

## Introdução à Programação | 2020/2021 (1º semestre)

### Projeto Individual

- Data de entrega: até às 24h de dia 11 de Dezembro de 2020 (por upload na plataforma de e-learning)
- Data da discussão: até à semana de 14 a 18 de Dezembro de 2020 (na respetiva aula teórico-prática)

**Aviso:** Os alunos podem partilhar e/ou trocar ideias entre si, sobre os trabalhos e/ou resolução dos mesmos. No entanto, o trabalho entregue deve corresponder ao esforço individual de cada aluno. O projeto é individual, e em nenhum caso deve ser copiado código que será entregue. A deteção de código copiado será realizada por software especializado bastante sofisticado. Casos de plágio óbvio serão penalizados com a anulação do projeto, o que implica a reprovação à Unidade Curricular (UC). Adicionalmente, a situação será reportada à Comissão Pedagógica da ISTA/Conselho Pedagógico do ISCTE-IUL. Serão penalizados da mesma forma tanto os alunos que fornecem código como os que copiam código de outros.

### Introdução

O objetivo deste projeto é criar um editor de camadas para fazer um poster ou um jornal de parede.



Figura 1 – Exemplo de um Poster

Nesse sentido, deverão ser desenvolvidas três classes:

- uma classe estática com funções e procedimentos úteis para criação e manipulação de imagens;
- uma classe de objetos *Layer* (camada);
- uma classe de objetos *Poster* para a manipulação das camadas.

A solução desenvolvida deverá fazer uso das classes *ColorImage*, *ImageUtil* e *Color* disponibilizadas nas aulas práticas e da classe *String* do Java. Não é permitida a utilização de outras classes.

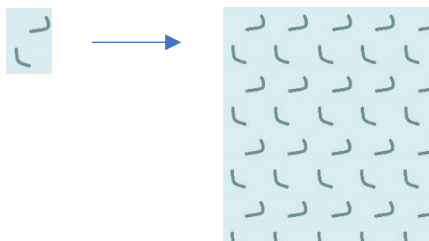
Recomenda-se que o desenvolvimento do projeto decorra por etapas, as quais devem corresponder às classes que se apresentam em seguida. Não será apropriado avançar para uma etapa mais avançada sem ter a etapa anterior minimamente completa.

## Parte 1: Funções e procedimentos úteis para a criação e manipulação de imagens

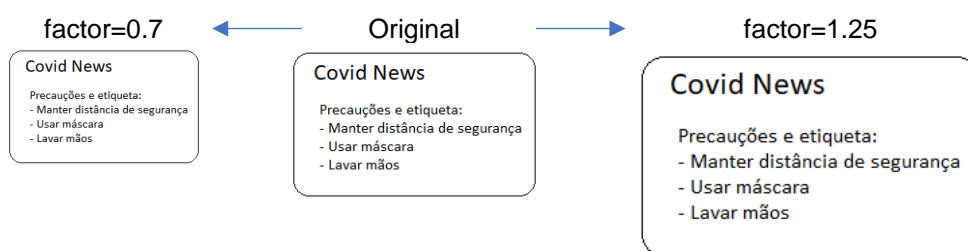
**Objetivo:** Desenvolver uma classe estática com procedimentos e/ou funções para a criação e manipulação de imagens a cores (*ColorImage*). Considere que em todo o editor é definida a cor *branco puro* (255,255,255) como a cor correspondente à transparência.

Esta classe deve ter funções e/ou procedimentos para (desenvolva todas as funções e/ou procedimentos adicionais que achar necessário):

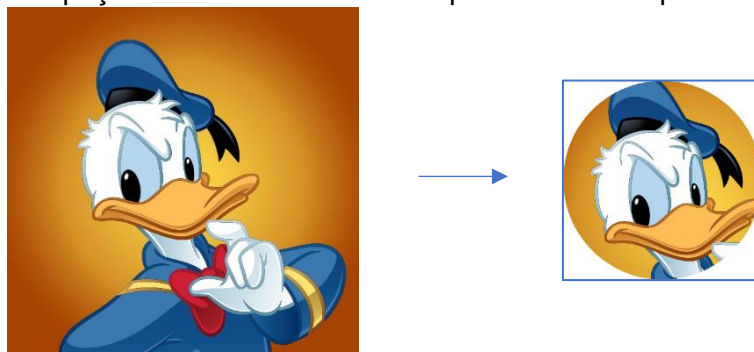
1. copiar a parte não transparente de uma imagem para cima de outra imagem.
2. criar, uma imagem (*ColorImage*) de fundo do poster com base numa imagem padrão dada que será replicada como padrão de fundo da tela (pode incluir transparência).



3. criar, uma cópia escalada da imagem original a partir de um *factor (double)* dado;



4. criar, uma imagem quadrada correspondente o uma seleção circular de uma imagem original, a partir das coordenadas do ponto central e do raio de seleção, o espaço remanescente deve ser pintado de transparente (branco);



5. criar, uma cópia em tons de cinzento de uma imagem dada (*ColorImage*).  
Utilize a seguinte fórmula para obter o tom de cinzento:  
 $(0.3 * R) + (0.59 * G) + (0.11 * B)$ .

## Parte 2: A classe de objetos *Layer*

**Objetivo:** Desenvolver uma classe de objetos que representa o conceito de *Camada*.

Um objeto do tipo *Layer* (que contém um atributo *ColorImage*) pode ser criado de três maneiras: i) fornecendo uma imagem a cores, o factor de escala, as coordenadas da posição (onde queremos colocar a imagem relativamente ao canto superior esquerdo do poster), e um nome identificador (*String*); ii) fornecendo a imagem e a posição (o factor de escala será unitário e o nome ficará vazio); iii) fornecendo apenas a imagem (a imagem será posicionada relativamente ao ponto (0,0)). Após a criação de um objeto do tipo *Layer*, deve ser possível:

1. modificar o nome da imagem;
2. modificar a escala e o posicionamento da imagem no poster;
3. definir a imagem como activa ou inactiva;
4. devolver a imagem completa da camada, com a imagem original escalada e posicionada, e com a transparência dada pela cor branca.

## Parte 3: A classe de objetos *Poster*

**Objetivo:** Desenvolver uma classe de objetos que representa o conceito de *Poster*.

Um objeto do tipo *Poster* pode ser criado fornecendo a largura e altura do Poster (ou tela) em pixéis. Um objecto do tipo *Poster* é compostos por uma colecção de objectos do tipo *Layer* e por uma imagem correspondente à tela. Após a criação de um objeto do tipo *Poster*, deve ser possível:

1. criar (ou substituir) a *layer[0]* dada a imagem com o padrão de preenchimento;
2. adicionar (no fim da colecção) uma *layer* nova;
3. remover uma *layer* da colecção numa posição dada, deslocando as restantes;
4. inserir uma *layer* na colecção numa dada posição, deslocando as restantes;
5. trocar as posições de duas *layers* na colecção;
6. obter a imagem final do poster, com a sobreposição de todas as *layers* activas obedecendo à transparência.



Figura 2 – Exemplo de uma imagem final de um Poster

## Exceções

No projeto, deverá lançar uma exceção de cada um dos seguintes tipos:

1. `IllegalArgumentException`;
2. `NullPointerException`;
3. `IllegalStateException`.

## Avaliação e Entrega

A realização do projeto é obrigatória para obter aprovação à UC. Não haverá qualquer possibilidade de obter aprovação à UC sem realizar o projeto. Além do peso na nota final, a classificação no projeto define limites máximos para a nota final na UC. O projeto será classificado da seguinte forma:

- A, Muito bom (>80%): a nota final obtida na UC poderá ser no máximo 20;
- B, Bom (<=80%): a nota final obtida na UC poderá ser no máximo 16;
- C, Suficiente (<=60%): a nota final obtida na UC poderá ser no máximo 12;
- D, Não aprovado (<45%): implica reprovação à UC.

O projeto será inicialmente avaliado em termos funcionais, i.e., se as funções e procedimentos produzem os resultados esperados e os objetos dos tipos das classes a desenvolver têm o comportamento esperado, independentemente da forma como estão implementados, de acordo com os seguintes pesos:

- classe estática com funções e procedimentos úteis para a manipulação de imagens, 35%;
- classe *Layer*, 20%;
- classe *Poster*, 40%;
- Exceções, 5%.

Desta primeira avaliação resultará uma classificação (A, B, C ou D). Em função da qualidade do código poderá ser aplicada uma penalização que implica descer um nível na classificação, p.e.:

- Classificação funcional C com má qualidade de código, é despromovida para D;
- Classificação funcional A com má qualidade de código, é despromovida para B.

Os alunos poderão obter feedback juntos dos professores das respetivas turmas sobre o progresso do projeto e qualidade do código, e deverão seguir as recomendações dadas.

**Os projetos só poderão ser entregues em suporte eletrónico (somente ficheiros .java desenvolvidos), por upload na plataforma de e-learning na data de entrega, e diretamente ao professor da turma a que o aluno pertence no dia da discussão. O projeto só será considerado entregue caso tenha sido recepcionado correctamente por estas duas vias.**