

## ESQUEMA AULA PRÁTICA #3

### □ Tipos de Dados Primitivos

**1 – O programa abaixo não resolve nenhum problema. Apenas serve para testar se consegue identificar erros na sintaxe da linguagem e para exercitar a interpretação das mensagens de erro do compilador.**

- Construa o programa abaixo e corrija os seus erros.

```
public class Valores {
    public static void Main(String[] args){
        numero int;
        int p[] = new int[2];
        Double decNum, rD;
        numero = -100000;
        decNum = 12345,6789;
        System.out.println("O valor da variável inteira é: " + numero);
        System.out.println("O valor da variável real é: " + decNum);
        char letra = "A";
        System.out.println( letra);
        letra = 65;
        System.out.println( letra);
        letra = -97;
        System.out.println( letra);

        Double$z = -1;
        float x=12.5, y=3E30F, zero, rF;
        byte b = -129, rB;
        short 3xpto = -130, sht=9, rS;
        long lng=0xEFFFFFFFFFFFFFFFFF, rL;

        System.out.println(lng);
        rL = lng *10;
        rF = lng + 1;
        rF = x * y / decNum;
        rD = x * y / p[1];
        rF = 0/0;
        rF = sht + b * y * x * lng;
        rD = - b * (sht + zero + x * lng + y * decNum * - numero / letra);
        System.out.println("rD: " + rD );
        rD* = 1E269;
        System.out.println("rD: " + rD );
    }
}
```

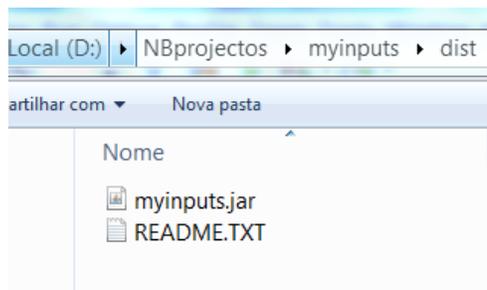
**2 – Faça um programa capaz de gerar uma chave do Totoloto (6 números inteiros distintos entre 1 e 49). Utilize o gerador de números pseudo-aleatórios da classe Math: Math.random(). Este método devolve um valor do tipo double pertencente ao intervalo [0, 1[. (É possível converter um dado tipo num outro compatível usando o operador unário de coerção (casting)).**

□ **Construir um arquivo (JAR) com classes java, e incluí-lo noutro projecto**

- Na folha prática 2, exercício 4, criou num projeto myinputs uma classe Ler. Verifique se completou a classe com métodos para ler cada um dos tipos primitivos da linguagem Java, ( int, float, double, boolean, char, byte e short). Depois de numa classe Teste (ou Main, se gerada automaticamente pelo IDE) testar cada um dos métodos, queremos agora usar a classe Ler, nos projetos a criar daqui adiante.

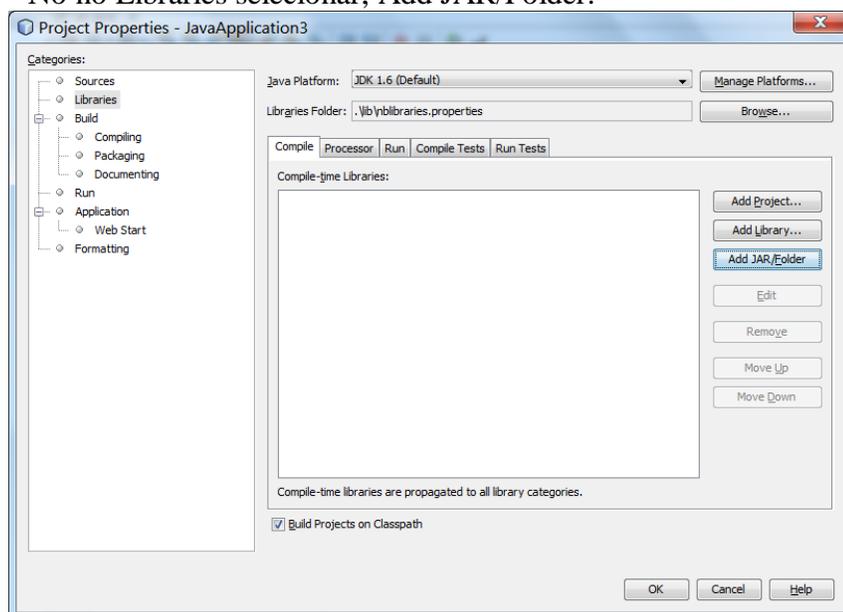
**3** – Para gerar um arquivo com a classe Ler, fazer o seguinte:

- Apagar do projeto a classe que contém o método main (não precisamos dela noutros projetos).
- Selecione o projeto myinputs e clicando no botão direito do rato selecione Clean and Build.
- Verifique na diretoria do projeto que foi criada uma diretoria de nome Dist, e que esta diretoria contém um ficheiro com o nome: **myinputs.jar**.

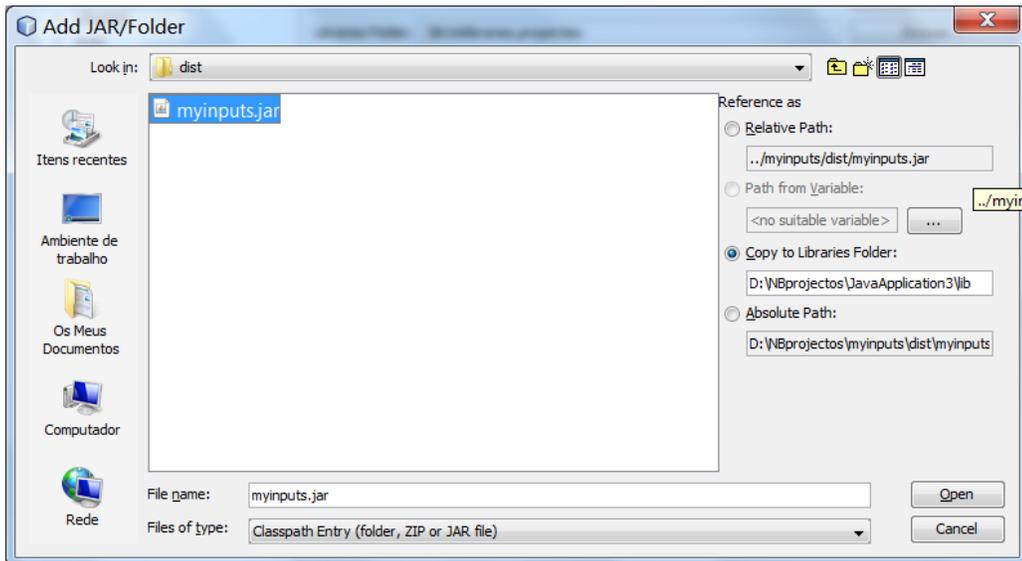


**4** – Para adicionar o arquivo criado (ficheiro JAR) a um outro projeto, fazer o seguinte:

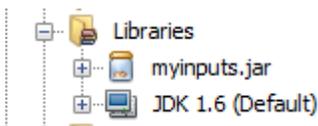
- Selecionar o projeto ao qual pretende adicionar o ficheiro JAR. Clicar no botão direito de rato e seleccionar **Properties**.
- No nó Libraries seleccionar, Add JAR/Folder.



- Selecione o ficheiro **myinputs.jar**, selecione **Open**, e clique **OK**.



- Pode verificar que a pasta Libraries do seu projeto inclui agora o ficheiro myinputs.jar.



- É agora possível usar a classe Ler neste novo projeto.  
 - Experimentar, no método main do seu projeto, chamar o método umInt da classe Ler, fazendo:

```
int i;
i = myinputs.Ler.umInt();
```

Pode omitir-se o nome myinputs se antes do cabeçalho da classe especificar a cláusula de import:

```
import myinputs.*; /o asterisco significa, todas as classes do arquivo.
```

Em alternativa podia fazer: **import myinputs.Ler;**

**5** – Modifique o programa do exercício 2 de modo a que o utilizador indique o número de chaves que pretende. Utilize a classe Ler na leitura do valor inteiro introduzido pelo utilizador.

**6** – Apresente o histograma dos números gerados pelo programa anterior.

- Problemas com Selecção
- Problemas com Iteração
- Subprogramas

Resolva os exercícios que se seguem utilizando os métodos disponibilizados pela classe Ler que implementou anteriormente.

7- Escreva um programa que leia do teclado dois números inteiros,  $p$  e  $u$ , e mostre o resultado de

$$\sum_{i=p}^u i$$

- a) usando um ciclo for
- b) usando um ciclo while
- c) usando um ciclo do-while
- d) estude o comportamento dos programas anteriores no caso em que  $u > p$
- e) modifique os programas anteriores de forma a só aceitar valores  $p$  e  $u$  tais que  $p \leq u$ .

8 – Escreva um programa que leia 3 inteiros do teclado e mostre o maior.

9 – Modifique o programa anterior, reescrevendo-o de tal forma que use um método que receba 3 valores inteiros e devolva o maior. Esse método deverá ser construído na mesma classe que o método main e ser um método de classe (static) a fim de poder ser invocado no método main.

10 - Construa métodos para ler, mostrar e multiplicar matrizes de valores reais (tipo double). O número de linhas e colunas das matrizes deverá ser parametrizado. Comece por construir e testar os métodos para ler do teclado e escrever no ecrã. Depois construa o método para multiplicar matrizes.

11 - Escreva um programa que leia uma matriz (2 x 3) de inteiros do teclado e

- a) Diga quantos elementos ímpares tem a matriz;
- b) Mostre a matriz transposta.

12 – Faça um programa para “inverter” um número inteiro positivo (e.g. 123 passa a 321).

13 – Escreva um programa que leia uma palavra do teclado e mostre a letra com menor valor de código ASCII correspondente.