

Número:

--	--	--	--	--	--

Curso: _____

Nome: _____

iscte

INSTITUTO
UNIVERSITÁRIO
DE LISBOA

Introdução à Programação

1º semestre – 2023/2024

Exame de 1ª época

Duração: 2 horas e 0 minutos (+30 minutos de tolerância)

Cotação: 13 valores (Parte I — 4 valores; Parte II — 9 valores)

Observações e Regras:

- A prova tem duas partes que devem ser respondidas nas respetivas folhas;
- É permitido entregar o teste escrito a lápis;
- Não serão respondidas questões após 30 minutos do início do teste;
- Não é permitido sair da sala durante a realização do exame;
- Só é permitido entregar ou desistir após a primeira hora;
- Podem e devem ser utilizadas funções, procedimentos ou classes desenvolvidas em questões anteriores, e não é necessário escrever quaisquer instruções de *import*.
- É estritamente proibido manipular quaisquer aparelhos eletrónicos (e.g. telemóveis).
A violação desta regra implica a anulação do exame;

Assuma que estão apenas disponíveis as classes seguintes:

Color

`new Color(r, g, b)` — constrói uma cor dados os valores de RGB

`getR()` — obtém o valor de R;

`getG()` — obtém o valor de G;

`getB()` — obtém o valor de B

`getLuminance()` — obtém a luminância (brilho) que é um inteiro correspondente ao tom de cinzento (inteiro) correspondente à cor que invoca a operação

ColorImage

`new ColorImage(w, h)` — constrói uma imagem preta com largura *w* e altura *h*

`getColor(x, y)` — obtém a cor (Color) do pixel (x, y)

`setColor(x, y, color)` — altera o pixel (x, y) para a cor *color*

`getWidth()` — obtém a largura da imagem em número de pixels

`getHeight()` — obtém a altura da imagem em número de pixels

`drawCenteredText(x, y, text, size, color)` — desenha o texto dado, centrado em (x, y)

`drawText(x, y, text, size, color)` — desenha o texto dado justificado à esquerda a partir da posição (x, y), com o tamanho e cor dados

String

`String.valueOf(..)` — devolve uma *String* *s*

`s.toCharArray()` — devolve um vetor de caracteres criado a partir da *String* *s*

`s.length()` — obtém o tamanho da *String* *s*

`s.charAt(index)` — obtém o caracter na posição *index* da *String* *s*

Math

`Math.abs(a)` — devolver o valor absoluto de a

`Math.max(a, b)` — devolve o valor máximo entre a e b

`Math.min(a, b)` — devolve o valor mínimo entre a e b

`Math.random()` — devolve um valor decimal aleatório no intervalo [0,0 ; 1,0 [

`Math.sqrt(n)` — devolve a raiz quadrada de n

`Math.pow(base, expoente)` — devolve $\text{base}^{\text{expoente}}$

PrintWriter

`new PrintWriter(new File(".."))` — cria o writer

`writer.print(..)` — escreve em modo texto através do writer

`writer.println(..)` — escreve em modo texto através do writer, e termina a linha

`writer.close()` — fecha o writer

Scanner

`new Scanner(...)` ou `new Scanner(new File(".."))` — cria o scanner

`scanner.hasNextLine()` — devolve true se houver uma linha para ler

`scanner.hasNext()` — devolve true se houver uma palavra para ler

`scanner.hasNextInt()` — devolve true se houver um valor inteiro para ler

`scanner.nextLine()` — devolve uma String com a linha lida

`scanner.next()` — devolve uma String com a próxima palavra

`scanner.nextInt()` — devolve um valor inteiro lido

`scanner.nextDouble()` — devolve um valor decimal lido

`scanner.close()` — fecha o scanner

`System.out.print()` ou `System.out.println()` — escreve texto na consola

Exceções

`IllegalArgumentException()` — para incorreções relacionadas com argumentos

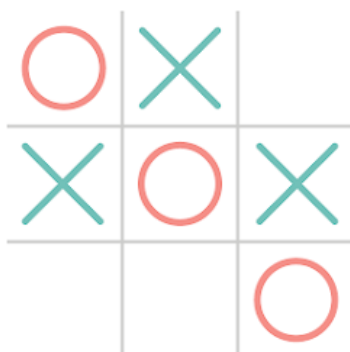
`IllegalStateException()` — para incorreções relacionadas com o estado dos objetos

`NullPointerException()` — para incorreções relacionadas com argumentos com valor null

`FileNotFoundException()` — para incorreções relacionadas com o acesso a ficheiros

PARTE I

Desenvolver uma classe estática `GaloAux` com procedimentos e/ou funções para a criação, manipulação, e visualização do tabuleiro de um “Jogo do Galo”. As cruzeiras ‘X’ devem ser desenhadas a verde, os círculos ‘O’ a vermelho, e a grelha deve ser cinzenta. Pode optar por desenhar a grelha a partir do canto superior esquerdo ou centrada na imagem. Os espaços vazios são codificados com ‘ ‘.



Questão 1 (1,5 valores)

1.1 Desenvolva um procedimento estático auxiliar que permite desenhar uma grelha cinzenta, numa imagem dada, com espessura unitária e com distância entre linhas da grelha dada.

```
static void drawGrid (ColorImage img, int sizeGrid){
```

1.2 Desenvolva um procedimento que permite preencher uma imagem dada com a grelha e as letras 'X' e 'O' do jogo, dados o tamanho da grelha e a matriz de caracteres previamente preenchida, com dimensão (3x3). Os espaços vazios não devem ser preenchidos.

```
static void drawBoard (ColorImage img, int sizeGrid, char [][] matrizJogo){
```

Questão 2 (0,75 valores)

Desenvolva um procedimento que desenhe um quadrado de lado size centrado na posição (x0, y0) dada, numa imagem dada, e desenhe por baixo, centrada, a palavra dada, ambos com cores dadas.

```
static void drawSquareAndText (ColorImage img, int x0, int y0, Color  
colorSquare, Color colorText, int size, String palavra){
```

Questão 3 (1,5 valores)

3.1 Desenvolva uma função estática que devolve a primeira palavra de uma String dada.

e.g. getFirstWord("João Gonçalves Zarco") → "João"

```
static String getFirstWord (String palavra){
```

3.2 Desenvolva uma função estática que devolve a última palavra de uma String dada.

e.g. getLastWord("João Gonçalves Zarco") → "Zarco"

```
static String getLastWord (String palavra){
```

Questão 4 (0,75 valores)

Desenvolva uma função que verifica se um vetor é composto apenas por caracteres repetidos, e em caso afirmativo, devolve esse caracter, caso contrário devolve '_'.

```
static char allEqual (char[] v){
```

Questão 5 (0,5 valores)

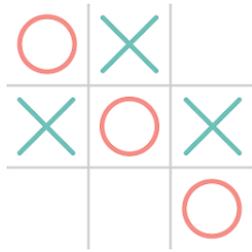
Desenvolva uma função que dada uma matriz de char e o índice de uma coluna, devolve essa coluna sob a forma de um vetor.

```
static char[] getColumn (char[][] m, int column){
```

PARTE II

Questão 1 (4 valores)

Desenvolva/complete a classe de objetos JogoDoGalo. Assuma que “X” joga primeiro.



```
class JogoDoGalo {  
  
    private char [][] galo;  
    private int jogada;  
    private ColorImage img;
```

- a) (0,75) Crie um jogo novo apenas com a grelha e com todas as posições vazias de duas formas: (1) dada uma imagem inicial e uma largura de grelha; (2) dada apenas a imagem inicial, e assumindo que a grelha divide a imagem em partes iguais;
- b) (0,75) Desenvolva uma função que permite fazer uma jogada, dados os índices da posição (i,j) na matriz de jogo, devendo ser lançadas as exceções apropriadas caso a posição seja inválida ou a posição esteja ocupada; a jogada será ‘X’ nas jogadas ímpares e vice-versa.
- c) (0,5) Desenvolva uma função que verifica se alguém fez uma linha completa e nesse caso devolve ‘X’ ou ‘O’, consoante o vencedor, caso nenhuma coluna esteja completa devolve ‘_’;
- d) (0,5) Desenvolva uma função que verifica se alguém fez uma coluna completa e nesse caso devolve ‘X’ ou ‘O’, consoante o vencedor, caso nenhuma coluna esteja completa devolve ‘_’;
- e) (0,5) Desenvolva uma função que verifica se alguém preencheu qualquer uma das diagonais completa, e nesse caso devolve ‘X’ ou ‘O’, consoante o vencedor, caso nenhuma diagonal esteja completa devolve ‘_’;
- f) (1,0) Desenvolva uma função que verifica se o jogo acabou, quer por vitória de algum jogador, quer por o tabuleiro estar completo, caso haja um vencedor, deve devolver ‘X’ ou ‘O’, consoante o vencedor, em caso de empate de devolver ‘E’ e enquanto não acabar o jogo deve devolver ‘_’; Caso o jogo acabe deve também ser escrita uma mensagem na consola com o vencedor ou uma mensagem de empate.

Questão 2 (4 valores)

Desenvolva/complete a classe de objetos ReservasAviao que gere e guarda as marcações no ficheiro “reservations.dat”.



O objetivo é gerir as reservas de assentos num avião, sendo que as mesmas devem ser registadas num ficheiro de texto com o seguinte formato CSV (com os campos separados por vírgulas, e uma linha por cada marcação), a identificação na reserva deve conter apenas o primeiro nome e o apelido:

```
“João Zarco,12,B”  
“Maria Silva,22,J”  
..
```

Considere as seguintes classes IdPassenger e Seat

```
class Seat {  
    private IdPassenger passenger;  
    private boolean state;           // true <-> reservado  
}  
  
class IdPassenger {  
    private String fullName;  
    private String reservationCode;  
    private String passportNumber;  
    ..  
}
```

Desenvolva:

- os construtores dos objetos da classe IdPassenger e Seat;
- os atributos e o construtor de ReservasAviao que deve receber o número de filas e de assentos por fila do avião;
- um método que devolva uma String, apenas com a primeiro e último nome correspondentes ao atributo fullName;
- um método para obter o número de lugares ocupados numa fila dada;
- um método que dado um vetor de objetos da classe IdPassenger, permita atribuir assentos automaticamente, de forma sequencial, nos assentos livres;
- um método para obter o número total de lugares vagos;
- um método que escreva no ficheiro o registo de todas as reservas;