

# Pacotes e interfaces

Lista04

10 de maio de 2021

## Instruções de Entrega

**Nome do projeto java:** ListaX-NUSP. Sendo X o número da lista e NUSP o seu número usp.

**Arquivo de envio:** Um ÚNICO projeto java zipado, ou seja, enviar o arquivo ListaX-NUSP zipado.

**Seguir as seguintes instruções, sob pena de desconto de nota por código confuso:**

- Não enviar arquivos soltos.
- Não enviar mais de um projeto java.
- Não enviar um projeto com muitas subpastas fazendo com que dê trabalho para encontrar o exercício.
- Não enviar rascunhos dos exercícios. Envie apenas os arquivos que vocês efetivamente querem que sejam corrigidos para que não corram o risco de corrigirmos os arquivos errados.
- Colocar cada exercício dentro de um pacote diferente (o pacote fica dentro do projeto java). Exemplo: dentro do projeto Lista04-NUSP, inserir os pacotes ex1, ex2, ex3, ex4 e ex5. Dentro do pacote ex1 poderia haver os arquivos ContaCorrente.java e ContaCorrenteTeste.java.
- Separar a classe funcional da classe de testes deixando-as em arquivos diferentes.

## Exercícios

1. (valor 2.0) Escreva uma classe de forma a fazer passar os testes FibonacciJUnit. Note que você não deve mudar os testes;
2. (valor 4.0) Escreva classes de forma a fazer passar os testes ExercíciosJUnit, mais uma vez, os testes não podem ser modificados;
3. (valor 2.0) Leia o capítulo 20 do livro de OO com Java (o livro está disponível em:

`http:  
//ccsl.ime.usp.br/files/publications/files/2008/intro-java-cc.pdf`

e faça o exercício 1 desse capítulo. Complemente esse exercício gerando testes (via JUnit) que testem cada uma das implementações dos métodos da interface.

4. (valor 2.0) Crie duas interfaces (e.g. `Incrementável` e `Decrementável`); a primeira com métodos abstratos `Incrementável meIncremente()` e `boolean souNulo()`; a segunda com `Decrementável meDecremente()` e `boolean souNulo()`. Crie uma classe `Inteiro` que implemente estas duas interfaces. Em vez de implementar `Incrementável meIncremente()`, tente implementar `Inteiro meIncremente()` e veja o que acontece; idem para `Inteiro meDecremente()`. Adicione métodos que facilitem os testes a seguir.

Gere testes em JUnit para a interface pública, em que ao menos um teste deve ter uma variável do tipo `Incrementável` e descreva o que acontece ao se tentar incrementá-la, e em outro teste o que acontece ao decrementá-la.

Descreva o que ocorre quando as duas interfaces possuem um **método igual**, com o mesmo nome, mesmos tipos de argumentos e mesmos tipo de retorno.