ESQUEMA AULA PRÁTICA #2

❑ Entrada Básica de Dados

1- Leitura e escrita usando a classe JOptionPane.

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 abaixo escreve duas mensagens numa caixa de diálogodo 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);
  }
}
```

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
```

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

- Construa um programa de teste que lhe permita ler valores do tipo float, double e boolean.

2– A classe `java.lang.System` disponibiliza alguns serviços básicos de entrada/saída através de 3 canais (*streams*) de dados que são variáveis de classe: `in` (teclado), `out` (monitor) e `err`.

Já vimos que System.out inclui os métodos System.out.print() e System.out.println() que nos permitem escrever no monitor, e que System.in inclui o método System.in.read() capaz de ler um byte a partir do teclado.

Uma forma de lermos outros tipos de dados é associar ao System.in um objecto do tipo BufferedReader. Este objecto possui um método que nos permite ler os dados introduzidos pelo teclado como se fossem uma linha de texto. Esse texto pode depois ser convertido para o valor de um dos tipos primitivos da linguagem.

Estude e implemente o programa que se segue:

```
import java.io.*;
public class IOsimples {

  public static void main(String[] args) throws IOException {
    BufferedReader canal;
    canal = new BufferedReader ( new InputStreamReader (System.in));

    System.out.println("Escreva um inteiro: ");

    String s = canal.readLine();
    int i = Integer.parseInt(s);

    System.out.println("O inteiro foi: " + i);
   }
}
```

- Por analogia com o programa anterior construa um programa de estudo que lhe permita ler um valor do tipo double e um valor do tipo boolean.

3 – Também podemos ler valores do teclado usando a classe Scanner associada a System.in. Implemente o exemplo abaixo:

```
import java.util.*;
public class Ler2 {

    public static void main(String[] args) {
            Scanner teclado = new Scanner(System.in);
            System.out.println("Digite um número ");
            int num = teclado.nextInt();
            System.out.println("o número digitado: " + num);
    }
}
```

- Explore a classe Scanner para ler outros tipos de dados.

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.

- Defina um projecto com o nome myinputs e nesse projecto crie uma classe Ler na qual vai começar por implementar o método umaString().

```
import java.io.*;
public class Ler{
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;
}
}
```

- Para testar a leitura de uma String implemente, dentro do mesmo projecto, uma nova classe como a definida abaixo:

```
public class Teste {
  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);
  }
}
```

Usando este método e a classe `Integer` podemos ler uma variável do tipo `int`:

```
public static int umInt(){
    while(true){
        try{
            return Integer.valueOf(umaString().trim()).intValue();
        }
        catch(Exception e){
            System.out.println("Não é um inteiro válido!!!");
        }
    }
}
```

- Inclua este método na classe Ler e teste a leitura de um inteiro.

- Por analogia escreva também os métodos para ler valores do tipo double, float, boolean, char, byte, short e long.

- Teste cada um desses métodos na classe Teste.