

## Introdução à Programação | 2019/2020 (1º semestre)

### Projeto Individual

- Data de entrega: até à semana de 9 a 13 de dezembro de 2019 (na respetiva aula teórico-prática)
- Data da discussão: até à semana de 16 a 20 de dezembro de 2019 (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 desenvolver alguns conceitos base de editor de gráficos bidimensionais.

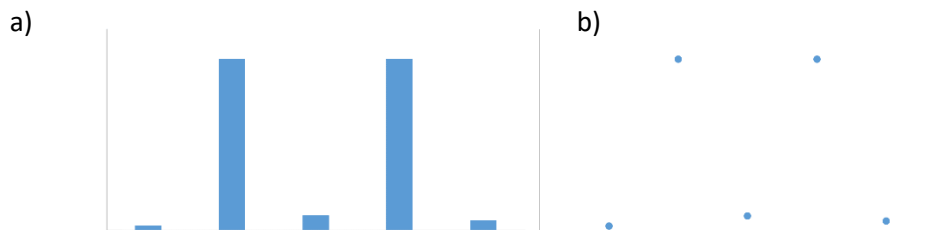


Figura 1 - a) gráfico de colunas; b) gráfico de dispersão.

Nesse sentido, deverão ser desenvolvidas três classes:

- uma classe estática com funções e procedimentos úteis para a criação e manipulação de imagens representativas de gráficos;
- uma classe de objetos Gráfico;
- uma classe de objetos para sobrepor gráficos.

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

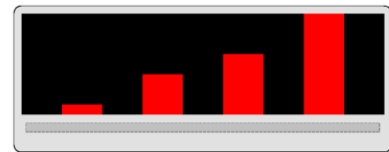
Recomenda-se que o desenvolvimento do projeto decorra por etapas, as quais devem corresponder às classes que se apresentam em seguida. Não será apropriado avançar para uma etapa mais avançada sem ter a etapa anterior minimamente completa.

## Parte 1: Funções e Procedimentos úteis para a criação e manipulação de imagens representativas de gráficos

**Objetivo:** Desenvolver uma classe estática com procedimentos e funções para a criação e manipulação de imagens a cores (*ColorImage*) representativas de gráficos.

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

1. criar um gráfico de colunas 2D verticais (devolvendo uma instância da classe *ColorImage*) dando como argumentos:
  - o o vetor de inteiros;
  - o a largura das colunas;
  - o o espaçamento entre colunas;
  - o a cor (instância da classe *Color*) das colunas (a cor de fundo é preta).

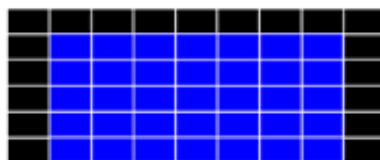


Sugestão: crie funções auxiliares para i) criar uma coluna; ii) devolver o valor máximo presente no vetor de inteiros.

2. criar um gráfico de colunas 2D verticais suavizando os contornos das colunas dando os mesmos argumentos da alínea anterior e o número de pixels do gradiente. O gradiente do contorno deve ser aplicado em direção ao interior da coluna.



a)



b)

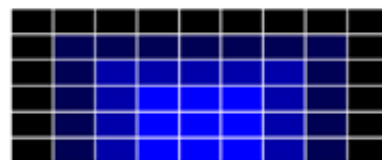
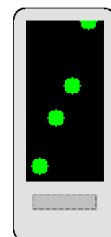


Figura 2 – a) coluna original com 7 pixels de largura; b) coluna com 7 pixels de largura e contornos suavizados (gradiente com 2 pixels).

3. criar um gráfico de dispersão (com um único valor por coordenada x) dando como argumentos:
  - o o vetor de inteiros (assumir que a posição do valor no vetor corresponde à sua coordenada x);
  - o o raio dos círculos;
  - o o espaçamento entre círculos;
  - o a cor dos círculos (a cor de fundo é preta).



4. obter uma imagem de um gráfico com rotação de 90° no sentido horário dando como argumento a imagem a cores representativa do gráfico.

## Parte 2: A classe de objetos Gráfico

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

Um objeto do tipo gráfico pode ser criado de duas maneiras: i) fornecendo apenas a imagem a cores; ii) fornecendo a imagem a cores, o título do gráfico, o título do eixo das abcissas e o título do eixo das ordenadas. Após a criação de um objeto do tipo gráfico, deve ser possível:

1. adicionar/modificar título do gráfico;
2. adicionar/modificar título do eixo abcissas;
3. adicionar/modificar título do eixo ordenadas;
4. definir o gráfico como transparente nos pixéis com coluna par e linha ímpar ou coluna ímpar e linha par;
5. obter imagem associada ao gráfico (nesta classe, os pixéis transparentes devem ser representados através da cor preta);
6. obter toda informação textual associadas a um gráfico (i.e. título do gráfico e dos eixos).

## Parte 3: A classe de objetos para sobrepor gráficos

**Objetivo:** Desenvolver uma classe de objetos que permita sobrepor gráficos.

Um objeto desta classe pode ser criado fornecendo um gráfico ou um vetor de gráficos. Após a criação de um objeto desta classe, deve ser possível:

1. armazenar uma pilha de gráficos (sem posições intermédias vazias no vetor que armazena os gráficos);
2. adicionar um gráfico no topo da pilha;
3. remover um gráfico do topo da pilha;
4. obter o gráfico presente no topo da pilha;
5. adicionar um gráfico numa dada posição (inferior à posição do elemento no topo da pilha);
6. trocar de posição dois gráficos da pilha;
7. obter todos os gráficos sem título;
8. obter os gráficos por ordem alfabética de título de gráfico (os gráficos sem título devem ficar no final do vetor);
9. obter a imagem resultante da sobreposição das imagens dos gráficos presentes na pilha (a imagem de um gráfico numa camada acima sobrepõe-se às que estão em camadas inferiores da pilha) – ver Figura 3;
10. obter a imagem resultante da sobreposição dos gráficos com rotação de 90º no sentido horário.

Sugestão: defina o número máximo de gráficos da pilha como constante.

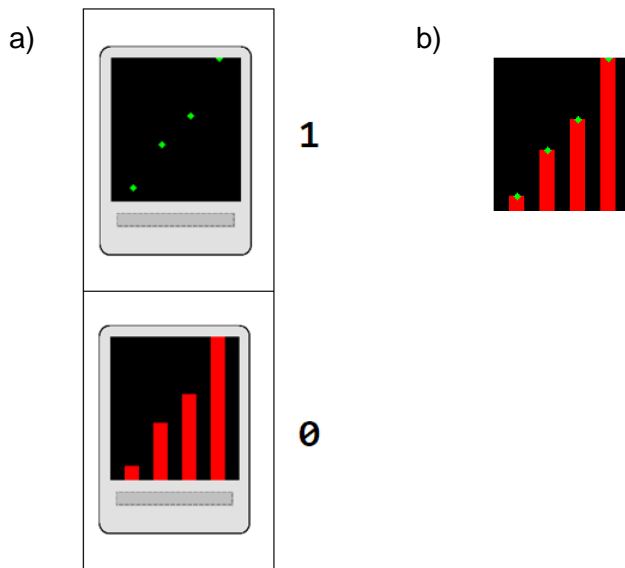


Figura 3 – a) pilha de gráficos; b) imagem resultante da sobreposição das imagens dos gráficos presentes na pilha.

## 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. A classificação no projeto não tem peso no cálculo da nota final. Contudo, a classificação no projeto define limites máximos para a mesma. 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 (<50%): 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, 30%;
- classe gráfico, 20%;
- classe sobreposição de gráficos 50%.

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 presencialmente em suporte eletrónico (ficheiros .java), diretamente ao professor da turma a que o aluno pertence.**