Instituto Superior de Engenharia de Coimbra

Programação Avançada – 2015/16

Exame da época de recurso (Prova A) 05-07-2016

Nome: _____

Número de Aluno: _____

Duração: 2h30 *Sem consulta*

Importante: Deve responder às perguntas 1 a 6 nesta folha. A pergunta 7 deve ser respondida na folha de prova. A cotação das respostas é variável, dependendo das alíneas escolhidas. Esta pode variar de valores negativos, para respostas absurdas, até à cotação máxima indicada, para a resposta completamente certa. Pode rasurar a resposta desde que esta seja perceptível.

1 – [12,5%] Considere a Listagem A. Indique, com um X, qual é o resultado da execução do programa. Escolha apenas uma das opções apresentadas.

____ false true 10 2 10 2

____ true true 10 2 10 2

____ false false 10 2 2 10

____ false false 2 2 2 10

____ false true 2 2 2 10

____ false true 10 2 2 10

____ Nenhuma das opções anteriores

2 – [12,5%] Considerando a Listagem D, indique se as seguintes afirmações são verdadeiras (V) ou falsas (F). Alíneas em branco consideram-se não respondidas. Alíneas erradas resultam num desconto de 0.9%, sendo a classificação mínima de 0%.

____ O programa imprime, entre outras coisas, a sequência “f1 f1 f1” durante a sua execução

____ O programa imprime, entre outras coisas, a sequência “f1 f1 f1 f1” durante a sua execução

____ O programa imprime, entre outras coisas, a sequência “f1 f1 f1 f1 f1” durante a sua execução

____ O programa antes de terminar imprime “e1”

____ O programa termina imprimindo “e1 f2: 3.0”

____ O programa termina imprimindo a sequência “e1 f2: 5.0”

____ O programa termina imprimindo “e1 f2: 8.0”

3 – [12,5%] Considere o código da Listagem B. Indique 7 linhas onde ocorre um erro de compilação.

_____ _____ _____ _____ _____ _____ _____

4 – [12,5%] Considere o código da Listagem B assumindo que os erros de compilação foram corrigidos. Indique, com um X, o resultado da execução das instruções das linhas 77, 78 e 79. Escolha apenas uma das opções apresentadas.

____ false false false ____ false false true

____ false true false ____ false true true

____ true false false ____ true false true

____ true true false ____ true true true

5 - Considere o código da Listagem E, em que está definida a funcionalidade da classe ***Semaforo*** destinada a encapsular um semáforo, ou seja, um objecto com cor variável que passa de verde para amarelo, de amarelo para vermelho, de vermelho para verde e assim sucessivamente. Também existe a possibilidade de o semáforo passar a estar desligado e voltar a estra ligado (neste caso, deve ser indicada a cor em que volta a funcionar). Pretende-se uma nova versão da classe ***Semaforo*** que utilize uma máquina de estados orientada a objectos. Considere igualmente a abordagem (padrão) estudada nas aulas laboratoriais em que uma máquina de estados é uma instância de uma classe que, entre outros atributos, inclui o seu estado actual.

a) **[6,25%]** Relativamente à implementação do **estado** da máquina de estados ***Semaforo***, indique se as seguintes afirmações são verdadeiras (V) ou falsas (F). Alíneas em branco consideram-se não respondidas. Alíneas erradas resultam num desconto de 0.4%, sendo a classificação mínima de 0%.

____ É suficiente definir três classes concretas e a(s) base(s) destas classes (classe de base e/ou interface)

____ É suficiente definir quatro classes concretas sem qualquer característica em comum

____ Conceitos como “Vermelho” ou “Verde” representam métodos

____ Conceitos como “PassaTempo”, “Liga” ou “Desliga” representam métodos

____ Todas as classes que representam os estados dispõem de implementações (com ou sem efeito nos dados e estado do semáforo) de todos os métodos que representam as acções a que o semáforo tem que responder

____ Três é o número mais adequado de métodos declarados na classe de base ou na interface comum

____ As implementações dos métodos definidos na classe de base ou na interface comum podem modificar os dados e/ou devolver uma referência para o objecto que representa o estado actual

____ Apenas um dos métodos (representando uma transição na máquina de estados) declarados na classe de base ou na interface comum necessita de receber um argumento

b) **[6,25%]** Represente o **diagrama** da máquina de estados que idealizou, com atribuição de nomes adequados aos estados e às transições/métodos. Deve representar todas as transições que podem ocorrer entre os estados. Cada transição, incondicional ou condicional, deve ser representada por uma seta graficamente distinta.

c) **[5%]** Considere o código da Listagem E, em que está definida a funcionalidade da classe ***Semaforo***. Suponha que vai ser utilizado o padrão comando de modo a encapsular as acções em objectos, tornando possíveis operações de *undo* e *redo*. As afirmações seguintes referem-se à implementação do padrão comando. Preencha os espaços vazios.

- Conceitos como "PassaTempo", "Liga" ou "Desliga" representam _____ (métodos/classes)
- As classes que representam os comandos concretos implementam uma interface ou derivam de uma mesma classe abstracta. Indique os métodos desse tipo base abstracto _____
- O membro "estadoAoLigar" da classe ***Semaforo*** da listagem E deveria ser membro da classe _____ da hierarquia de comandos.

6 – [12,5%] Considere ainda o código da Listagem E, em que está definida a funcionalidade da classe ***Semaforo***. Considere que se pretende mostrar, numa interface de utilizador gráfica, a informação de um semáforo através de uma instância da classe ***Semaforo***. Assuma, igualmente, que se pretende recorrer ao modelo MVC de modo a assegurar que a representação gráfica do semáforo, num painel ***painelDesenho*** do tipo ~~`JPanel`~~ `javafx.scene.layout.`***`Pane`*** ou derivado deste, é actualizada de forma adequada quando o estado do semáforo muda. Tendo em conta estes objectivos, indique quais das seguintes opções são correctas ou adequadas (V ou F em cada uma das opções, ou deixe em branco para não responder). Alíneas erradas resultam num desconto de 0.7%, sendo a classificação mínima de 0%.

____ A classe ***Semaforo*** deve implementar a interface ~~***Observer***~~ ***PropertyChangeListener***

____ A classe ***Semaforo*** deve ser derivada de ~~***Observable***~~ ***PropertyChangeSupport*** ou referida por uma classe derivada de ~~***Observable***~~ ***PropertyChangeSupport***

____ O objecto ***painelDesenho*** pode ser instância de uma classe derivada de ~~***Observable***~~ ***PropertyChangeSupport***

____ O objecto ***painelDesenho*** pode ser instância de uma classe que implementa a interface ~~***Observer***~~ ***PropertyChangeListener***

____ O objecto ***painelDesenho*** pode ser instância de uma classe que não implementa a interface ~~***Observer***~~ ***PropertyChangeListener*** nem é referida por uma classe que implementa ~~***Observer***~~ ***PropertyChangeListener***, desde que tenha definida o método ~~***update***~~ ***propertyChange***

____ O local mais apropriado para colocar o código que tratar da representação gráfica do semáfaro é um método designado ~~***update***~~ ***propertyChange***

~~____ O objecto ***painelDesenho*** não precisa de ter uma implementação da função ***update***, desde que tenha uma implementação da função ***paintComponent***~~

____ O objecto ***painelDesenho***, se tiver uma referência para a classe que deriva de ~~***Observable***~~ ***PropertyChangeSupport*** (***modelo***), já não precisa de se registar como ~~observer~~ *listener* (~~*modelo.addObserver(this)*~~ *modelo.addPropertyChangeListener(this)*)

____ A classe que deriva de ~~***Observable***~~ ***PropertyChangeSupport*** deve ter métodos, como ***passaTempo*** e ***desliga***, que permitam alterar a informação encapsulada pelo modelo. Nestes métodos, é necessário chamar ~~as funções~~ o método ~~***setChanged*** e ***notifyObservers***~~ ***firePropertyChange*** antes de fazer as alterações ao semáforo.

7 – [20%] Pretende-se um programa destinado a modelizar um conjunto de pássaros, especificamente canários e periquitos. Em termos de funcionalidades requeridas, apenas se pretende inserir novas instâncias de pássaros e manter um mapeamento entre os tipos e as respectivas quantidades. Sendo assim, complete o código da **Listagem C** de modo a que o resultado da sua execução seja: *{Periquito=10, Canario=10}*. A classe ***Passaro*** encontra-se declarada como abstracta de modo a não poder ser instanciada, não sendo necessário alterá-la. As classes ***Periquito*** e ***Canario*** requerem a redefinição dos seguintes métodos: ***String toString()***, ***boolean equals(Object o)*** e ***int hashCode()***.

A interface ***Map<K , V>***, sendo ***K*** o tipo da chave e ***V*** o tipo do valor associado, inclui, entre outros, os métodos ***V put(K k, V v)*** e ***V get(Object k)***. Ambos devolvem o valor associado à chave ***k*** fornecida, antes de substituir o valor já existente pelo novo valor ***v*** fornecido no caso do ***put***, ou *null* caso esta não seja encontrada.