

Escrita no ecran

Exemplos:

```
System.out.println(" x = " + x ) ;  
System.err.println(" Indice " + i + " fora dos limites " ) ;
```

Leitura a partir do teclado

Criar um objecto da classe `Scanner` da seguinte maneira:

```
Scanner sc = new Scanner(System.in);
```

Um *scanner* permite ler sequências de caracteres separados por delimitadores.

```
int n = sc.nextInt();           // Lê um inteiro  
double x = sc.nextDouble() ;   // Lê um double
```

Por defeito, os delimitadores são os espaços em branco, tabs e mudanças de linha.

Pode definir-se outro delimitador:

```
sc.useDelimiter("-");
```

É possível testar o tipo do próximo valor a ser lido:

```
if( sc.hasNextDouble() )       // Verifica se próximo valor é um double  
    x = sc.nextDouble() ;
```

Arrays de tipos primitivos

Unidimensionais:

```
int [] a = new int[10];
```

Percorrer um *array* unidimensional (`a.length` – número de elementos do *array* `a`):

```
for( int i=0 ; i < a.length ; i++) {  
    // a[i] ...  
}
```

Bidimensionais

tendo todas as linhas o mesmo número de elementos :

```
double [][] b = new double[3][4];
```

não tendo todas as linhas o mesmo número de elementos :

```
int [][] b = new int[3][];  
b[0] = new int[3];  
b[1] = new int[4];  
b[2] = new int[2];
```

`b.length` – número de linhas do *array* `b`

`b[i].length` – número de elementos da linha `i`

Percorrer um *array* bidimensional:

```
for ( int i=0 ; i < b.length ; i++)
    for ( int j=0 ; j < b[i].length ; j++){
        // b[i][j] ...
    }
```

Números aleatórios

A classe `Math` (pertencente ao *package* `java.lang`) tem funções matemáticas estáticas.

`Math.random()` retorna um valor `double`, gerado aleatoriamente, superior ou igual a 0,0 e inferior a 1,0.

A expressão

```
(int) (Math.random()*49) + 1
```

gera um número aleatório inteiro entre 1 e 49.

Exercícios

1. Escreva uma aplicação que gere um inteiro aleatório entre 0 e 100 e que permita ao utilizador adivinhar o número gerado.
2. Escreva uma aplicação que solicite ao utilizador para pensar num número entre 1 e 100. Através de várias perguntas ao utilizador, a aplicação deve adivinhar esse número.
3. Escreva uma aplicação de consola que crie um *array* de inteiros (perguntando ao utilizador a dimensão do *array* e quais os valores dos seus elementos), imprima o *array* lido e calcule e imprima o maior elemento desse *array*.
4. Defina uma classe que contenha, como um membro, um *array* de inteiros com 20 elementos. Os valores a incluir neste *array* são inteiros aleatórios entre 0 e 100. Cada inteiro gerado aleatoriamente só deve ser colocado no *array* caso este valor ainda não esteja armazenado; isto é, o *array* não deve conter valores em duplicado. Usar uma função booleana para verificar se um determinado valor existe no *array*. A cada valor gerado, chamar esta função para verificar se o mesmo já existe no *array*; se ele existe, deve ser descartado e outro valor deve ser gerado.

A classe deve ter funções para listar:

os 20 valores do *array*

a quantidade de valores que foram gerados em duplicado: total de valores gerados menos valores aproveitados (20).

5. Defina uma classe que represente uma aposta no totoloto. Uma aposta é definida por seis números inteiros compreendidos entre 1 e 49, todos diferentes entre si. A classe deve permitir:
 - Preencher um número de cada vez com vista à construção da aposta.
 - Verificar se a aposta está completa.
 - Preencher automaticamente uma aposta completa.
 - Comparar a aposta corrente com uma aposta ganhadora.
6. Escreva uma aplicação que some matrizes rectangulares.
7. Defina uma classe que contenha, como um membro, uma matriz de **m x n** elementos. Esta classe deve ter funções para alterar os elementos da matriz, calcular a soma de cada linha e a soma de todos os seus elementos e imprimir toda a informação.

Exemplo:

Matriz:

1	0	2	-1	3
4	3	2	1	0
1	-2	3	4	5
8	5	1	3	2

Soma:

5
10
11
19

Total: 45

8. Escreva uma aplicação que calcule e imprima o triângulo de Pascal.