

# Programação

Primeiro trabalho prático  
Semestre de Inverno de 2020/2021

Cada grupo terá que entregar até 23 de novembro, no respetivo *site* da turma, os ficheiros (extensão .kt) com o código do trabalho devidamente indentado e comentado.

O trabalho consiste em fazer um programa em *Kotlin* para desenhar um *smile*, tal como mostra a figura 1, que pode ser redimensionado, usando as teclas (+ e -), e deslocado usando as teclas (←, →, ↓ e ↑) ou com *clicks* do rato.

Inicialmente o *smile* está centrado no *canvas* e tem um raio com 50 pixels, tal como é apresentado na figura 1.

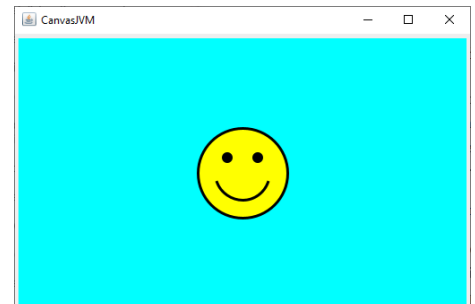


Figura 1 : Janela CanvasJVM

O *smile* é desenhado com as proporções indicadas na figura 2, em que todos os elementos são calculados em função do raio **R**. Por exemplo, a espessura da linha desenhada na circunferência exterior e no arco da boca é de **0,07 R**, que no caso de **R=50** é **3,5** que corresponde a **3** depois de convertido o valor para Int.

O *smile* deve ser representado por um valor do tipo agregado *Smile* definido como:  
`data class Smile(val x :Int, val y :Int, val radius :Int)`  
em que **x** e **y** é centro da figura e **radius** é o valor de **R**.

O programa só pode ter um ponto de mutabilidade, ou seja, só deve ter uma variável (var) do tipo *Smile* cujo valor é alterado por cada modificação da representação da figura.

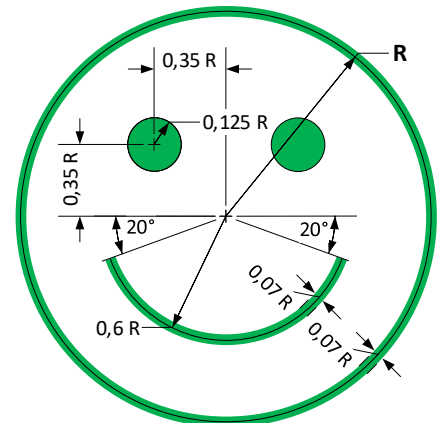


Figura 2: Proporções do desenho

Quando é premida a tecla (+) o raio é aumentado de 1 pixel e o *smile* é redesenhado com as novas proporções. Quando é premida a tecla (-) o raio é reduzido de 1 pixel e o *smile* também é redesenhado com as novas proporções, mas só no caso da espessura da linha ainda ficar com um valor maior ou igual a 1 pixel.

Quando for premida uma das teclas de cursor (←, →, ↓ ou ↑) o valor de x ou de y do centro do *smile* é aumentado ou reduzido em 4 pixels. Quando for dado um *click* com rato na área do *canvas* o centro do *smile* é alterado para a posição do *click*. Em ambos os casos o *smile* é redesenhado no novo centro.

Na realização do trabalho devem ser respeitadas as seguintes regras:

- Evitar mutabilidade. Este programa só necessita de um ponto de mutabilidade, já referido anteriormente.
- Não repetir código. Definir funções que são chamadas em diversos locais.
- Não fazer funções demasiado extensas. Decompor funções extensas em várias funções. Para ser compreensível, cada função dever ter menos do que 25 linhas.
- Não repetir valores com o mesmo significado nem usar “valores mágicos”, que não se percebe porque estão ali. Uma determinada característica só deve estar expressa num local e esse deve ser o local esperado. Caso seja necessário, definir valores nomeados, cujo nome é esclarecedor e será usado num ou vários locais do programa. Assim, quando for necessário alterar esse valor, só será necessário modificar essa definição.

Por exemplo, definir o valor de deslocamento provocado pelas teclas de cursor com a declaração global:

```
const val DELTA_CURSOR = 4
```

Uma implementação do programa pretendido está disponível no ficheiro [smile.jar](#). Para executar este programa, provavelmente basta mandar executá-lo no seu sistema operativo, mas, em alternativa, abra uma janela de comandos localizada onde reside o ficheiro e execute o comando: `kotlin smile.jar` ou então o comando: `java -jar smile.jar`

Bom trabalho.

ISEL, 4 de novembro de 2020