

UNIVERSIDADE DE AVEIRO

PROJETO DE MPEI

Pesquisa da Polícia

Autor:

Paulo VASCONCELOS

:84987

Autor:

Francisco MARTINHO

:85088

Professor regente:

Prof. Carlos BASTOS



December 8, 2017

1 Lista de módulos

- Bloom Filter (*BloomFilter.java*)
- MinHashing (*MinHashing.java*)
- Ficheiro a executar (*ProjectMain.java*)
- Ficheiro auxiliar (*GenerateRandomPeople.java*)
- Diversos (*Crime.java*, *Person.java*, ...)
- Ficheiro para testes (*teste.java*)

2 Procedimento de Execução

Este projeto contém apenas dois ficheiros que possuem, dentro dos mesmos, funções *main* e que, portanto, podem ser executados.

São esses os ficheiros *ProjectMain.java* (ficheiro principal de todo o projeto) e *GenerateRandomPeople.java* (ficheiro a executar apenas uma vez).

2.1 ProjectMain

Este é o ficheiro a executar para correr o programa desenvolvido no âmbito deste projeto.

O código presente no mesmo faz uso de todas as classes desenvolvidas (*Crime.java*, *CrimeReported.java*, entre outras) direta ou indiretamente.

É apresentado ao utilizador um menu para poder decidir que ações realizar com o programa.

2.2 GenerateRandomPeople

Dado que este projeto é construído a partir de uma base de dados da polícia, há que garantir que essa mesma base de dados existe.

Para evitar o trabalho e o absurdo gasto de tempo que seria estar a criar manualmente n ficheiros correspondentes a n pessoas para que fosse possível trabalhar com uma base de dados com um número razoável de entradas, desenvolvemos o ficheiro *GenerateRandomPeople.java*.

A função *main* deste ficheiro irá gerar n ficheiros contendo informações relativas a esse mesmo número de pessoas. Essa informação é gerada de forma aleatória podendo assumir os valores previamente estipulados nesse mesmo ficheiro. Veja-se o exemplo da cor do cabelo de uma determinada pessoa: o programa está preparado para gerar pessoas com cabelos louros, castanhos ou pretos sendo que a estes valores poderão ser adicionados outros de acordo com a vontade do utilizador tendo apenas de os inserir antes da execução do dito ficheiro.

3 Outros Módulos

3.1 BloomFilter

Ficheiro para instanciação de um *BloomFilter* genérico (input feito através de *Strings*).

O nosso Bloom Filter foi aplicado às pessoas constantes da base de dados que tinham registo criminal sendo as entradas do mesmo a concatenação do nome do crime cometido com o código hash da pessoa que o cometeu. É-nos assim possível verificar com maior rapidez se determinada pessoa cometeu ou não um determinado crime.

O número adequado de funções de hash para este filtro é calculado no mesmo (fórmula lá presente) tendo em consideração a taxa de falsos positivos considerada aceitável (definida como 0.001 por *default* mas passível de alteração) e o número de elementos que se espera inserir nesse mesmo filtro.

3.2 MinHashing

Ficheiro para instanciação de um MinHashing direcionado para o contexto do problema.

O nosso MinHashing contém os mínimos dos valores codificados dos diversos atributos de uma pessoa por 5000 funções de hash (por *default* mas, como anteriormente, passível de alteração).

3.3 Crime e classes derivadas

As classes *Crime*, *CrimeCommitted* e *CrimeReported* (bem como a classe *CriminalRecord*) são utilizadas para processar de forma mais eficiente e in-

tuitiva os dados acerca dos diversos crimes.

3.4 Person

A classe `Person` define os diversos atributos a serem considerados em cada pessoa enquanto elemento da base de dados ou enquanto suspeito avistado de um crime.

4 Testes

As classes foram testadas individualmente juntamente com os seus diversos métodos a partir do ficheiro *teste.java*. Este foi alterado com regularidade pelo que não é possível apresentar um registo dos diversos testes feitos mas a solidez do programa é assegurada por estruturas como *try-catches* e outros que tornam este programa extremamente fiável.

Os `Scanners` e alguns dos métodos e classes implementados foram convertidos de tipo primitivo *int*, *double* e *boolean* para as respetivas classes de encapsulamento *Integer*, *Double* e *Boolean* para suportarem também o valor ***null*** para permitir a não inserção de atributos desconhecidos de pessoas entre outras situações ou pormenores nas quais a nulidade tinha de ser um valor aceite.

5 Manual do utilizador

5.1 Introdução

Este programa deve ser tomado como algo a ser executado/operado por um funcionário de uma esquadra da polícia (leia-se "alguém experiente que teve formação e sabe que *software* está a operar).

Ainda assim, a utilização deste programa é algo bastante intuitivo e simples.

5.2 Inexistência de Base de Dados

Tal como referido anteriormente, caso a base de dados ainda não exista e se pretenda simular uma, o primeiro passo corresponderá à execução do ficheiro

GenerateRandomPeople.java. Este gerará uma base de dados que, por *default*, conterá 10000 pessoas (não só criminosos mas também pessoas com um registo criminal limpo).

5.3 Programa

Tendo já uma base de dados para trabalhar, o utilizador deve executar o ficheiro *ProjectMain.java*.

Este programa irá apresentar um menu com quatro opções:

1. Report Crime
2. Add Subject to Database
3. Search Subject
4. Exit

É a partir deste que o desenrolar do programa se dá.

5.3.1 Report Crime

Ao reportar um crime, o programa irá primeiro pedir ao utilizador para indicar que crime quer reportar.

Depois desta informação ser introduzida, o programa questionará o utilizador quanto ao avistamento ou não de suspeitos. Caso haja suspeitos avistados, serão recolhidas as informações dos mesmos (características físicas).

O programa retornará então, com auxílio do MinHashing, os utilizadores presentes na base de dados que mais se assemelharem a cada suspeito descrito sendo depois estes utilizadores separados conforme tenham ou não cometido previamente este crime. Esta separação é feita com recurso ao Bloom Filter.

5.3.2 Add Subject to Database

A segunda opção do menu permite ao utilizador inserir as características relativas a uma pessoa e inserir essa nova pessoa criada na base de dados da polícia.

Este procedimento cria a pessoa não só enquanto um objeto da classe *Person* no programa a correr como também cria um ficheiro para que essa pessoa continue na base de dados da próxima vez que alguém ligar o programa.

5.3.3 Search Subject

Após selecionar esta opção, o utilizador deverá inserir as informações relativas ao sujeito que pretende procurar na base de dados.

Através de MinHashing, o programa retornará as pessoas com uma descrição o mais próxima possível do sujeito.

5.3.4 Exit

Apesar de não ser uma opção que necessite de muita explicação, a quarta opção do menu permite ao utilizador terminar o programa.