

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

 Guião 1

 Guião 2

 Guião 3

 Guião 4

 Guião 5

 **Guião 6**

> 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 6

O objectivo desta aula é desenvolver programas recorrendo a abstrações procedimentais, que tornam o código mais fácil de escrever, entender e reutilizar. Assim, nos exercícios que lhe 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 @ensures.

Exercício 1

1. Faça download do ficheiro `Triangles.java`, analise a classe `Triangles` e teste a
3. Junte à classe `Triangles` um novo procedimento que imprima no ecrã um triângulo rectângulo com uma dada altura, como o mostrado abaixo (neste caso, com altura 5).

```
 *
 **
 ***
 ****
 *****
```

4. Certamente não se esqueceu de, adicionalmente ao código, escrever a especificação do novo método em javadoc. Gere a documentação correspondente utilizando os seguintes comandos:

```
Primeiro crie uma pasta onde vamos gerar a documentação e depois gere a documentação nessa pasta.

$ mkdir TrianglesDocs

$ javadoc -d TrianglesDocs -tag requires:m:"Requires:" -tag ensures:m:"Ensures:" Triangles.java
```

Use o browser para ver o que é gerado no ficheiro `Triangles.html` .

Exercício 2

O ISBN é um número com 10 dígitos que identifica um livro de forma única. O dígito mais à direita é um *checksum digit* que pode ser univocamente determinado pelos restantes 9 dígitos, utilizando a seguinte condição:

$$10.d_{10}+9.d_9+\dots+2.d_2+1.d_1 \text{ tem de ser um múltiplo de } 11$$

onde d_i representa o i -ésimo dígito do número, a contar da direita. Na condição acima, d_i representa o *checksum digit*. O *checksum digit* pode ser qualquer valor entre 0 e 10. A convenção do ISBN é usar o carácter ‘X’ para representar o 10.

Programa uma classe `IsbnGenerator` que como se mostra no exemplo abaixo, recebe um inteiro com 9 dígitos como argumento na linha de comando e imprime o ISBN válido com o *checksum digit*:

```
$ java IsbnGenerator 020131452

ISBN: 0201314525
```

Recorra à definição de funções e/ou procedimentos para que o seu código fique mais fácil de entender, modificar e de reutilizar.

Exercício 3

Programa uma classe `SquareRoots` com várias funções para calcular raízes, nomeadamente:

1. Uma função que calcula a raiz inteira de um número positivo n (do tipo `int`).
2. Uma função que calcula a aproximação da raiz real de um número positivo n (do tipo `int`) recorrendo ao *Algoritmo Guess&Check*, com uma margem de erro inferior a um valor de erro `epsilon` fornecido (`epsilon>0`). Teste a sua função com diferentes valores de n e `epsilon`.
3. Uma função que calcula a aproximação da raiz real de um número positivo n (do tipo **int**) recorrendo ao *Algoritmo de Newton*, com uma margem de erro inferior a um valor de erro `epsilon` fornecido (com `epsilon>0`).
4. Inclua em cada função um contador que conte o número de iterações executadas por cada algoritmo e imprima esse valor.

Algoritmo `Guess&Check`:

- Calcula a raiz inteira de n , i.e., o maior número inteiro x cujo quadrado não ultrapassa n .
- Considera o intervalo $[min,max]$ com $min=x$ e $max=x+1$.
- Verifica se min é uma aproximação suficientemente boa da raiz de n (i.e., se $|min-n/min|<epsilon*min$). Se sim, está encontrado o valor.
- Senão, verifica se $((max+min)/2)^2 < n$. Em caso positivo repetir o processo agora para o intervalo $[(max+min)/2,max]$ e, caso contrário, com o intervalo $[min,(max+min)/2]$.

Algoritmo de Newton:

- Considera a sucessão definida por: $t_0=n$, $t_{i+1}=1/2 * (t_i+n/t_i)$.
- Calcula os elementos da sucessão t_0 , t_1 , $t_2\ldots$ enquanto $|t_i -n/t_i |> epsilon*t_i$.
- Para quando se encontra o primeiro elemento que não satisfaz esta condição.

Last modified: Friday, 5 November 2021, 6:20 PM



PREVIOUS ACTIVITY

Guião 5

Jump to...

