de texto que não tem função para o compilador Java, mas é útil ao leitor humano. Assim sendo, um comentário é identificado mas ignorado completamente pelo compilador Java. A utilidade dos comentários é óbvia: deve conter explicações sobre um particular desenvolvimento do código, permitindo ao leitor compreender claramente o que se deseja realizar.

Os comentários são introduzidos no código Java de duas formas distintas:

- Colocado em qualquer parte do programa e delimitado entre `"/**"` e `"*/"`. Escrevendo `"/"` antes do comentário, que se estenderá até o final da linha.

Por exemplo, o código:

// Este é um exemplo de como somar dois números

```
public class Numero {  
  
    public static void main(String args[]) { /* Método principal */  
        double x,y; // estes sao numeros reais de dupla precisao  
        // System.out.print("x = 2.0"); /* inicializando o "x" */  
        x = 2;  
        y = 3.0; /* iniciando o y,  
                e fazendo y = y+x; */  
        y = y + x;  
        // escrevendo a soma  
        System.out.println("x+y = " + (x+y));  
    }  
} /* fim de Numero */
```

Tudo isso é equivalente ao código:

```
public class Numero {  
    public static void main(String args[]) {  
        double x,y;  
  
        x = 2;  
        y = 3.0; y = y + x;  
  
        System.out.println("x+y = " + (x+y));  
    }  
}
```

TIPOS DE DADOS

Uma visão geral dos tipos que serão abordados neste capítulo:

Tipo	Descrição
boolean	Pode assumir o valor true ou o valor false
char	Caractere em notação Unicode de 16 bits. Serve para a armazenagem de dados alfanuméricos. Também pode ser usado como um dado inteiro com valores na faixa entre 0 e 65535.
byte	Inteiro de 8 bits em notação de complemento de dois. Pode assumir valores entre $-2^7 = -128$ e $2^7 - 1 = 127$.
short	Inteiro de 16 bits em notação de complemento de dois. Os valores possíveis cobrem a faixa de $-2^{15} = -32.768$ a $2^{15} - 1 = 32.767$
int	Inteiro de 32 bits em notação de complemento de dois. Pode assumir valores entre $-2^{31} = -2.147.483.648$ e $2^{31} - 1 = 2.147.483.647$.
long	Inteiro de 64 bits em notação de complemento de dois. Pode assumir valores entre -2^{63} e $2^{63} - 1$.
float	Representa números em notação de ponto flutuante normalizada em precisão simples de 32 bits em conformidade com a norma IEEE 754-1985. O menor valor positivo representável por esse tipo é $1.40239846e-46$ e o maior é $3.40282347e+38$
double	Representa números em notação de ponto flutuante normalizada em precisão dupla de 64 bits em conformidade com a norma IEEE 754-1985. O menor valor positivo representável é $4.94065645841246544e-324$ e o maior é $1.7976931348623157e+308$

Exemplo:

```
public class TiposDeDados
{
    public static void main(String args[])
    {
        byte a = 127;
        short b = 32767;
        int c = 2147483647;
        long d = 9223372036854775807L;
        int e = 0;
        a += 1;
        b += 1;
        c += 1;
        d += 1;
        System.out.println("Valor de a = " + a);
        System.out.println("Valor de b = " + b);
        System.out.println("Valor de c = " + c);
        System.out.println("Valor de d = " + d);
        d /= e; // Vai dar erro porque e = 0
    }
}
```

Operações com inteiros

Podemos realizar uma série de operações com os dados do tipo inteiro. A tabela seguinte mostra uma listagem:

Operação	Descrição
=, +=, -=, *=, /=, %=	Operadores de atribuição
==, !=	Operadores de igualdade e diferença
<, <=, >, >=	Operadores de desigualdade
+, -	Operadores unários
+, -, *, /, %	Adição, subtração, multiplicação, divisão e módulo
+=, -=, *=, /=, %=	Operadores de atribuição com adição, subtração, multiplicação, divisão e módulo
++, --	Incremento e decremento

Referência

<http://www.dm.ufscar.br/~waldeck/curso/java/>

www.caelum.com.br/

java.sun.com/

<http://www.criarweb.com/java/>