

Aluno: \_\_\_\_\_ Nota: \_\_\_\_\_

## **IIª Avaliação Individual – 2017.2**

### **Instruções (leia com atenção):**

- Controle o seu tempo. Ele faz parte da avaliação
- É permitida consulta exclusivamente a material físico e próprio.
- É vedado o empréstimo ou troca de qualquer material.
- É vedado o acesso a qualquer material digital ou acesso à Internet

Um sistema de avaliação de mapas militares funciona como se segue:

O sistema é composto por um atlas. Um atlas armazena uma coleção de mapas. Estes mapas são representações digitais de imagens do mundo real. Cada mapa é composto por uma coleção bidimensional de pixels. Como os mapas provêm de fontes distintas de captação, eles podem ser representados por sistemas de representação de cores distintos. Atualmente, o atlas pode armazenar dois tipos de mapas, os mapas RGB e os mapas CMYK, contudo estuda-se a possibilidade da incorporação de outros modelos de representação (mapas HSV, por exemplo). Os mapas RGB são compostos por coleções bidimensionais de pixels de cores no formato RGB, enquanto os mapas CMYK são compostos por coleções bidimensionais de pixels de cores no formato CMYK.

As cores RGB modelam um sistema de cores aditivas em que o Vermelho (Red), o Verde (Green) e o Azul (Blue) são combinados de várias formas de modo a reproduzir um largo espectro cromático. Uma cor RGB é representada por uma 3-upla, onde cada elemento pode assumir um valor de 0 a 255, representando a quantidade de vermelho, verde e azul que compõem a cor.

Já as cores CMYK modelam um sistema de cores subtrativas. Neste caso, cada cor é representada por uma 4-upla formado por ciano (Cyan), magenta (Magenta), amarelo (Yellow) e preto (Black (Key)). Cada um dos elementos desta tupla pode assumir um valor entre 0 e 100.

O sistema também opera com a noção de fragmentos. Um fragmento é um segmento bidimensional de pixels não estruturados (que pode ser compostos por pixels representados por cores de vários tipos. Isto ocorre porque um fragmento pode ser montado à partir da combinação de imagens oriundas de fontes distintas, (por exemplo, um drone e a câmara de um foguete). Assim, fragmentos podem ser em parte descritos em CMYK e em parte em RGB.

Pelo fato de operar com vários padrões de cores distintas, o sistema verifica a similaridade de duas cores quaisquer, através da distância (módulo da diferença) entre suas luminosidades (tom de cinza). Portanto, toda cor tem uma luminosidade associada independente do padrão. A luminosidade do padrão RGB é calculada através da expressão  $\text{luminosidade} = (R \cdot 0.3 + G \cdot 0.59 + B \cdot 0.11) / 255$ , que deve ser truncada para um valor inteiro. A luminosidade de uma cor CMYK é dado pela quantidade de preto (K) na cor.

- 1) (3.0) – Escreva um modelo de classes que descreva adequadamente as classes de negócio do problema acima.
- 2) (4.0) – Escreva todos os métodos necessários para que o sistema receba um fragmento e retorne o mapa que possua o trecho com maior nível de similaridade ao fragmento. Nesta operação, o sistema somente deve considerar mapas que tenham altura e largura maior que a do fragmento. Em seguida, o sistema deve verifi-

car qual o trecho do mapa é mais similar ao fragmento (através do cálculo da média da similaridade dos pixels). Este maior valor é considerado como a similaridade do fragmento ao mapa. Por fim, o sistema deve retornar o mapa que possui maior similaridade ao fragmento. A assinatura do método, na classe que representa o sistema, deve ser similar à exibida abaixo.

```
public Mapa getMapaMaisSimilar(Fragmento fragmento);
```

3) (3.0) – O sistema quer incorporar a possibilidade de converter mapas de um sistema de cor para outro sistema de cor. Por exemplo, converter um Mapa em RGB para um Mapa CMYK; de um mapa CMYK para um mapa RGB, etc. O conversor deve criar um mapa do tipo adequado (aquele que se quer converter) dada as dimensões do mapa original, e depois acrescentar na posição correta um novo pixel representado na cor destino da conversão. Um conversor foi escrito, com esta finalidade e o código é fornecido abaixo.

Para que o conversor seja genérico ele foi baseado em interfaces. Escreva o código da interface ConversorCor, e uma classe que possa ser utilizada pelo conversor para converter Mapas de CMYK para RGB considerando as seguintes equações de conversão

- $R = 255 \times (1-C) \times (1-K)$
- $G = 255 \times (1-M) \times (1-K)$
- $B = 255 \times (1-Y) \times (1-K)$

```
public class Conversor {
    public Mapa converter(Mapa mapa, ConversorCor conversor){
        Mapa novo = conversor.getNovoMapa(mapa.getAltura(), mapa.getLargura());
        for(int iCont = 0; iCont < mapa.getAltura(); iCont++){
            for(int jCont = 0; jCont < mapa.getLargura(); jCont++){
                Cor cor = mapa.getPixel(iCont, jCont);
                Cor novaCor = conversor.converter(cor);
                novo.setPixel(iCont, jCont, novaCor);
            }
        }
        return novo;
    }
}
```