

Universidade do Minho

MESTRADO INTEGRADO EM ENGENHARIA
INFORMÁTICA

PROGRAMAÇÃO ORIENTADA A OBJETOS

Java Fatura

Membros do Grupo 66

Bruno Martins A80410 Leonardo Neri A80056 Márcio
Sousa A82400

1 Introdução

Este projeto teve como objetivo criar um programa em na linguagem de programação C que permitisse o processamento de **notebooks**. Um **notebook** é um ficheiro de texto que contém comentários e comandos e que depois do seu processamento também contém o resultado dos comandos. O conteúdo do ficheiro, como referido anteriormente, contém um comando, que começa obrigatoriamente com o símbolo \$, e o seu output está delimitado entre *\$\$\$*.

2 Estruturas de Dados

2.1 COMMAND

Para o armazenamento do conteúdo dos **notebooks** foi criada uma estrutura *COMMAND* que contém três campos. Uma *String input* que contém o comando a executar, uma *String output* que contém o output do comando e uma *String comment* que contém o comentário que está imediatamente antes de comando.

2.2 LIST

Para assegurar uma conexão entre todas as células *COMMAND* foi utilizada uma lista ligada. Esta lista está ordenada tendo em conta a organização do *notebook* a ser processado.

3 Execução

3.1 Leitura e parsing do ficheiro

A execução do programa começa pela abertura do ficheiro passado como argumento com a *system call open*, de seguida é lido o texto que lá está dentro linha a linha, um caracter de cada vez usando o *read*. Logo após ser finalizada a leitura da linha, o conteúdo dela é processada por uma função de *parsing* que verifica se é um comando, um comentário ou um output e insere no campo designado da estrutura *COMMAND*.

3.2 Empresa

- **Obter as faturas da empresa ordenadas por data ou valor:** o procedimento é o mesmo que o que permite aos contribuintes verem todas as faturas emitidas em seu nome, mas desta vez, aplicado a uma empresa. Depois de ter a lista das faturas, esta é ordenada pela ordem desejada.
- **Obter a lista de faturas por contribuinte num intervalo de datas:** as faturas da empresa em questão são filtradas de acordo com o NIF cliente, e depois por data, sendo retornadas a lista de faturas que corresponde a esses dois critérios.
- **Obter a lista de faturas por contribuinte ordenadas por valor de despesa:** as faturas da empresa em questão são filtradas por NIF do contribuinte, e é retornada a lista de Faturas ordenada por valor decrescente de despesa, recorrendo a um comparator.
- **Total faturado por uma empresa:** é percorrida a lista das faturas da empresa, somando o valor das faturas que estão dentro do período de tempo estipulado, sendo retornado o valor total.

3.3 Admin

- **Relação 10 contribuintes que mais gastam:** é percorrido o HashMap que contém todas as identidades fiscais e são filtrados todos os contribuintes individuais para uma lista que é ordenada segundo os gastos de cada um deles. No final são mantidos apenas na lista os primeiros dez contribuintes.
- **Relação 10 empresas que mais faturam:** é percorrido o HashMap que contém todas as identidades fiscais e são filtradas todas as Empresas para uma lista que é ordenada segundo o total faturado de cada um deles. No final mantemos apenas na lista os primeiros dez contribuintes

4 Conclusão

Este projeto foi uma excelente oportunidade para aplicar conhecimentos relativos ao paradigma de programação orientada a objetos, usando Java como

linguagem de desenvolvimento do projeto. Relativamente a possíveis aspetos a melhorar no projeto, salienta-se que poderia ter sido usado um algoritmo de cálculo de dedução mais específico, e uma lista de concelhos que abrangesse todos os concelhos do país.