

Curso Profissional: Programador/a de Informática**PSD – 11.º ano: UFCD 0816 - Programação de sistemas distribuídos - JAVA****Ficha de Trabalho 6****Ano letivo 22/23**

classes wrapper (permitem que tipos de dados primitivos sejam acedidos como objetos)

Método - parseXxx()

Descrição

Este método é usado para obter dados do tipo primitivo a partir de determinada String. `parseXxx()` é um método estático (static) que pode ter um ou dois argumentos.

Sintaxe:

Variantes do método

```
static int parseInt(String s)
static int parseInt(String s, int radix)
```

Parâmetros

- **s** – string que representa o valor numérico (exemplo: “34.3”)
- **radix** – Deve ser usado para converter a string em inteiro

Valores de retorno

- **parseInt(String s)** – devolve um inteiro (apenas decimal).
- **parseInt(String s, int i)** – devolve um inteiro, dada uma string representativa de um número decimal, binário, octal, ou hexadecimal (radix igual a 10, 2, 8, ou 16 respetivamente) como argumento.

Método - valueOf

Descrição

- Este método é usado para obter dados do tipo String a partir de um dado do tipo primitivo.

Sintaxe (algumas variantes)

```
static String valueOf(boolean b)
ou
static String valueOf(char c)
ou
static String valueOf(char[] data)
ou
static String valueOf(double d)
ou
static String valueOf(float f)
ou
static String valueOf(int i)
ou
static String valueOf(long l)
```

Exemplos

```
public class Test {  
    public static void main(String args[]) {  
        String s="9";  
        int x =Integer.parseInt(s);  
        double c = Double.parseDouble("5");  
        int b = Integer.parseInt("444",16);  
        System.out.println(x);  
        System.out.println(c);  
        System.out.println(b);  
    }  
}  
-----  
import java.io.*; /* significa que é importada qualquer classe java.io  
public class Test {  
    public static void main(String args[]) {  
        double d = 102939939.939;  
        boolean b = true;  
        long l = 1232874;  
        char[] arr = {'a', 'b', 'c', 'd', 'e', 'f','g' };  
        System.out.println("Valor devolvido: " + String.valueOf(d) );  
        System.out.println("Valor devolvido: " + String.valueOf(b) );  
        System.out.println("Valor devolvido: " + String.valueOf(l) );  
        System.out.println("Valor devolvido: " + String.valueOf(arr) );  
    }  
}
```

9
5.0
1092

EXERCÍCIOS (cria um projeto de nome JAVA1_F6):

1. Reproduz as aplicações desta ficha e compila-as e executa-as (melhora os programas, caso aches pertinentes).
2. Reproduz o exemplo da página 8 mas fazendo a leitura dos dados a fornecer pelo utilizador (exceto para o array de char).
3. Escreve um programa em que dadas as 3 notas de um aluno (PORT, MAT, ING), imprima no ecrã se ele foi aprovado, se vai à oral ou se ficou reprovado, sabendo que:
 - A média de aprovação é ≥ 7.0
 - a média para oral é ≥ 5.0 e < 7.0
 - a média do reprovado é < 5.0
4. Escreve um programa que imprima no ecrã, ordenados crescentemente, três números fornecidos pelo utilizador.

Classe Java.Lang.Math

https://www.tutorialspoint.com/java/lang/java_lang_math.htm ou <https://docs.oracle.com/javase/7/docs/api/java/lang/Math.html>

Esta classe contém métodos para a realização de operações numéricas básicas como a exponenciação, logaritmo, raiz quadrada e funções trigonométricas.

É necessário importar a classe para se poderem usar os seus métodos:

```
import java.lang.Math;
```

Classe Java.Lang.String

https://www.tutorialspoint.com/java/lang/java_lang_string.htm ou <https://docs.oracle.com/javase/7/docs/api/java/lang/String.html>

Esta classe representa cadeias de caracteres. Todas as strings dos programas em Java, como por exemplo "abc", são implementadas como instâncias da classe String e são constantes, após a sua criação os seus valores não podem ser alterados. Por serem imutáveis podem ser compartilhados.

Exemplo:

```
String str = "abc";
```

É equivalente a:

```
char data[] = {'a', 'b', 'c'};  
String str = new String(data);
```

Outros exemplos de utilização de strings:

```
System.out.println("abc");  
String cde = "cde";  
System.out.println("abc" + cde);  
String c = "abc".substring(2,3);  
String d = cde.substring(1, 2);
```

Classe Java.Lang.Arrays

https://www.tutorialspoint.com/java/util/java_util_arrays.htm ou <https://docs.oracle.com/javase/7/docs/api/java/util/Arrays.html>

Esta classe contém diversos métodos de classe para manipulação de arrays. Sendo os métodos sort (classificação) e binarySearch (pesquisa) 2 exemplos.

Os métodos desta classe desencadeiam NullPointerException se a referência do array especificado é nulo.

`Arrays.binarySearch(numeros, 1);` //pesquisa a existência do valor 1 no array numeros. Se encontrado devolve a sua posição caso contrário devolve `-(insertion point) - 1`

insertion point - índice do elemento maior que o dado, caso não exista nenhum maior é última posição mais 1

Exercícios

5. Indica o output do exemplo seguinte:

```
public class Cat {  
    public Cat(String nome) {  
        // este construtor/método tem um parâmetro, nome (risca o que é inválido)  
        System.out.println("O nome do gato é : " + nome );  
    }  
}
```

```

public static void main(String []args) {
    // CRIAÇÃO DO atributo/objeto myCat (risca o que é inválido)
    Cat myCat = new Cat( "Félix" );
}
}

```

Relembra: construtor -> um bloco de código que é executado sempre que utilizamos a palavra reservada new.

6. Cria uma classe, no projeto Ficha6, que permita converter temperaturas de Celcius para Farenheit e vice-versa (por exemplo: converter 27 graus Celcius para Farenheit e 80 graus Farenheit para Celcius). Usa uma função (método estático) para efetuar a conversão e onde o utilizador possa optar de quer converter para Celcius (selecionando 'c') ou para Farenheit (selecionando 'f') – usa a estrutura switch. Dá à classe o nome de ConversorTemperaturas.

A temperatura é fornecida pelo utilizador.

Fórmulas:

de conversão de Farenheit para Celcius

$$T_c = (5/9) * (T_f - 32)$$

De Celcius para Farenheit

$$T_f = (9/5) * T_c + 32$$

Os dados são do tipo float

7. Cria uma classe, no projeto Ficha6, que te permita calcular uma potência, dada a base e o expoente e também a raiz quadrada e o quadrado de um valor dado.

(desafio1):

Cria uma classe no projeto criado no exercício 3 onde se construa o seguinte jogo:

O computador gera um número de 1 a 100 (usar o método random da classe Java.lang.math). O utilizador irá tentar adivinhá-lo no menor n.º de jogadas. Após cada jogada a única indicação é se o número a adivinhar está acima ou abaixo do palpite dado.

- Gerar número de 1 a 100;

```

double gn =Math.random();
int num= (int) (100*d);

```

- Usar uma variável do tipo boolean (sucesso) para controlar se o jogador adivinhou ou não o número, com valor inicial false;
- Usar um contador de tentativas para exibir no final para indicar ao jogador em quantas tentativas adivinhou o número;
- Se o utilizador digitar -1 significa que quer desistir;
- Se o valor inserido pelo jogador coincidir com o número gerado sucesso assume o valor true;
- Se o valor inserido pelo jogador for superior ou inferior ao número informar o jogador e voltar a pedir novo número;
- Implementar uma estrutura de repetição para repetir o jogo ou terminar o jogo.

Bibliografia:

<https://www.w3schools.com/java>

Jesus, C. (2013). Curso Prático de Java.Lisboa:FCA

Coelho, P (2016). Programação em JAVA – Curso Completo. Lisboa: FCA

https://www.tutorialspoint.com/java/util/java_util_arrays.htm

https://www.tutorialspoint.com/java/lang/java_lang_math.htm