

Exame de Paradigmas de Programação

Exame de Época Normal - 02/07/2018
1º Ano da Licenciatura em Engenharia Informática do ISEP

Parte prática (14 valores); Cotações: 1. -20%; 2. - 50%; 3. -30%

Prova sem consulta; Duração: 1h45

Responda a cada pergunta numa folha separada, identificada com número e nome.

Recomenda-se a leitura integral do enunciado antes de começar a responder.

Pretendem-se classes Java para uma aplicação de apoio à gestão de eventos que decorram num de dois tipos de espaços: espaços abertos ou espaços fechados.

Responda às alíneas seguintes tendo sempre em consideração os principais princípios da programação orientada por objetos: abstração, encapsulamento, herança e polimorfismo.

1. Defina a classe `Evento` para representar eventos, caracterizados por `designacao`, `data`, `espaco` e `ocupacao`. Os valores de `designacao` e `data` são do tipo alfanumérico, `espaco` é o espaço onde decorre o evento (a especificar no ponto 2), enquanto que `ocupacao` é um valor inteiro, indicativo da quantidade de lugares já ocupados. Assuma existentes: os métodos reescritos `toString` e `equals`, os métodos de consulta (`get`) e de modificação (`set`) para todas as características, exceto para a `ocupacao`. Especifique:
 - Os atributos da classe e o construtor completo.
 - Uma classe de exceção própria `unchecked`, chamada `OcupacaoInvalidaException`. Esta exceção deve ser lançada quando a modificação da `ocupacao` não for possível.
 - As seguintes funcionalidades:
 - Ocupar um lugar;
 - Libertar um lugar ocupado;
 - Obter a quantidade de lugares disponíveis (diferença entre a lotação do espaço e a sua ocupação no evento em causa);
 - Uma funcionalidade que permita a ordenação dos eventos, por ordem alfabética da `designacao`. No caso de existirem eventos com designações iguais, esses eventos devem ser ordenados pela ocupação de forma decrescente. Esta ordenação será feita recorrendo a código nativo do Java.
2. Elabore classes representativas para os tipos de espaços já referidos e que satisfaçam os seguintes requisitos:
 - Os espaços fechados são caracterizados por `identificador`, `descricao`, `localizacao` e uma lista com a quantidade de lugares que cada fila do espaço possui. A lista deve ser um contentor do tipo `array`.
 - Os espaços abertos são caracterizados por `identificador`, `descricao`, `localizacao` e `lotacao`.
 - O `identificador` de cada espaço deve ser “Ex” sendo `x` um valor sequencial inteiro iniciado em 1.
 - Cada um dos espaços é também caracterizado pela sua finalidade que deverá corresponder a apenas uma das seguintes opções: Desporto, Teatro, Música ou Lazer.
 - Deve ser declarada e implementada uma interface designada `Lotacao`, para determinar a lotação de um espaço. Esta interface deve definir o método `obterLotacao` que determina e devolve a lotação de um espaço.
 - Em cada classe, considere já existentes apenas os métodos `get` e `set` dos atributos de instância.
 - Cada classe deve fornecer um construtor completo e o método `toString` reescrito. Além disso deve fornecer o método `equals` que considera iguais dois espaços, desde que tenham o mesmo `identificador`.
 - Deve ser fornecida uma funcionalidade que permita obter a quantidade de filas de um espaço aberto.
 - Deve ainda ser fornecida uma funcionalidade que permita obter a quantidade de espaços.
3. Defina a classe `ListaEventos` para representar lista de eventos. Os eventos devem ser armazenados num contentor do tipo `ArrayList`. Especifique os atributos da classe e apenas os seguintes métodos:
 - Método `adicionarEvento` para adicionar o evento recebido por parâmetro ao contentor. Se o elemento já existir no contentor, não deve ser adicionado. Deve ser possível identificar o sucesso ou insucesso da operação.
 - Método `obterEventosEmEspacoAberto` para obter um novo contentor com os eventos armazenados na lista que decorram em espaço aberto. Os elementos do novo contentor devem ser ordenados pela designação e ocupação.
 - Método para armazenar numa lista de eventos um contentor previamente guardado num ficheiro binário. O nome do ficheiro binário é passado por parâmetro e é retornado um valor booleano indicando o sucesso/insucesso da ação executada. Considere que pode ser gerada uma exceção do tipo `IOException` ou `ClassNotFoundException` e, nestes casos, faça uma correta gestão do contentor.