

Jogo MasterMind

Pretende-se desenvolver um programa para jogar ao [MasterMind](#) usando símbolos (letras ou dígitos) em vez dos pinos coloridos.

Inicialmente, o programa gera aleatoriamente um código secreto sem repetições, depois vai respondendo a cada tentativa introduzida pelo utilizador, indicando quantos símbolos estão corretos (C - cor certa na posição certa) e quantos estão trocados (T - cor certa na posição errada), até que o utilizador adivinhe o código ou esgote o máximo de tentativas.

Um exemplo de execução do programa é o apresentado à direita, para adivinhar um código com 4 cores das 6 possíveis, no máximo de 10 tentativas, em que as tentativas introduzidas pelo utilizador estão em **itálico**.

O programa só aceita tentativas válidas (número de símbolos correcto e só símbolos válidos).

Cada vez que o utilizador introduz uma tentativa válida, são apresentadas todas as tentativas já realizadas com as respectivas respostas.

Para implementar o jogo foram definidos os seguintes valores:

```
const val MAX_TRIES = 10           // in 5..20
const val SIZE_POSITIONS = 4       // in 2..6
const val SIZE_COLORS = 6          // in 2..10 and >= SIZE_POSITIONS
const val FIRST_COLOR = 'A'        // 'A' or 'a' or '0'
val COLORS = FIRST_COLOR ..< FIRST_COLOR+SIZE_COLORS
```

Modificando apenas estes valores, o programa poderá funcionar com mais ou menos tentativas, com mais ou menos posições e cores, ou com outro intervalo de símbolos para representar as cores.

A função **main** do programa poderá ser a apresentada a seguir, faltando implementar as funções **generateSecret**, **readGuess**, **getCorrects**, **getSwapped** e **printTry**.

```
fun main() {
    val secret: String = generateSecret()
    println("Descubra o código em $MAX_TRIES tentativas.")
    println("$SIZE_POSITIONS posições e $SIZE_COLORS cores $COLORS")
    for (numTries in 1..MAX_TRIES) {
        val guess = readGuess(numTries)
        if (guess == secret) {
            println("Parabéns!\nAcertou à ${numTries}ª tentativa.")
            return
        }
        val corrects = getCorrects(guess, secret)
        val swapped = getSwapped(guess, secret)
        printTry(numTries, guess, corrects, swapped)
    }
    println("Não acertou em $MAX_TRIES tentativas.")
}
```

```
Descubra o código em 10 tentativas.
4 posições e 6 cores A..F
1ª tentativa: ABCD
1ª : ABCD -> 0C + 2T
2ª tentativa: ISEL
Tentativa inválida.
2ª tentativa: BAEF
2ª : BAEF -> 1C + 2T
3ª tentativa: AACC
3ª : AACC -> 1C + 1T
4ª tentativa: CAFE
Parabéns!
Acertou à 4ª tentativa.
```

Os ficheiros fonte (*.kt) da solução devem ser entregues, por cada grupo, até ao final do dia 21 de Outubro na página do moodle da respetiva turma.

Bom trabalho,
Pedro Pereira