

# Resolução / Resumo de correção do p-Fólio

U.C. 21025

**Desenvolvimento de Software**

12 de julho de 2017

## 1.ª Parte (4 Valores)

1.

a) Linhas do código de referência onde há operações de *input*: 01, 20

b) Linhas do código de referência onde há operações de *output*: 11, 18, 19

2. a) Segundo a abordagem de Krasner & Pope (1988): M:02, C:09, V:11, M:12, V:18, C:20, M:23, M:25

b) Segundo a abordagem de Curry & Grace (2008): M:02, C:09, V:11, M:12, V:18, V:20, M:23, M:25

c) Dúvidas ou dilemas com que se debateu para responder às alíneas a) e b) e justificações: Espera-se dos alunos a identificação de aspetos do código cuja classificação como M, V ou C possa ser alvo de argumentação lógica a favor de mais do que uma destas componentes, apresentando esses dilemas e fundamentando as opções que tomaram. Apresentam-se aqui alguns exemplos de aspetos que poderiam mencionar como oferecendo dúvidas:

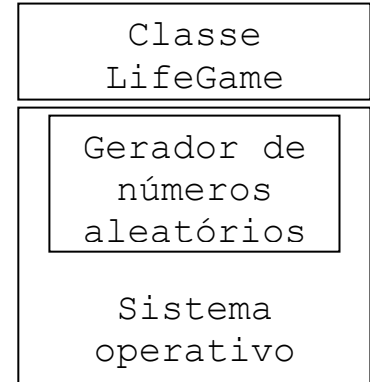
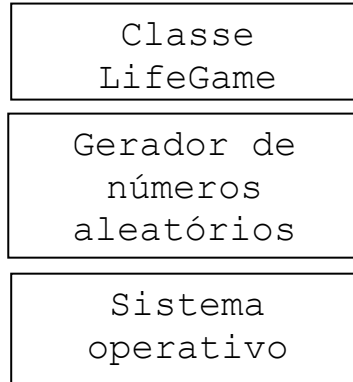
- existência de variáveis usadas para controlo do fluxo, como "exit" e "first", na secção "Valores iniciais"; a minha opção é por integrá-las no Model, pois este pode contar o estado do programa, sendo o controller apenas responsável por reagir a ele; mas poderia ser interpretado por um aluno que estas variáveis deveriam constar do Controller, atribuindo por isso esta secção a ele (justificação que leva a ser considerada correta essa resposta nas alíneas anteriores);
- considerar que embora a linha 18 seja claramente View, por mostrar o estado do jogo, a linha 19, que faz output mas apenas para instruir o processo de input, possa ser feita no controller na abordagem de Krasner & Pope, em particular em sistemas onde seja necessário processar esta instrução e a leitura do input no mesmo módulo (justificação que leva a ser considerada correta essa resposta nas alíneas a).

3. O código anterior depende de vários componentes não explicitados: (1) o sistema operativo, que fornece argumentos de linha de comandos, processa os pedidos de escrita no ecrã e leitura do

teclado; (2) a classe LifeGame, que fornece a lógica do jogo da vida; (3) o gerador de números aleatórios do sistema, a que apoia a classe LifeGame.

a) Desenhe um diagrama onde o código anterior e estes três componentes surjam num padrão de desenho do tipo “Arquitetura estratificada”.

Consideram-se como sendo corretas respostas que mostrem e hierarquia indicada no enunciado e as que optem por incluir o gerador dentro do sistema. Exemplos à direita.



b) Um dos objetivos da etapa de desenvolvimento dedicada à produção de código-fonte é *desenvolver os aspetos que não tenham sido cobertos pela fase de análise e conceção do sistema de software* (Guerreiro, 2015, p. 155). Considere que a fase de análise e conceção produziu, como únicos resultados, a lista de *inputs* e *outputs* da pergunta 1 e a arquitetura estratificada da alínea 3 a). Que aspetos do código-fonte fornecido não foram cobertos? Se a análise e especificação só produziu a lista de inputs e outputs, não cobriu o desenho arquitetural, aspeto central da resposta. Pode ser complementado ou substituído por exemplos como sejam: a forma como os componentes se relacionam ou “comunicação entre intervenientes”, os tipos de requisitos não contidos na mera lista de inputs e outputs, etc. \_\_\_\_\_

4. Considere o código de referência sob a perspectiva da verificação e validação do software. Dê alguns exemplos de falhas, erros e faltas que possam ocorrer nesse código. O aspeto essencial da resposta é dar exemplos que demonstrem a compreensão do significado concreto destes três conceitos. São aceitáveis exemplos que sejam baseados em conceitos gerais de programação, ainda que o comportamento exato na plataforma .Net possa ser diferente. Por exemplo, pode não haver memória para criar o objeto LifeGame na linha 3. Considera-se correto indicar como "falta" tanto a ausência de código para verificação do resultado do new como a não inclusão do new dentro de um bloco try. Considera-se correto indicar como "erro" tanto o facto de lifeGame ter ficado a null como indicar que foi gerada uma exceção de falta de memória. E considera-se correto tanto indicar como "falha" que o programa termina abruptamente ou 'crasha' por não ser apanhada a exceção de falta de memória como pela tentativa de aceder a um método de uma referência null na linha 05. Outros exemplos poderiam ser usados.

### 2.ª Parte (8 Valores)

**Em duas páginas**, efetue uma descrição do trabalho desenvolvido ao longo dos e-fólios, refletindo sobre os temas que se seguem.

- a) A separação em módulos (Model-View-Controller ou o estilo arquitetónico efetivamente seguido);
  - i) qual o principal problema que encontrou;
  - ii) qual a abordagem seguida e como resultou;
  - iii) em retrospectiva, que mudaria na abordagem a esta separação.
  
- b) O recurso a eventos e interfaces para separação de interesses e independência de dados;
  - i) qual o principal problema que encontrou;
  - ii) qual a abordagem seguida e como resultou;
  - iii) em retrospectiva, que mudaria na abordagem a esta técnica.
  
- c) O recurso a exceções personalizadas para gestão de erros.
  - i) qual o principal problema que encontrou;
  - ii) qual a abordagem seguida e como resultou;
  - iii) em retrospectiva, que mudaria na abordagem a esta técnica.

Espera-se encontrar na descrição detalhes demonstrativos de efetiva participação na elaboração do trabalho apresentado nos e-fólios. Os aspetos concretos dependem dos projetos individuais, mas devem ser concretos, vividos, relatos de quem vivenciou o processo:

- a)
  - i) Descrição de um dilema ou decisão difícil na altura de definir a separação entre módulos ou a comunicação entre eles.
  - ii) Explicar como acabou por resolver o dilema, com aspetos concretos de como isso afetou a divisão de responsabilidades entre módulos e/ou as tarefas de comunicação entre eles.
  - iii) Toda as decisões acarretam problemas mais tarde, espera-se aqui uma reflexão sobre aspetos da solução acabaram por atrapalhar ou insatisfazer nas semanas seguintes, à medida que o código se complexificava e evoluía.
- b) Idem, mas para a opção tomada na fase de uso de eventos;
- c) Idem, mas para a opção tomada na fase de uso de exceções.