

Navigation

- ▼ Dashboard

 Site home

> Site pages

▼ My courses

> Coordenação - LEI (2021/22)

▼ Introdução à Programação (26722)

S1 (2021/22)

> Participants

 Badges

 Competencies

 Grades

> General

> Objectivos e Tópicos

> Horários

▼ Séries de Exercícios e Guiões de Laboratório

 Série 0

 Série 1

 Série 2

 Série 3

 Série 4

 Guião 1

 Guião 2

 Guião 3

 Guião 4

 Guião 5

 Guião 6

 **Guião 7**

> Aulas

> Arquiteturas de Sistemas Computacionais (26748) S1...

> Cálculo (13538) S1 (2021/22)

> Lógica de Primeira Ordem (13539) S1 (2021/22)

> CADI - Comissão de Alunos do Departamento de Infor...

> Elementos de Matemática I (13565) S1 (2020/21)

> Introdução às Probabilidades e Estatística (22701)...

> Produção de Documentos Técnicos (26759) S1 (2020/21)

> Programação II (LTI) (26757) S2 (2020/21)

More...

Guião 7

O objectivo desta aula é desenvolver programas recorrendo a

- **abstracções procedimentais**, que tornam o código mais fácil de escrever, entender e reutilizar
- **abstrações de dados** oferecida por tipos referência disponíveis na linguagem como *String*, *StringBuilder* e *Color*

Assim, nos exercícios que propomos de seguida, o programa a desenvolver deve ter a seguinte forma:

```
public class NomeDaClasse {

    public static void main (String[] args) {
        //corpo do main
    }

    public static ... methodName1( ... ) {
        //corpo do método
    }

    public static ... methodName2( ... ) {
        //corpo do método
    }
}
```

Para cada método que definir deve escrever um cabeçalho *javadoc* incluindo uma descrição sucinta e geral do método e, sempre que apropriado, @param, @requires, @return e

Exercício 1

Descarregue o ficheiro [Triangles.java](#) e guarde-o.

1. Modifique o programa de forma a que a altura do triângulo passe a ser pedida ao utilizador. No processo de leitura deve validar que o valor introduzido para a altura é um número inteiro positivo. No caso de o valor ser inválido, deve informar o utilizador e repetir a leitura, como no seguinte exemplo:

```
Insira a altura do triangulo: -2
A altura do triangulo deve de ser um numero inteiro positivo.
Insira a altura do triangulo: 4
*
***
*****
*****
*****
```

Sugestão: Recorra à classe *Scanner*, incluída no pacote *java.util*, cuja documentação pode consultar [aqui](#).

2. Modifique novamente o programa de modo a que este também leia o símbolo a ser utilizado na representação do triângulo.

3. Modifique novamente o programa, substituindo o procedimento *void printSequence(int n, char c)* por uma função *generateSequence(int n, char c)* que devolve um objecto do tipo *String* (com a sequência de *n* cópias de *c*). Repare que esta nova forma de estruturar o código é mais flexível, já que a criação da sequência e o que se deve de fazer com ela (imprimir no ecrã) estão separadas.

Sugestão: Recorra à classe *StringBuilder*, incluída no pacote *java.lang*, cuja documentação pode consultar [aqui](#).

Exercício 2

Escreva uma função *boolean passwordValida(String s)* que verifica se a *string* dada é uma palavra chave válida de acordo com as seguintes regras:

- tem no mínimo 8 caracteres;
- contem apenas letras e algarismos;
- tem pelo menos dois algarismos.

Sugestão: Recorra à classe *Character*, incluída no pacote *java.lang*, cuja documentação pode consultar [aqui](#).

Exercício 3

Descarregue os ficheiros [Painting.java](#) e [StdDraw.class](#) e guarde-os na mesma pasta.

- Crie uma função *void paintMystery(int dim, Color c, int centerX, int centerY)* que itera em todos os pontos (*x*, *y*) onde $0 \leq x < dim$ e $0 \leq y < dim$ e verifica se a condição *isMystery(x, y, centerX, centerY)* é satisfeita. Em caso afirmativo, pinte um pixel a cor *c* na posição (*x*, *y*) usando a primitiva fornecida *void paintPixel(Color c, int x, int y)*.
- Utilize a função *paintMistery* com argumentos *dim* = 128, *centerX* = 64, *centerY* = 64 e uma cor à sua escolha.
- Crie uma função *boolean isInsideCircle(int x, int y, int radius, int centerX, int centerY)* que retorna true se o ponto (*x*, *y*) está dentro de um círculo centrado em (*centerX*, *centerY*) com raio *radius* e *false* caso contrário.
- Construa uma função semelhante a *paintMystery* que desenha um círculo, utilizando a função *isInsideCircle* que definiu anteriormente e chame-a na *main*.

Sugestão: Recorra à classe *Color*, incluída no pacote *java.awt*, cuja documentação pode consultar [aqui](#).

Exercício 4

Escreva um programa que gere aleatoriamente uma matrícula de automóvel portuguesa. Considere apenas as novas matrículas que estão atualmente a ser criadas, isto é, as que têm números no meio e sem separadores (e.g. AA22BB).


Sugestão: para gerar um carácter aleatório, prepare uma *String* com o conteúdo “ABCDEFGHJKLMNOPQRSTUVWXYZ”, e invoque o método *charAt* utilizando como argumento um número inteiro aleatório adequado. Para isso, utilize a classe *Random* do pacote *java.util*, cuja documentação pode consultar [aqui](#).

Exercício 5

Escreva uma função *String representacaoBinario(int n)* que, assumindo que *n* é um número positivo, devolve uma *String* com a representação de *n* em binário.

Sugestão: Recorra à classe *StringBuilder*, incluída no pacote *java.lang*, cuja documentação pode consultar [aqui](#).

Last modified: Wednesday, 11 November 2020, 2:28 PM



PREVIOUS ACTIVITY
Guião 6

Jump to... 