

Laboratórios de Informática III (LI3)

LEI – 2º ano – 2º semestre – Universidade do Minho – 2013/2014

F. Mário Martins, João Luís Sobral

Abril de 2014

Projecto de Java: Rede de Autores de artigos científicos

1.- Introdução.

O projecto de Java da disciplina de LI3 tem por objectivo fundamental ajudar à consolidação experimental dos conhecimentos teóricos e práticos adquiridos na disciplina de Programação Orientada pelos Objectos.

Não será solicitada aos alunos a aprendizagem de novas construções da linguagem Java, procurando-se até capitalizar o mais possível do trabalho e dos resultados obtidos no projecto de C, e na utilização de parte desses conhecimentos para a criação de uma estruturação de dados em Java baseada na utilização das colecções e interfaces de JCF (“Java Collections Framework”), que permita a inserção e a realização de consultas interactivas de informações relativas a autores de artigos científicos publicados no DBLP (tema de base do projecto de C).

2.- Pré-Requisitos.

O projecto de Java tem como pressuposto a existência de um ficheiro de artigos científicos `publicx.txt`, já usado no projecto C, no qual cada linha representa os autores de uma dada publicação científica apresentada numa conferência ou numa revista no ano indicado.

Apresentam-se em seguida alguns exemplos destas linhas:

Kim-Sang, Hu Xiang, Alan C. Carter, 2001
John Kalmus, Gustavo Paz, João C. Lopes, 1991
K. Dix, Joana Sousa, Rui Sá, 2013

3.- Requisitos do programa a desenvolver.

Pretende-se desenvolver um programa em Java que seja capaz de, antes de mais, ler e armazenar numa estrutura de dados em memória tais informações, para que sobre a mesma possam ser realizadas diversas consultas (“queries”), algumas estatísticas e alguns testes de “performance”.

Tomando por referência o trabalho anterior em C, em que se tornou necessário criar um **Catálogo de Autores** global, ou seja, contendo todos os autores de todos os

artigos, **pretende-se agora criar uma estrutura, que designaremos uniformemente por Rede Global de Autores, em que, para cada ano para o qual existam publicações, se pretende associar uma Rede Anual de Autores.** Esta **Rede Anual de Autores** deverá representar, **para cada autor que publicou nesse ano**, o número total de artigos que o autor publicou a solo (sem co-autores), o número total de artigos com outros autores e ainda, todos os nomes dos seus co-autores desse ano e quantos artigos publicou com cada um deles.

O desenho e a implementação do código do programa deverão ter em atenção que o mesmo deverá ter dois grandes grupos de funcionalidades (correspondentes a 80% do valor do trabalho):

- a) Leitura de dados de memória secundária e população das estruturas de dados em memória central; Gravação da estrutura de dados em memória em ficheiros de objectos;
- b) Queries: operações de consulta sobre a estrutura de dados;

Complementarmente, equivalendo a 20% do valor do trabalho, pretende-se que seja desenvolvido um conjunto de módulos (ou de pequenos programas) independentes da aplicação anterior que realizem a funcionalidade designada por:

- c) Medidas de performance do código e das estruturas de dados.

3.1.- Leitura, população das estruturas e persistência de dados.

O programa deverá poder ler o ficheiro **publicx.txt** a qualquer momento e carregar os respectivos dados para memória. Na primeira execução do programa a realização desta operação é obrigatória.

A qualquer momento deverá estar disponível uma opção que permita ao utilizador gravar toda a estrutura de dados de forma persistente usando **ObjectStreams**, criando o ficheiro **publicx.obj**.

A qualquer momento também deverá o utilizador poder carregar os dados a partir de uma **ObjectStream** de nome dado, repopulando assim toda a informação da estrutura de dados até então existente em memória.

O arranque da aplicação deve também poder fazer-se inicializando a estrutura de dados a partir de uma **ObjectStream** sobre o ficheiro **publicx.obj**.

O programa deverá também poder ler um outro qualquer ficheiro de texto do formato indicado mas cujo nome é dado pelo utilizador.

3.2.- Estatística e queries interactivas.

Sendo inúmeras as possíveis informações a extrair da estrutura de dados, elas deverão ser agrupadas para o utilizador da seguinte forma:

3.2.1.- Consultas estatísticas.

1.1.- Apresenta ao utilizador os dados referentes ao último ficheiro de texto lido, designadamente, nome do ficheiro, número total de artigos, número total de nomes lidos, número total de nomes distintos e intervalo fechado dos anos dos artigos lidos (cf. Anos = [1971 a 2012]);

1.2.- Apresenta ao utilizador os números gerais respeitantes aos dados actuais na estrutura, designadamente: número total de autores, número total de artigos de um único autor, número total de autores que apenas publicaram a solo (sem co-autores), número total de autores que nunca publicaram a solo e número total de autores que publicaram, no intervalo de anos do ficheiro, mais de X artigos, sendo o valor X dado pelo utilizador;

1.3.- Apresenta uma tabela com o número total de publicações registadas em cada ano a considerar, por ordem crescente de ano.

3.2.2.- Consultas interactivas.

2.1.- Consultas indexadas por ano ou anos.

Dado um intervalo fechado de anos válido (os limites podem ser iguais, cf. [1999,1999]), mas os anos devem ser válidos, estas consultas deverão determinar:

- Os nomes (por ordem alfabética crescente) dos X autores que mais artigos publicaram**, sendo X dado pelo utilizador;
- Os X pares de autores (e respectivo número de artigos) com mais artigos publicados em co-autoria**, sendo X dado pelo utilizador (ordem decrescente do número de artigos publicados);
- A listagem, por ordem alfabética crescente, de todos os **co-autores comuns** aos autores de uma dada lista de nomes a introduzir pelo utilizador (lista limitada a, no máximo 3 autores);
- A listagem dos **nomes dos autores que publicaram artigos em todos os anos** do intervalo de anos dado (não fazer para um único ano, mas apenas para intervalos > 1).

2.2.- Consultas globais especiais.

- a) Contar o número de linhas em duplicado no ficheiro **publicx.txt** (o ficheiro deverá ser relido no momento da realização desta consulta);
- b) Total e lista dos nomes dos autores começados por uma dada letra de A a Z;
- c) Dado um qualquer nome de autor e um ano, determinar todos os seus co-autores e o somatório dos artigos publicados por tal autor no total das suas parcerias (cf. (A, B, 2) + (A, C, 3) => (A -> (B, C), 5)), ou seja, se o autor é A e A publicou com B 2 artigos, e com C 3 artigos, então A com os parceiros (B, C) publicou no total 5 artigos.
- d) Determinar o conjunto dos nomes de todos os co-autores de um dado autor, por ordem alfabética.

A criação das consultas deve ser realizada de forma suficientemente estruturada de modo a que se torne simples alterar o identificador e texto da mesma no menu de consultas e invocar o método associado respectivo.

Todas as *queries* realizadas devem, antes mesmo da apresentação dos resultados, e qualquer que seja a forma como os resultados finais são apresentados, indicar ao utilizador os seus tempos de execução. A classe **Crono**, cujo código fonte foi colocado no BB, usa o método **long System.nanoTime()** e os métodos **start()**, **stop()** e **print()** para realizar tais tarefas de medição de tempos.

3.3.- Medidas de performance (20%).

Pretende-se realizar alguns testes de “performance” meramente experimentalistas e apenas com o intuito de introduzir a ideia, ainda não muito interiorizada por razões óbvias, de que os programas são entidades que possuem alguns atributos que são mensuráveis e cuja análise pode ser útil (cf. *profiling*).

Estes testes podem e devem ser realizados fora do contexto do programa anteriormente desenvolvido (não farão parte da funcionalidade do programa) usando as partes do código e as estruturas de dados necessárias para cada teste.

Assim, como requisito complementar do projecto e valendo 20% da nota final do mesmo, pede-se aos alunos que realizem os seguintes testes e apresentem os resultados dos mesmos da forma mais simples e ilustrativa possível:

- 1.- Tempos de leitura (sem *parsing*) do ficheiro de base, **publicx.txt**, usando as classes **Scanner()** e **BufferedReader()**; Usar em seguida os ficheiros **publicx_x4.txt**,

`publicx_x6.txt` e fazer novas medições de tempos; Realizar os mesmos testes mas, agora, incluindo o tempo de *parsing*, ou seja, de separação dos autores e ano.

2.- Sendo para todos evidente que a estrutura de dados do projecto se vai basear na utilização de colecções do tipo `Map<K, V>`, bem como na utilização de colecções que são do tipo `Set<E>` e `List<E>`, comparar as “performances” de alguns dos queries mais complexos que foram pedidos.

Para tal, deve usar-se `HashMap<>` onde se usou `TreeMap<>` ou vice-versa, usar `HashSet<>` ou `LinkedHashSet<>` onde se usou `TreeSet<>` e vice-versa, e usar `Vector<>` onde se usou `ArrayList<>`.

Nota: *Estes testes aparentemente complexos e trabalhosos não o são de facto dado que a substituição de uma colecção por outra do mesmo tipo preserva a linguagem, ou seja, por terem a mesma API os métodos são os mesmos. Assim, apenas há que realizar find/replace nas declarações.*

Por outro lado, se programou os seus métodos usando interfaces e não classes concretas para definir os tipos dos parâmetros de entrada e dos resultados nada será alterado.

Afine os testes para o ficheiro `publicx.txt` e depois execute-os para os restantes ficheiros referidos acima. Apresente os resultados sob a forma de uma tabela e também de forma gráfica.

4.- Apresentação do projecto e Relatório.

O projecto será submetido por via electrónica sob a forma de uma pasta em formato zip/rar do projecto em `BlueJ`. Serão igualmente aceites projectos de outros IDEs como `NetBeans` e `Eclipse`. O código Java avaliado será o código que foi submetido.

O relatório do projecto de Java deverá ter a estrutura já conhecida, sendo de salientar agora a importância do diagrama de classes, do desenho da estrutura de dados usada e da apresentação dos resultados dos testes.

O relatório será entregue aquando da apresentação presencial do projecto e servirá de guião para a avaliação do mesmo.

F. Mário Martins