UNIVERSIDADE AbERTA www.uab.pt

Ministério da Ciência, Tecnologia e Ensino Superior

Exame

U.C. 21179

Laboratório de Desenvolvimento de Software

6 de julho de 2018

INSTRUÇÕES

PARA A RESOLUÇÃO DA PROVA, ACONSELHA-SE QUE LEIA ATENTAMENTE O SEGUINTE:

1. Verifique o exemplar que lhe foi entregue e, no caso de estar incompleto ou com qualquer deficiência, dirija-se ao professor vigilante. A prova termina com a palavra **FIM**.

2. A prova é efetuada sem consulta de outros materiais. No final do enunciado tem material de referência para consulta durante a prova.

3. A prova está dimensionada para um tempo de resolução de duas horas e trinta minutos.

4. A prova é constituída por 3 partes. A cotação de cada uma das partes está indicada, bem como a cotação das alíneas individuais.

5. A resolução deve ser escrita a esferográfica. Utilize, sempre, uma letra legível e não use uma caneta de outra cor que não seja o preto ou o azul. As respostas a lápis não serão consideradas.

6. A interpretação dos enunciados das questões também faz parte da sua resolução, pelo que, se existir alguma ambiguidade, deve indicar claramente como foi resolvida.

Bom Trabalho!

Nome: .....

N.º de Estudante: ..... **B. I. n.º** .....

Turma: ..... **Assinatura do Vigilante:** .....

1.ª Parte (4 Valores)

Código de referência

Este bloco de código é um exemplo do uso da Face API para deteção de rostos em imagens.
Adaptado de https://docs.microsoft.com/en-us/azure/cognitive-services/face/tutorials/faceapiincsharptutorial

```
1   public partial class MainWindow : Window
2       {
3           (...) // Dados sobre as faces detetadas na imagem.
4           Face[] faces;
5           String[] faceDescriptions;
6           (...)
7           // Reage ao clique no botão "Procurar".
8           private void BrowseButton_Click(object sender, RoutedEventArgs e)
9           {
10              // Obter do utilizador o ficheiro de imagem a analisar.
11              var openDlg = new Microsoft.Win32.OpenFileDialog();
12              openDlg.Filter = "Imagem JPEG (*.jpg)|*.jpg";
13              bool result = openDlg.ShowDialog(this);
14
15              // Regressar em caso de cancelamento.
16              if (!result) { return; }
17
18              // Mostrar a imagem.
19              Uri fileUri = new Uri(openDlg.FileName);
20              BitmapImage bitmapSource = new BitmapImage();
21              (...)
22              bitmapSource.UriSource = fileUri;
23              (...)
24              FacePhoto.Source = bitmapSource;
25              (...)
26          }
27
28          // Mostra a descrição da face quando o rato se lhe sobrepõe.
29          private void FacePhoto_MouseMove(object sender, MouseEventArgs e)
30          {
31          (...)
32          }
33      }
```

As alíneas seguintes são todas de resposta curta.

1.

a) Indique as linhas do código de referência onde há operações de *input*: _____ 0,5

b) Indique as linhas do código de referência onde há operações de *output*: _____ 0,5

Caso tenha tido dúvidas de interpretação do código que lhe afetaram a resposta, exponha-as aqui:

_____

_____

_____

_____

Nome: .....

N.º de Estudante: ..... **B. I. n.º** .....

Turma: ..... **Assinatura do Vigilante:** .....

2. Suponha que queria reescrever o código de referência segundo o estilo arquitetónico MVC. Indique como distribuiria pelos componentes as tarefas descritas nos comentários desse código.

Por ex., se considerar que a tarefa "`//Dados sobre as faces detetadas na imagem.`" (linha 03) corresponde ao Controller, escreva "C: 03", se acha que corresponde ao Model, escreva "M: 03", para indicar que corresponde à View, escreva "V: 03".

a) Segundo a abordagem de Krasner & Pope (1988): _____ 0,3

b) Segundo a abordagem de Curry & Grace (2008): _____ 0,3

c) Explique as dúvidas ou dilemas com que se debateu para responder às alíneas a) e b) e justifique as principais opções que tomou ao dar as suas respostas: _____ 0,4

_____

_____

_____

_____

_____

3. O código anterior depende de componentes não explicitados. Para que funcione a linha 24 (`FacePhoto.Source = bitmapSource;`), é preciso que (i) esta atribuição lance um evento que origine a atualização do ecrã; ou (ii) o código da propriedade Source faça essa atualização do ecrã diretamente; ou (iii) haja um ciclo na aplicação durante o qual se vai usar a propriedade Source para atualizar o ecrã. 1

Considere que há pelo menos três componentes: (1) A aplicação global; (2) a classe FacePhoto; (3) o componente que gere a visualização. Se sentir necessidade, pode indicar outros.

Desenhe três <u>arquiteturas encaminhamento e filtragem</u>, respetivamente para os casos i, ii, e iii:

_____

Nome: .....

N.º de Estudante: ..... **B. I. n.º** .....

Turma: ..... **Assinatura do Vigilante:** .....

4. Considere o código de referência nas linhas 15 e 16. Se `result` for `false`, tal indica que o utilizador clicou no botão "Cancelar".

a) Se `result` for `false`, considera que tal pode ser um falha, um erro ou uma falta? Ou não? Justifique. _____ 0,4

_____

_____

b) Escolha uma linha que não as 15 e 16 e indique uma falha que possa ocorrer na execução dessa linha. _____ 0,3

c) Escolha uma linha que não as 15 e 16 e aponte uma falta neste código (justifique)._____ 0,3

_____

_____

_____

2.ª Parte (8 Valores)

5. Tomando como base a sua resposta à pergunta 2, opte por uma das variantes do estilo arquitetónico MVC. Reescreva o código no estilo arquitetónico MVC, considerando o seguinte:

a) Indique a variante escolhida por si para responder a esta pergunta e desenhe o diagrama MVC correspondente, **substituindo os rótulos genéricos que encontra nos materiais de referência com os aspetos específicos deste caso**. 2

b) Defina as classes que pertencem aos componentes Model, View e Controller. Não utilize eventos, delegados, interfaces, nem exceções, apenas os atributos e métodos habituais. 3

c) Escreva o código de um método `Controller.Main();` que controle o funcionamento global da aplicação. 3

Nome: .....

N.º de Estudante: ..... **B. I. n.º** .....

Turma: ..... **Assinatura do Vigilante:** .....

3.ª Parte (8 Valores)

6. Tomando como base a sua resposta à pergunta 5, reescreva o código para obter independência de componentes e separação de interesses, da seguinte forma:

a) Defina e utilize eventos e delegados para reportar alterações nos módulos. 2

b) Defina e utilize interfaces para que a comunicação de dados ocorra com menos dependências entre módulos. 3

c) Defina exceções e escreva o código que as lança em caso de erro. Escreva igualmente o código que as apanha, respeitando as responsabilidades de cada módulo no MVC. 3

FIM

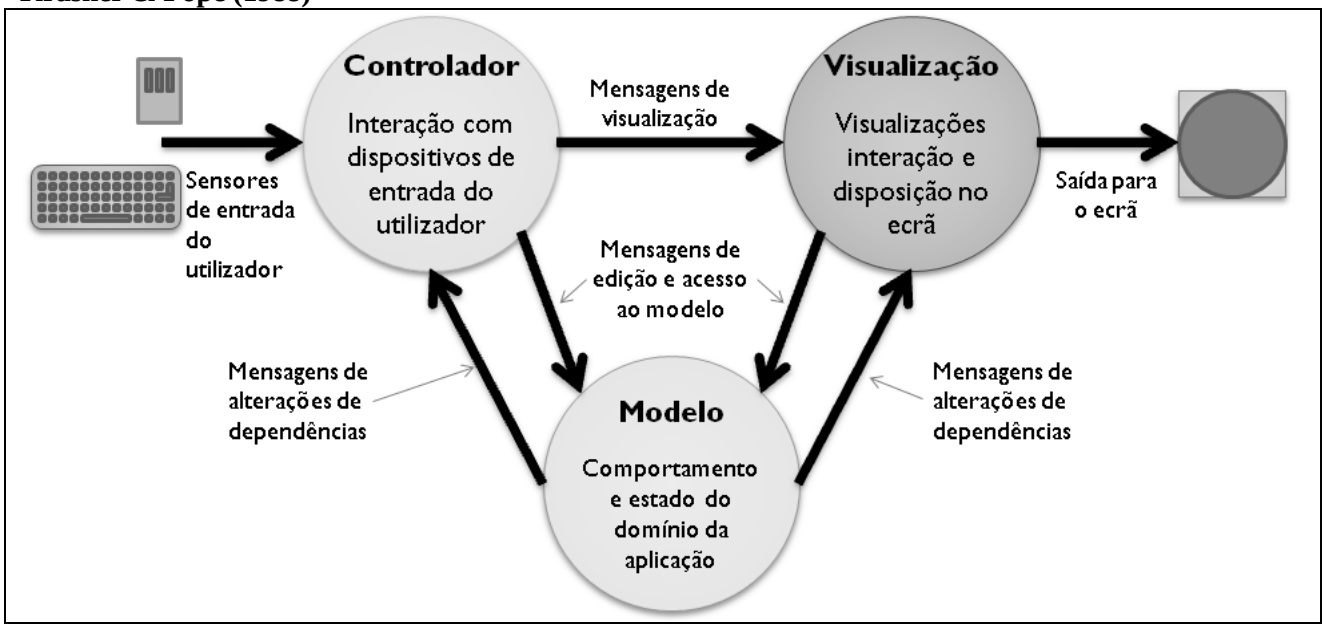
Nome: .....

N.º de Estudante: ..... **B. I. n.º** .....

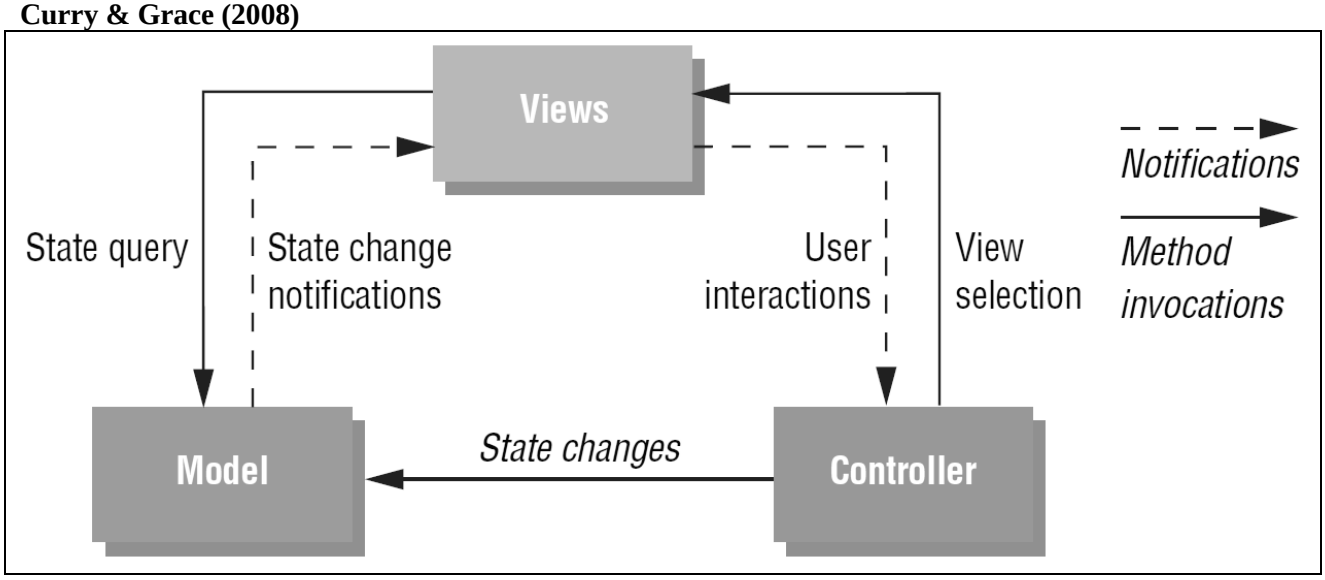
Turma: ..... **Assinatura do Vigilante:** .....

Material de referência

Krasner & Pope (1988)



Curry & Grace (2008)



Apoios de sintaxe

```
throw new ClasseDaExcecao(ParametroDaExcecao);
catch (ClasseDaExcecao VariavelDaExcecao) { }
public delegate TipoDevolvido NomeDoTipoDeDelegado(Parametro1,Parametro2);
NomeDoTipoDeDelegado delegado = new NomeDoTipoDeDelegado(MetodoAtribuido);
public event NomeDoTipoDeDelegado NomeDoEvento;
NomeDoEvento += delegado;
NomeDoEvento += new NomeDoTipoDeDelegado(MetodoAtribuido);
```