Aula Prática 8

Resumo:

- Listas e vectores ordenados.
- Recursão e iteração.

Exercício 8.1

A função seguinte calcula a soma de um subarray de números reais:

```
// sum of subarray [start,end[ of arr:
static double sum(double[] arr, int start, int end) {
  assert 0 <= start && start <= end && end <= arr.length;
  double res = 0;
  for(int i = start; i < end; i++)
    res += arr[i];
  return res;
}
```

Implemente uma versão recursiva – `sumRec` – desta função. Para a testar, complete o programa `SumArgs` para fazer a soma de todos os seus argumentos.

Exercício 8.2

No material da aula, encontra a classe `SortedListInt`, idêntica à desenvolvida na aula teórica. Usando esta classe, faça um programa `SortInts` que leia números inteiros de um ou mais ficheiros e, no final, imprima todos os números lidos por ordem. Os nomes de ficheiros serão passados como argumentos ao programa. O ficheiro poderá conter outras palavras: as que não representem inteiros devem ser ignoradas.

Exercício 8.3

Faça as adaptações necessárias para transformar a classe `SortedListInt` numa classe genérica `SortedList` que deverá colocar no pacote `p2utils`. Na declaração da classe, deve especificar que os elementos do tipo `E` são comparáveis:

```
public class SortedList<E extends Comparable<E>> {...}
```

Na implementação, terá que usar a função `compareTo()` para comparar elementos.

O programa `TestSortedList` demonstra a utilização da nova classe. Experimente correr `java -ea TestSortedList 9 3.14 24 10`. Altere o programa para aceitar também

argumentos não numéricos e mostrá-los ordenados lexicograficamente numa lista separada. Experimente `java -ea TestSortedList 9 dois pi 24 dez 3`. (Compare com o programa `.jar` fornecido.)

Exercício 8.4

Acrescente progressivamente os métodos necessários à classe `SortedList` para conseguir compilar e executar o programa `TestSortedList2`.

- O método `lst.contains(e)` verifica se um elemento `e` existe na lista `lst`. Como a lista está ordenada, pode implementar uma solução mais eficiente do que a desenvolvida para a classe `LinkedList` da aula anterior. Comece por criar e testar uma solução iterativa. Em seguida, desenvolva e teste uma solução recursiva.
- O método `lst.toString()` devolve uma cadeia de caracteres que representa o conteúdo da lista, num formato semelhante a `"[1, 2, 3]"`. Desenvolva uma solução iterativa.
- O método `lst1.merge(lst2)` devolve uma nova lista ordenada contendo os elementos das listas `lst1` e `lst2`, mas sem as alterar. Desenvolva uma solução recursiva.

Exercício 8.5

A classe `SortedList` genérica permite, por exemplo, manter uma lista ordenada dos aniversários dos seus familiares e amigos. Para isso:

- Desenvolva uma classe `Pessoa` com as seguintes funcionalidades: construtor tendo como parâmetros o dia de nascimento (uma `Data`) e o nome da pessoa (`String`); métodos de acesso aos campos data de nascimento e nome; método `toString()`, que devolve uma representação da pessoa em cadeia de caracteres.
- Adicione um método de comparação que deve comparar apenas o dia e mês de nascimento. `p1.compareTo(p2)` deve devolver um inteiro negativo, positivo ou zero, consoante a pessoa `p1` faça anos antes, depois ou no mesmo dia que `p2`, respetivamente. Finalmente, para que `Pessoa` seja reconhecido como um tipo comparável, e portanto aceitável como argumento numa `SortedList`, a declaração da classe terá que ser: `public class Pessoa implements Comparable<Pessoa> { ... }`
- Complete o programa `Birthdays` que obtém os dados de algumas pessoas pelos argumentos e os apresenta ordenados por data de aniversário. Utilize uma lista ordenada de pessoas para resolver o problema.

Exercício 8.6

Acrescente uma nova classe `SortedArray` ao pacote `p2utils`. Esta classe deverá ter funcionalidade semelhante à da `SortedList`, mas deverá ser implementada com base num vector de dimensão fixa. A dimensão será dada como argumento do construtor. Como o vector tem capacidade limitada, a classe deverá ter um método `isFull()`, que devolve `true` se o vector estiver cheio e `false` caso contrário. Use o programa `TestSortedArray2` para testar.