Bibliotecas

Lista 08

24 de maio de 2024

Instruções de Entrega

Nome do projeto java: ListaX-NUSP1-NUSP2. Sendo X o número da lista para ser realizado em duplas; NUSP1 e NUSP2 são os números USP dos participantes.

Instruções especiais só deste projeto: cada dupla deverá entrar no e-disciplinas **apenas** o endereço do GitHub onde os monitores deverão encontrar o projeto de vocês. Os monitores irão clonar o projeto de vocês, que deverá conter o projeto desta lista. O projeto não deve estar zipado.

Instruções Gerais

Nesta aula, o uso de ferramentas de busca como o Google, a documentação do Java[1], sites sobre programação como o StackOverflow, ou mesmo a Wikipedia será essencial, pois a ideia é vocês pesquisarem como resolver um problema. Por isso, antes de tentarem resolver algum exercício, procurem um pouco e vejam como resolvê-lo de maneira correta

Vocês só estão autorizados a usar o chatGPT no Exercício 1.

Exercício 1 (4,0)

Usando a biblioteca Java Swing[2] do Java, crie uma 'interface' simples para o programa de Fatorial criado nas aulas anteriores. Basta colocar um lugar onde o usuário possa entrar um número, um botão para iniciar o cálculo e um lugar para mostrar o resultado. Separem o código da 'interface' (a *View*) do código que faz o cálculo de Fatorial (o *Controller*). Não esqueçam do tratamento de exceções, o programa de vocês não devem retornar resultados errados nem fazer o usuário se deparar com uma mensagem de erro esquisita!

Exercício 2 (2,0)

Como fazer testes automatizados do exercício anterior? Qual a diferença dessa categoria de teste com os testes usando o JUnit que estávamos fazendo até

[1] `https://docs.oracle.com/javase/7/docs/api/`
[2] `https://wikipedia.org/wiki/Swing_(Java)`

agora? Dica: procurem sobre testes de unidade e testes de aceitação. Fazer um teste "de aceitação" parcialmente automatizado e explicar, em um comentário, como um usuário deve proceder.

Exercício 3 (2,0)

Crie quatro classes que representam multiconjuntos de elementos quaisquer, ou seja, conjuntos em que um elemento pode ocorrer mais de uma vez.

Você pode criar classes desses objetos usando classes existentes nas bibliotecas java. Você pode usar herança de classes existentes, ou usá-las como membros. Crie 4 classes de objetos desse tipo através de `ArrayList`, `Set`, `LinkedList` e `Stack`.

Em cada um dos quatro casos, implemente os métodos `add(T element)`, `equals(Multiconjunto<T> m)`, `addAll(Multiconjunto<T> m)`.

Faça testes automatizados que evidenciem o funcionamento de cada estrutura de dados e suas possibilidades, mostrando como ocorre a "junção" de duas ou mais estruturas, dentre outras funcionalidades.

Exercício 4 (2,0)

Procure na internet sobre o padrão estrutural de projetos **Adaptor**.

Crie um adaptador que permite que o Multiconjunto do exercício anterior seja utilizado como um Conjunto (ignorando as repetições). Use iteradores para percorrer o multiconjunto e insira um iterador no adaptador para que o multiconjunto possa ser percorrido adaptadamente como um conjunto.

Exercício extra (+1,0)

Crie um programa que compara o de desempenho na operação de concatenação de Strings quando se usa a classe `String` e a classe `StringBuffer`. Dê uma breve explicação de porque isso ocorre.