

# Introdução à Linguagem Java

## Lista 02

21 de março de 2024

**Instruções de entrega:** Criar um único projeto de nome ListaXX-NUSP (onde XX é o número da lista e NUSP é o seu número USP). Entregar no edisciplinas um único arquivo zipado contendo o seu projeto `ListaXX-NUSP.zip`. É esperado que o projeto contenha uma pasta `src` com seus programas.

1. Dentro do package `Math` existe o método estático `random()` com a seguinte especificação: `double random()`; Consulte a documentação disponível e escreva uma classe com um método que imprime uma sequência aleatória de 20 números `double` entre 0 e 1. Note que duas execuções sucessivas da impressão devem gerar a *mesma sequência de 20 números*
2. Faça uma classe com um método que calcula uma função trigonométrica `arco tangente` a sua escolha usando séries de Taylor; use a biblioteca `Math` do Java para criar outro método que computa a mesma função trigonométrica. Faça um terceiro método na mesma classe que compara o funcionamento dos métodos anteriores, devolvendo os valores destas duas funções. Use esta função para calcular o valor de  $\pi$  com 6 casas e faça um teste que imprime a diferença entre estes dois valores.
3. Observe a seguinte classe em Java:

```
class ContaCorrente {
    private String nome;
    public int nconta;
    public double saldo;
    public void setNome(String n){
        nome = n;
    }
    public String getNome(){
        return nome;
    }
}
```

- (a) Modifique a classe, fazendo com que o encapsulamento seja respeitado para os atributos `nconta` e `saldo`;

- (b) Crie dois construtores, um que recebe nome, número da conta e saldo, e outro que recebe apenas o nome e o número da conta;
  - (c) Crie um método `toString` que transforma todos os dados de uma conta em um string que poderá ser impresso na tela;
  - (d) Crie um método `imprime` que imprime (na tela) todos os dados de uma conta, usando uma chamada a `toString`;
  - (e) Crie em outro arquivo uma classe que testa objetos do tipo Conta-Corrente.
4. Adicione ao exemplo anterior um campo estático correspondente ao nome do banco, criado pelo construtor e acessível pela interface. Verifique, escrevendo código testador, que este campo tem o mesmo valor para todos os objetos o qual deve ser revelado no método `toString`.
5. Reescreva a classe de números complexos da Lista 01, com três construtores, que recebe zero, um ou dois parâmetros do tipo `double`; no primeiro caso, o número é 0, no segundo a parte imaginária deve ser 0 e no terceiro é um complexo com parte real e imaginária dados pelos parâmetros. Além desses, `getReal`, `getImaginary`, módulo, ângulo e as 4 operações aritméticas básicas em duas versões; e uma classe que testa a interface pública da sua classe.

Crie um método chamado `addToMe` na classe `Complex`, o qual recebe como parâmetro um objeto da classe `Complex` deve devolver também um objeto da classe `Complex`, com valor `this`. Assim, este método altera o estado interno do objeto, somando o parâmetro de entrada aos seus atributos membros. O comando `c1.addToMe(c2).addToMe(c3)` não deve dar erro de compilação e deve ter o efeito pretendido de transformar `c1` em `c1+c2+c3`. Crie também os métodos `subtractFromMe`, `multiplyMeBy` e `divideMeBy`, os quais devem ter comportamento similares de alterar o estado do objeto para as operações de subtração, multiplicação e divisão.