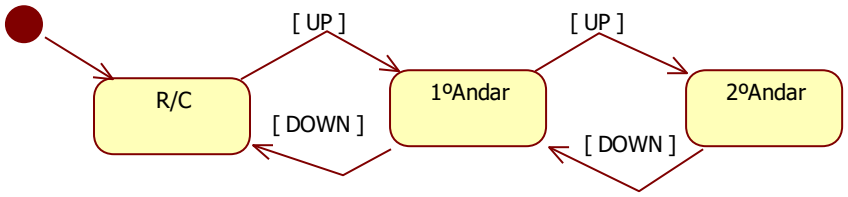
Instituto Superior de Engenharia de Coimbra

Programação Avançada (2020/21)

Exercícios

Máquinas de Estados

1 – Considere o seguinte diagrama de estados, que descreve o funcionamento de um elevador num prédio com 3 pisos.

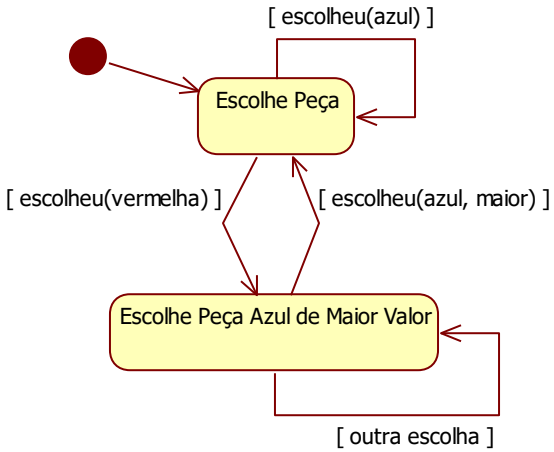


a) Construa as classes necessárias para implementar esta máquina de estados.:
   a. Em cada classe que representa um estado, deve criar métodos adequados para representar as transições possíveis (UP/DOWN).
   b. Na classe que representa a interacção com o utilizador, deve criar um método que apresente as opções adequadas a cada estado (i.e: carregar no botão para subir / descer no 1º piso, mas apenas carregar no botão para subir no R/C). Solicita qual a acção a tomar e actualiza a máquina de estados de forma adequada.
b) Faça as modificações necessárias para garantir a seguinte funcionalidade:
   a. Quando o utilizador carrega no botão para subir/descer, existe uma probabilidade de o elevador ficar avariado de 10% se ocorrer no rés do chão, de 40% no 1º andar e de 75% no 2º andar. Sempre que é detectado um erro, o elevador entra num novo estado em que deixa de aceitar os comandos para subir e descer. O elevador só abandona este estado após a chave de segurança ser aplicada, regressando ao estado em que se encontrava quando o erro ocorreu. (sugestão: utilize uma variável membro do novo estado para armazenar o estado anterior.)

2 – Considere o seguinte jogo:

"O jogador começa o jogo com 10 peças (5 vermelhas e 5 azuis). Cada uma das peças tem um número aleatório entre 1 e 4. Em cada jogada, deve largar uma das peças à sua escolha. Se largar uma peça azul, ganha esse número de pontos. Se largar uma peça vermelha, deve escolher uma peça azul de valor superior e largá-la também. Nesse caso recebe um número de peças aleatórias iguais ao número da peça vermelha que largou. O objectivo do jogo é conseguir a maior pontuação possível. O jogador pode terminar o jogo em qualquer altura."

O diagrama de estados seguinte ilustra o fluxo do jogo:



Dado que as classes seguintes são interdependentes, leia atentamente TODAS as alíneas antes de começar a resolver cada uma delas.

a) Construa as classes necessárias para representar o diagrama de estados. Deve implementar na interacção com o utilizador, correspondendo a cada um dos estados, um método para solicitar a sua opção (por exemplo, qual a peça que pretende largar) .

b) Construa a classe abstracta `Peça`, que representa uma peça com um valor entre 1 e 4. Esta peça deve ser criada com um valor aleatório na gama indicada. Deve conter métodos abstractos com protótipo:

```
abstract public Estado larga();
abstract public Estado largaSegunda(Peca pecaAnterior);
```

onde `Estado` é uma classe abstracta que representa um estado da máquina de estados indicada. O método larga é invocado quando a peça é largada (no estado *escolhe peça*), enquanto que o método `largaSegunda()` é chamado quando a peça é escolhida para ser largada no estado *Escolhe Peça Azul de Maior Valor*.

c) Crie as classes derivadas `PeçaAzul` e `PeçaVermelha`.

d) Construa uma classe `Jogo` que deverá conter a pontuação do jogo, peças actualmente na mão do jogador e a máquina de estados que indica o estado do jogo. Deve conter um método `inicio()` que dá inicio a um novo jogo, bem como outros métodos auxiliares que venha a achar necessários.