

Aula Prática 13

Objectivos

Resolução de problemas gerais de programação

~~Problema 13.1~~

Considere as seguintes entidades:

- ~~País: caracterizado por um nome (String), uma capital (Localidade) e por um conjunto pré-definido de regiões (Região).~~
- ~~Região: caracterizada por um nome (String) e uma população (int)~~
- ~~Estado: caracterizada por um nome (String), uma população (int) e uma capital (Localidade do tipo TipoLocalidade.CIDADE)~~
- ~~Província: caracterizada por um nome (String), uma população (int) e um governador (String)~~
- ~~Localidade: caracterizada por um nome (String), uma população (int) e um tipo (TipoLocalidade.CIDADE, TipoLocalidade.VILA, TipoLocalidade.ALDEIA)~~
- ~~Organização: conjunto ilimitado de países (vector dinâmico). Os países não poderão estar repetidos. Deve-se possível adicional (add), remover (remove) e verificar países (exists).~~

Nota: Deve construir todas as estruturas de dados necessárias – genéricas - sem usar java collections.

Construa um programa que represente adequadamente estas entidades.

Crie construtores, ~~os métodos set/get que lhe pareçam adequados para cada atributo~~, bem como qualquer outro método que seja fundamental para o bom funcionamento do programa. Inclua mecanismos de **controlo de exceções em todos os pontos do programa em que possam surgir situações imprevistas.**

Teste as classes desenvolvidas com a seguinte função main (pode usar outros nomes para as estruturas de dados):

```
public static void main(String argv[]) {
    Localidade cid1 = new Localidade("Sacramento", 5125, TipoLocalidade.CIDADE);
    Localidade cid2 = new Localidade ("Vila Rija", 9345, TipoLocalidade.CIDADE);
    Estado est1 = new Estado("Euro Norte", 23133, cid1);
    Estado est2 = new Estado("Euro Sul", 43321, cid2);
    Provincia prov1 = new Provincia("Centro", 232475, "Manuel Pereira");
    Provincia prov2 = new Provincia("Litoral ", 232475, "Maria Vilar");
    Provincia prov3 = new Provincia("Afro Poente", 287349, "Mané Lima");

    MyList<Regiao> reg1 = new MyList<Regiao>();
    reg1.add(est1);
    reg1.add(est2);
    MyList<Regiao> reg2 = new MyList<Regiao>();
    reg2.add(prov1);
    reg2.add(prov2);
    reg2.add(prov3);
    Pais p1 = new Pais("Eurais", est1.getCapital(), reg1);
}
```

```

Pais p2 = new Pais("United XPTO", cid2, reg2);

System.out.println(p1);
Organizacao org = new Organizacao("Nações Reunidas");
org.add(p1); org.add(p2);

Iterator itr = org.iterator();
while ( itr.hasNext() )
    System.out.println( itr.next() );

if (!org.add(p1))
    System.out.println("País já pertence à Organização.");
org.remove(p2);
if (!org.remove(p2))
    System.out.println("Pais não pertence à Organização.");
}

```

Problema 13.2

(Usando java collections..)

Construa um programa que leia um ficheiro de texto e que conte todos os pares de palavras encontrados no ficheiro e o número de ocorrências de cada par. Despreze todas as palavras de tamanho inferior a 3 e considere como separadores os seguintes caracteres:

\t\n.,'";?!*{}=+&/()[]""'\"'

O resultado deverá ser armazenado por ordem alfabética num ficheiro “output.txt”. Se utilizar a primeira frase deste problema o resultado no ficheiro de saída deverá ser o seguinte:

```

construa={programa=1}
conte={todos=1}
ficheiro={texto=1}
leia={ficheiro=1}
palavras={encontrados=1}
pares={palavras=1}
programa={que=1}
que={conte=1, leia=1}
texto={que=1}
todos={pares=1}

```

Para cada palavra encontrada no ficheiro o programa deverá associar um conjunto de todas as palavras seguintes contando igualmente quantas vezes cada par ocorre.

Teste o programa com o ficheiro “Policarpo.txt”. O resultado deverá ser semelhante a:

```

1864={tratava=1}
abacateiros={entoando=1, mangueiras=1, oitenta=1, suas=1, troncos=1}
abacates={ora=1, pouco=1}
abacaxis={coroados=1, que=1}
abafada={comprimida=1, tens=1}
...
voltava={aos=1, biblioteca=1, com=1, ficava=1, idéia=1, olhava=1, para=1, seu=1}
voltavam={mesmo=1, oficial=1, para=1, sorridentes=1}
voltou={acidente=1, agarrou=1, alegria=1, aos=1, bonde=1, general=1, instante=1,
lhe=1}

```