

Parte 1

Exercício 1

1.1 Escreva uma classe `User` que represente os seguintes dados de uma pessoa:

- **id** - Identificação da pessoa
- **name** - Nome de uma pessoa
- **email** - Email de uma pessoa

De seguida, teste a classe através de uma outra que implemente o método `main()`.

Resolução:

1. Num projecto cujo nome seja “*PP_FP04*”, crie uma classe (Java Class...) com o nome: `User`, e declare os atributos necessários para representar a informação base de uma pessoa:

```
package pkg2016.pp.fp04;

/**
 * <h3>
 * ESTGF – Escola Superior de Tecnologia e Gestão de Felgueiras<br>
 * IPP – Instituto Politécnico do Porto <br>
 * LEI – Licenciatura em Engenharia Informática<br>
 * PP – Paradigmas de Programação / Programming Paradigms<br>
 * </h3>
 * <p>
 * <strong>Description: </strong><br>
 * Class that represents the user structure
 * </p>
 */
public class User {
    private static final int ID_SIZE = 3;
    protected static char[] id = new char[ID_SIZE];
    protected static char[] name;
    protected static char[] email;
    protected static Expenses expenses;
}
```

Repare que o atributo `ID_SIZE` possui o modificador `private` (apenas visível dentro da classe), enquanto que os restantes atributos possuem o modificador `protected` (visível na classe e no package¹). Note ainda que todos os atributos são membros de classe (modificador `static`).

¹ Os atributos com o modificador `protected` são também visíveis noutros `packages` em classes que derivem da classe à qual pertence.

2. No método `main`, escreva o código necessário para inicializar os valores dos atributos da classe `User`.

```
public class PPFP04 {  
  
    static User user1;  
    public static void main(String[] args) {  
        user1.id[0] = 'A';  
        user1.id[1] = 'B';  
        user1.id[2] = 'C';  
        user1.name = new char[]{'B', 'r', 'u', 'n', 'o'};  
        user1.email = new char[]{'b', 'm', 'o', '@', 'e', 's', 't', 'g',  
            'f', '.', 'i', 'p', 'p', '.', 'p', 't'};  
    }  
}
```

3. Imprima o valor dos atributos:

```
System.out.println("-----User 1-----");  
System.out.print("ID:");  
System.out.println(user1.id);  
System.out.print("Name:");  
System.out.print(user1.name);  
System.out.println("Email:");  
System.out.println(user1.email);
```

1.2 Declare um segundo utilizador inicializando-o com valores à sua escolha. Imprima o seu conteúdo. Interprete o resultado associando-o aos vários avisos (*warnings*) apresentados pelo IDE.

- 1.3 Considere agora a seguinte versão do método `main` para resolver o exercício da alínea 1.1:

```
public static void main(String[] args) {  
    User user1=null;  
    user1.id[0] = 'A';  
    user1.id[1] = 'B';  
    user1.id[2] = 'C';  
    user1.name = new char[]{'B', 'r', 'u', 'n', 'o'};  
    user1.email = new char[]{'b', 'm', 'o', '@', 'e', 's', 't', 'g',  
        'f', '.', 'i', 'p', 'p', '.', 'p', 't'};  
  
    System.out.println("-----User 1-----");  
    System.out.print("ID:");  
    System.out.println(user1.id);  
    System.out.print("Name:");  
    System.out.println(user1.name);  
    System.out.println("Email:");  
    System.out.println(user1.email);  
}
```

Existe alguma diferença no resultado da sua execução em relação à versão anterior? Identifique as diferenças e o seu impacto nesta nova versão.

Exercício 2

2.1. Crie uma classe `Expenses` que permita armazenar dados relativos a dois tipos de despesas (despesas com o automóvel e despesas alimentares) do utilizador (criado no exercício anterior) durante um mês. Cada despesa deverá possuir os seguintes atributos:

- **number**: número da despesa;
- **value**: valor em euros da despesa num determinado dia do mês, associando cada dia a um índice do vetor.
- **description**: descrição sucinta da despesa;

Resolução parcial:

1. Estrutura da classe `Expenses`:

```
/**
 * <h3>
 * ESTGF - Escola Superior de Tecnologia e Gestão de Felgueiras<br>
 * IPP - Instituto Politécnico do Porto <br>
 * LEI - Licenciatura em Engenharia Informática<br>
 * PP - Paradigmas de Programação / Programming Paradigms<br>
 * </h3>
 * <p>
 * <strong>Descrição: </strong><br>
 * Class that represents the expense structure
 * </p>
 */
public class Expenses {

    private static final int NUMBER_OF_DAYS = 31; //para o mês de março

    //Para despesas com o automóvel
    protected static int number1;
    /**
     * Vetor com o valor das despesas de automóvel durante o dia do mês
     * (o índice do vetor corresponde ao dia do mês)
     */
    protected static double[] carValues = new double[NUMBER_OF_DAYS]; //em Euros
    protected static char[] descriptionCar = new char[]{'C', 'A', 'R'};

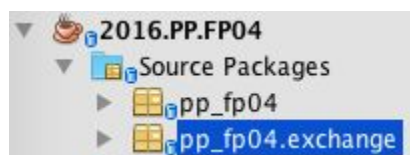
    //Para despesas alimentares
    protected static int number2;
    protected static double[] foodValues = new double[NUMBER_OF_DAYS]; //em Euros
    protected static char[] descriptionFood = new char[]{'F', 'O', 'O', 'D'};
}
```

2.2. Associe as despesas ao utilizador criado no exercício 1. De seguida e utilizando uma outra classe com um método `main`, inicialize e imprima os valores do utilizador e as suas despesas.

2.3. Apresente o total gasto em despesas de cada tipo durante o mês, assim como o valor total e valor médio de despesas gasto durante o mês pelo utilizador.

Exercício 3

3.1 Crie uma classe `CurrencyRates` num novo package com o nome: `pp_fp03.exchange`. **Atenção que terá de continuar a existir o package `pp_fp03`:**



Na classe criada, defina os valores da taxa de câmbio de acordo com a seguinte tabela:

Moeda	Dólar	Euro	Iene	GBP
Dólar	1.00000	0.76900	96.1141	0.67131
Euro	1.30030	1.00000	124.989	0.87290
Iene	96.06148	125.00000	1.00000	143.26648

3.2. Recorrendo à classe `CurrencyRates` que criou, calcule o valor total e valor médio de despesas gasto no mês pelo utilizador em dólares, ienes e libras.

Coloque os atributos da classe `CurrencyRates` como `protected`. O que aconteceu? Porque é que na classe `main` dá um erro relativo à classe `CurrencyRates`?

Parte 2

Exercício 1

Num *package* com o nome: `pp_fp03.book`, crie uma classe `Book` que permita armazenar os dados de um livro. Os dados a armazenar são os seguintes:

- Título Original;
- Editor;
- Dados relativos a dois autores, criando para isso uma nova classe `Authors` que permita o armazenamento dos seguintes dados para dois autores (os dados de teste ficam ao seu critério):
 - Acrônimo do autor (três caracteres);
 - Nome do autor;
 - Nacionalidade do autor.
- Ano;
- Páginas;
- Cotação (0-9) dada pelos 3 principais críticos de cinema.

De seguida crie uma classe `BookDemo` para testar a classe `Book`, imprimindo os valores dos seus atributos..

Gere o Javadoc para o projecto utilizado na resolução desta ficha de trabalho.