

Teste de Avaliação A

Beja, quarta-feira, 15 de maio de 2019 (14:30-15:45)

Leia com muita atenção as regras seguintes para a realização do teste.

1. Logo que receba o enunciado só poderá sair quando entregar o enunciado e quando terminar o tempo definido para a realização do teste.
2. A entrega no moodle APENAS pode ser feita depois do docente na sala a autorizar.
3. **Apenas pode consultar as resoluções já dadas do GP3 e GP4 disponíveis na página moodle da unidade curricular e indicadas no fórum.**
4. O teste é estritamente individual.
5. O teste é feito num terminal da sala de aula na área do próprio aluno.
6. Com exceção para os docentes presentes na sala, não é permitido comunicar com quaisquer outras pessoas.
7. Não é permitido utilizar a Internet.
8. A troca de qualquer material com outra pessoa tem de ser autorizada pelo docente presente na sala.
9. O não cumprimento de qualquer dos 8 pontos anteriores terá como consequência zero valores na avaliação por teste o que implicará reprovação na unidade curricular no presente ano letivo. Excepcionalmente, poderá ser atribuída uma penalização menor.
10. Durante o teste, o conteúdo no ecrã dos terminais pode ser visto e gravado remotamente.
11. Não se esclarecem dúvidas sobre os enunciados dos problemas propostos: a interpretação do enunciado é considerada como parte da resolução dos problemas propostos. Todas as eventuais dúvidas na interpretação do enunciado deverão ser resolvidas por si próprio. Caso detecte ambiguidades, omissões e/ou incorreções, resolva-as em função da sua melhor interpretação. Caso considere necessário, explique a sua interpretação. Tal será considerado aquando da avaliação.
12. Qualquer ficheiro com erros de compilação conta como não entregue pelo que é classificado com zero valores. Assim, é preferível entregar uma versão que compile ainda que mais incompleta.
13. A resolução é efetuada sob a forma de DOIS projetos IntelliJ com os seguintes nomes:
NNNNN_PrimeiroNomeUltimoNome_P02_Testes2A-QUESTA01.zip
NNNNN_PrimeiroNomeUltimoNome_P02_Testes2A-QUESTA02.zip
14. Os DOIS projectos devem estar DENTRO de uma pasta com o seguinte nome:
NNNNN_PrimeiroNomeUltimoNome_P02_Testes2A
15. Deve fazer um zip da pasta NNNNN_PrimeiroNomeUltimoNome_P02_Testes2A que contém os dois projectos. É este zip com o nome NNNNN_PrimeiroNomeUltimoNome_P02_Testes2A.zip que deve entregar no moodle.
16. As resoluções entregues que não tenham o nome e número de aluno do autor não serão avaliadas.
17. Os critérios de avaliação serão a funcionalidade (o código faz o que é pedido), simplicidade/elegância e o cumprimento das regras de estilo referidas ao longo do semestre. Exceptuam-se os comentários no código, os quais poderão ser utilizados apenas para a eventual apresentação da interpretação prevista no ponto 11.

Enunciado

A resolução das questões seguintes tem obrigatoriamente de ser feita partindo do código base já indicado e disponível na página moodle da unidade curricular.

Note que deve entregar DOIS projetos dentro de um único zip.

1. Questão 1 para resolver num projecto com o nome `NNNNN_PrimeiroNomeUltimoNome_PO2_Teste2A-QUESTA01.zip`

- (a) **(1,0 valores)** Adicione a seguinte funcionalidade ao `Model` do programa do código base: cada posição seleccionada fica com o valor 10 ou 20 alternadamente, ou seja, o primeiro click coloca um 10, o segundo um 20, o terceiro um 10, e assim sucessivamente.
- (b) **(6,5 valores)** Defina um método de teste, para um método na classe `Model` com o cabeçalho `List< List<Integer> > n10And20()`. A lista devolvida pelo método é uma de listas de dois números. Cada lista de dois números contém a quantidade de 10 e a quantidade de 20 na coluna respetiva. Assim, o primeiro elemento da lista de listas é uma lista com a quantidade de 10 e a quantidade de 20 na primeira coluna da grelha, o segundo elemento da lista de listas é uma lista com a quantidade de 10 e a quantidade de 20 na segunda coluna da grelha, e assim sucessivamente.
- (c) **(5,5 valores)** Utilize o método `List< List<Integer> > n10And20()` para mostrar numa *label* (`javafx.scene.control.Label`) quantos 10 e quantos 20 há em cada coluna. O formato do texto na *label* deve ser o seguinte, exemplificado para o um tabuleiro com 5 colunas: 3/0, 0/2, 0/0, 4/1, 0/1; neste caso a primeira coluna tem três 10 e zero 20, a segunda coluna tem zero 10 e dois 20, a terceira coluna tem zero 10 e zero 20, a quarta coluna tem quatro 10 e um 20, a quinta coluna tem zero 10 e um 20. O texto na *label* é actualizado sempre que um botão é clicado. Note que quando um botão é clicado, nessa posição fica um 10 ou um 20, de forma alternada.

2. Questão 2 para resolver num projecto com o nome `NNNNN_PrimeiroNomeUltimoNome_PO2_Teste2A-QUESTA02.zip`

- (a) **(2,0 valores)** Quando é clicado um botão da grelha o programa altera o valor na célula para a soma da respetiva linha e coluna vezes 10. Por exemplo, se clicar na posição (2, 3) a célula deve ficar com o valor $50 = (2 + 3) \times 10$. A alteração deve ser visível na interface gráfica e para tal deve utilizar um novo método na interface `View` com o cabeçalho `void updateCell(Position pos)`.
- (b) **(5,0 valores)** Defina um método de teste, para o método `List<List<Position>> groups()` na classe `Model`. O método considera a lista de posições já anteriormente clicadas e devolve uma partição dessa lista. Ou seja, devolve uma lista de várias listas de posições em que cada uma destas listas de posições é uma sublista da lista de posições clicadas. Em cada sublista, cada posição tem de ser vizinha (na vertical ou horizontal) da posição anterior na lista de posições já anteriormente clicadas. Por exemplo, se a lista de posições já clicadas for (0, 1), (0, 2), (1, 2), (3, 1), (3, 2), (5, 6), (5, 5), (7, 5) a lista de listas deve conter as seguintes quatro listas por esta ordem:
 - (7, 5)
 - (5, 5), (5, 6)
 - (3, 2) (3, 1)
 - (1, 2), (0, 2), (0, 1)

Boa sorte!