

Aula Prática 1

Objectivos

- Ambiente de programação Eclipse (IDE).
- Usar construtores de Classes.
- Utilizar mecanismos de Encapsulamento (*Information Hiding*) e Visibilidade
- Criar métodos sobrepostos (Overloading).

Nota Introdutória

A parte mais importante de uma classe é a sua interface pública. Por esta razão os nomes dos métodos públicos – e também a restante assinatura: nomes e tipos de argumentos caso existam – devem ser simples e fáceis de compreender. As abreviaturas poderão ser utilizadas se, no contexto da classe, o seu significado for perfeitamente claro.

Problema 1.1

Crie um projecto (p3) e um package (aula1) no Eclipse que implemente um programa em *Java*. Esse programa deve ler uma String do teclado e apresentar os seguintes resultados:

- Total de caracteres numéricos (0..9).
- Indicar se só tem minúsculas.
- Indicar se só tem maiúsculas.
- Total de palavras lidas, imprimindo-as no ecrã.
- Imprimir uma nova String na qual todos os caracteres foram trocados 2 a 2 (ex: "comando" → "ocamdno").

Problema 1.2

Faça um programa que, de uma forma interactiva (recorrendo a um menu), permita gerir uma lista de pessoas. Cada pessoa é caracterizada pelo nome, número do cartão do cidadão e data de nascimento. Numa primeira versão o programa deve permitir a introdução de novas pessoas na lista, apagar pessoas existentes e apresentar a lista completa (com uma formatação visual minimamente aceitável).

- Divida o programa em módulos de forma a potenciar a sua reutilização futura. Cada módulo deve encapsular os seus dados de forma a ocultar os aspectos de implementação e proteger os dados do exterior. Tenha especial atenção aos aspectos de robustez da aplicação como, por exemplo, a validação das datas.

```
public class Pessoa {
    String nome;
    int cc;
    Data dataNasc;

    // .....
}

public class Data{
    int dia;
    int mes;
    int ano;

    // .....
}
```

- b) Faça uso de modificadores de acesso para garantir que todos os atributos das classes não estão acessíveis do exterior. Em caso de necessidade, defina novos métodos a incluir na interface pública da classe.
- c) Crie métodos adequados para permitir a inicialização dos seus atributos no momento de criação de cada objecto.

```
Pessoa p = new Pessoa("António Nunes", 98012244, dNasc);
Data d = new Data(11, 4, 1974);
```
- d) Acrescente a possibilidade de ordenar a lista por nome, ou por número de CC.

Problema 1.3

Faça um programa que permita calcular a **área** e o **perímetro** de qualquer uma das seguintes figuras geométricas: **Quadrado, Rectângulo e Circulo**.

O programa deve permitir a introdução dos elementos necessários à caracterização de cada uma das figuras apresentando de seguida os respectivos valores da área e perímetro.

- a) Crie uma classe própria para suportar cada uma das figuras.
- b) Crie uma classe Ponto para suportar pontos bidimensionais (abscissa e ordenada) e respectivas operações. Utilize, por exemplo, esta classe como componente da classe Circulo. Tenha em atenção os conceitos de encapsulamento e visibilidade.

```
public class Circulo {
    private double raio;
    private Ponto centro;

    // .....
}

public class Ponto {
    private double x;
    private double y;

    // .....
}
```

- c) Implemente os seguintes construtores:

```
public Circulo(double x, double y, double r) {...}
public Circulo(Ponto centro, double r) {...}
public Ponto(double x, double y) {...}
```

- d) Acrescente o método "public String toString()" que retorne uma descrição de todos os elementos de cada objecto.
- e) Implemente também os seguintes métodos:
 - a. Verifique se dois Círculos são iguais;
 - b. Verifique se dois Círculos se interceptam;