ESQUEMA AULA PRÁTICA 2

- **Ler dados do teclado**

O método System.in.read() permite ler dados a partir de um buffer associado ao teclado.

1 **- Construa e teste o seguinte programa:**

```
class LerCarater {
  public static void main(String[] args) throws java.io.IOException //!!!!
    {
      char c; int i;
      System.out.print("Introduza um caráter pelo teclado: ");
      i = System.in.read();
      System.out.println("O código ASII do caráter que introduziu é : " + i);
      c = (char)i;
      System.out.println("O caráter que introduziu foi: " + c);
    }  }
```

- Repare que o código anterior não lhe permite ler valores numéricos. Apenas lê um caracter de cada vez.

- **Leitura e escrita usando a classe JOptionPane.**

2 - Em vez de escrevermos as mensagens e resultados dos nossos programas para o canal standard de output (o monitor), podemos escrever para um objecto gráfico. O programa que se segue escreve duas mensagens numa caixa de diálogo do tipo JOptionPane:

```
import javax.swing.*;
public class IOGrafico {
  public static void main(String[] args) {
      JOptionPane.showMessageDialog(null,"A minha primeira caixa de diálogo");
      JOptionPane.showMessageDialog(null,"Adeus");
      System.exit(0);
  }
}
```

a) Teste o programa listado acima.

Usando o mesmo objecto podemos ler valores do teclado. O segmento de código que se segue permite ler um valor inteiro.

```
String s;
int i;
s = JOptionPane.showInputDialog(null,"Introduza um inteiro: ");
// o valor é lido como uma String
i = Integer.parseInt(s);
// a String é convertida para o tipo int
```

b) Construa um pequeno programa que pergunte ao utilizador o seu nome e número e que após a leitura escreva os valores lidos.

3 - Declaração de constantes:

```
    final double PI=3.14159273269;
```

a) Construa um programa que peça ao utilizador o valor do raio e depois calcule o perímetro e a área de um círculo.

b) Construa um programa de teste que lhe permita ler valores do tipo double.

4 – Nos exercícios anteriores não foi feito qualquer tratamento de erros. Se o programa espera um inteiro e o utilizador introduz um valor real o programa termina assinalando um erro.
Vamos implementar uma classe, Ler, que nos permita ler os principais tipos primitivos da linguagem de uma forma mais robusta.
Mais tarde aprenderemos os conceitos que nos permitirão compreender esta classe na totalidade.

a) Defina um projecto com o nome myinputs.

Nota: Repare que no projecto foi criada uma pasta (package) com o nome myinputs e que nessa pasta está uma classe com o nome Myinputs que contém o cabeçalho do método main.

b) Nessa pasta myinputs crie uma classe **Ler** para a qual vai **copiar o código listado abaixo**.

A classe contém um método para ler uma cadeia de caracteres (isto é uma String) e um método para ler cada um dos tipos primitivos da linguagem java (char, byte, short, int, long, float, double e boolean)

Acima do cabeçalho da classe deve inserir a linha: import java.io.*;

Ficando:

```
package myinputs;

import java.io.*;
public class Ler {
// Método para ler uma String:
public static String umaString (){
   String s = "";
   try{
     BufferedReader in = new BufferedReader ( new InputStreamReader (System.in));
     s= in.readLine();
   }
   catch (IOException e){
     System.out.println("Erro ao ler fluxo de entrada.");
   }
   return s;
}
// Método para ler um int:
public static int umInt(){
        while(true){
                try{
                        return Integer.parseInt(umaString().trim());
                }
                catch(NumberFormatException e){
                        System.out.println("Não é um inteiro válido!!!");
                }
        }
}
```

```
// Método para ler um byte:
public static byte umByte(){
        while(true){
                try{
                        return Byte.parseByte(umaString().trim());
                }
                catch(NumberFormatException e){
                        System.out.println("Não é um byte válido!!!");
                }
        }
}
// Método para ler um short:
public static short umShort(){
        while(true){
                try{
                        return Short.parseShort(umaString().trim());
                }
                catch(NumberFormatException e){
                        System.out.println("Não é um short válido!!!");
                }
        }
}
// Método para ler um long:
public static long umLong(){
        while(true){
                try{
                        return Long.parseLong(umaString().trim());
                }
                catch(NumberFormatException e){
                        System.out.println("Não é um long válido!!!");
                }
        }
}
//// Método para ler um float;
public static float umFloat(){
        while(true){
                try{
                        return Float.parseFloat(umaString().trim());
                }
                catch(NumberFormatException e){
                        System.out.println("Não é um float válido!!!");
                }
        }
}
```

```
// Método para ler um double:
public static double umDouble(){
        while(true){
                try{
              return Double.valueOf(umaString().trim());
                }
                catch(NumberFormatException e){
                        System.out.println("Não é um double válido!!!");
                }
        }
}
// Método para ler um char:
public static char umChar(){
        while(true){
                try{
                return umaString().charAt(0);
                }
                catch(Exception e){
                        System.out.println("Não é um char válido!!!");
                }

        }
}
// Método para ler um boolean:
public static boolean umBoolean(){
        while(true){
                try{
                        return Boolean.parseBoolean(umaString().trim());
                }
                catch(Exception e){
                        System.out.println("Não é um boolean válido!!!");
                }
        }
}  }
```

c) Para testar a leitura de dados vamos usar a classe Myinputs onde já tem o cabeçalho do método main.

Para testar o método umaString, adicione à classe Myinputs o código abaixo a negrito:

```
public class Myinputs {
  public static void main(String[] args) {
    System.out.println("Introduza uma string:");
    String s = Ler.umaString();
    System.out.println("A string que introduziu foi: " + s);
```

```
  }
}
```

d) Por analogia, teste agora o método para ler um valor do tipo int.
Ainda para testar o método umInt, experimente introduzir um valor do tipo char. O que acontece?
E se introduzir um valor do tipo double. O que acontece?

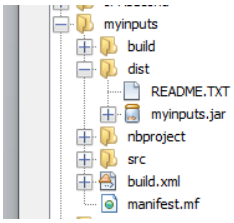
e) Teste o funcionamento de cada um dos outros métodos da classe Ler.

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

Queremos agora usar a classe Ler, nos projetos a criar daqui adiante.

5 – 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 projecto, em files, que foi criada uma diretoria de nome **dist**, e que esta diretoria contém um ficheiro com o nome: **myinputs.jar.**



6 – Usar a classe Ler num outro projeto.

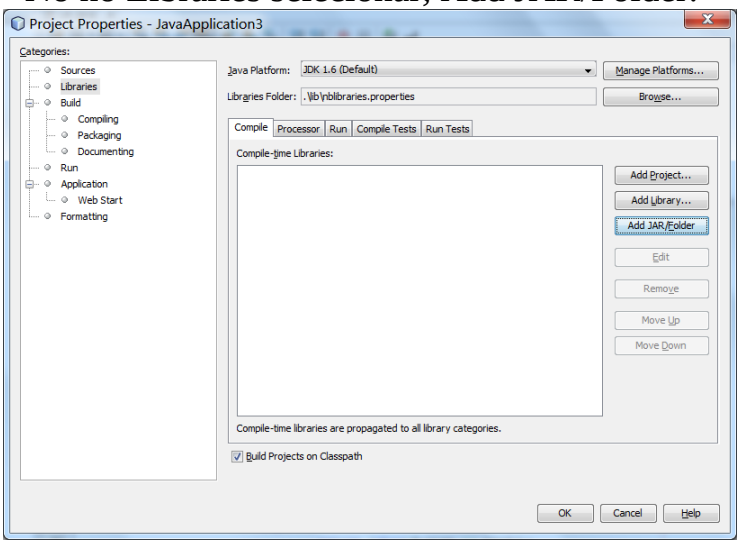
a) Crie um novo projecto de nome **TestarJar**.

b) Para adicionar o arquivo criado (ficheiro JAR) ao projecto **TestarJar**, fazer o seguinte:

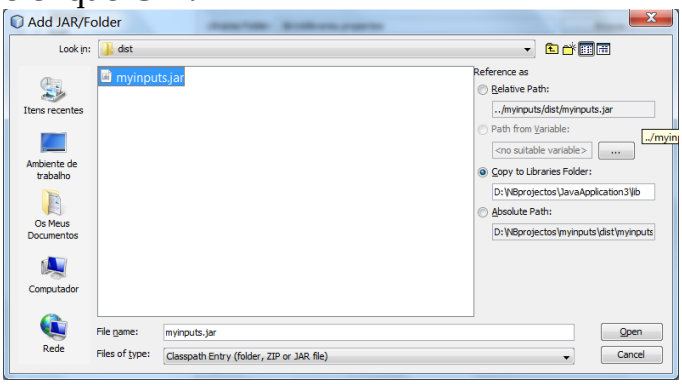
NetBeans 8.2:

- Selecionar o projeto ao qual pretende adicionar o ficheiro JAR. Clicar no botão direito de rato e selecionar **Properties.**

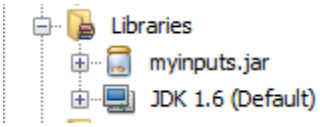
- No nó Libraries selecionar, Add JAR/Folder.



- Selecione na directoria **dist** do projeto o ficheiro **myinputs.jar**, selecione **Open**, e clique **OK.**



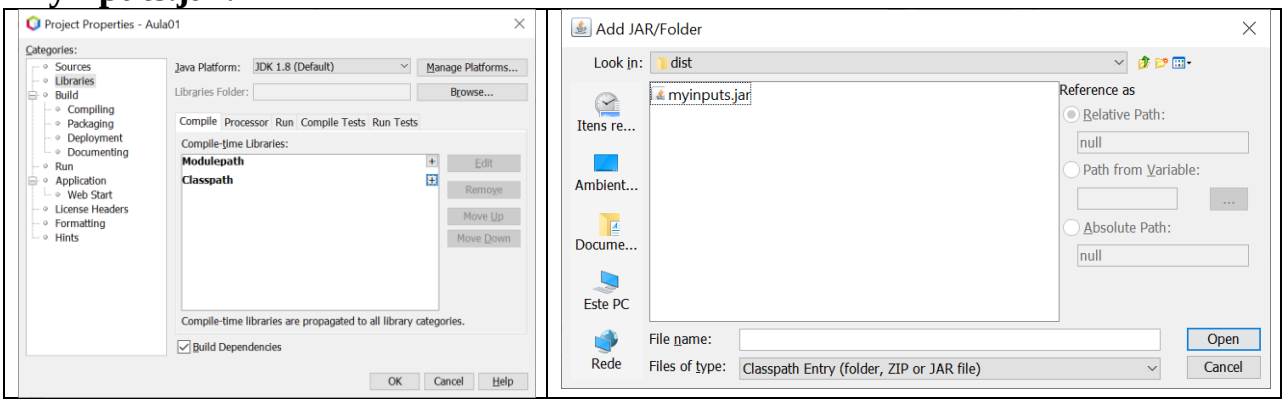
- Pode verificar que a pasta Libraries do seu projeto inclui agora o ficheiro myinputs.jar.

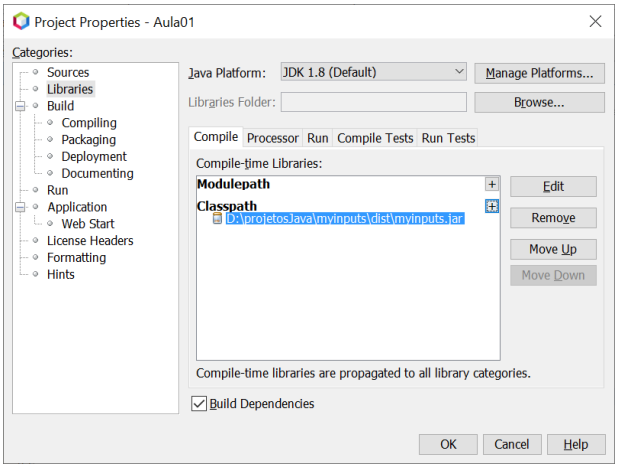


Apache NetBeans 12.*:

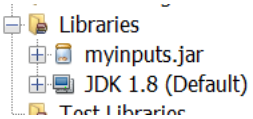
- Selecionar o projeto ao qual pretende adicionar o ficheiro JAR. Clicar no botão direito de rato e selecionar **Properties.**

- No nó libraries / compile, abrir Classpath e seleccionar Add JAR/Folder,
- Posicione-se na diretoria dist do projecto myinputs e seleccione myinputs.jar:





Após <ok> poderá observar que o nó 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 projecto **TesteJar** , 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;**

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

Para fazer em casa:

8 – 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.

a) 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=0xEFFFFFFFFFFFFFFF, rL;
```

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