

Enunciados de Trabalhos

Relatório da 2ª parte do trabalho prático



FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

Grupo H

Hugo Cunha – up201404587

Tiago Carvalho – up201504461

Ilona Generalova – up201400035

2 Fevereiro 2017

Mestrado Integrado em Engenharia Informática e Computadores

Algoritmos e Estruturas de Dados

Índice

Descrição do tema	2
Solução implementada	2
Diagrama UML	4
Lista de Casos de Utilização	5
Relato de Dificuldades	6
Esforço dedicado	6

Descrição do tema

Com este trabalho pretendemos implementar um sistema que permite um docente de uma unidade curricular gerir todos os enunciados para que possa apresentar aos seus alunos. O programa deve também permitir escolher, em situações de pouca imaginação por parte do docente, enunciados de anos passados que não tenham tido muita adesão ou mesmo que não tenham sido utilizados há um número significativo de anos.

O sistema deve, ainda, permitir ao utilizador(docente) alterar pequenas informações como, por exemplo, nomes de alunos, monitores de projetos etc...

Solução implementada

No nosso trabalho, tivemos em atenção a necessidade de facilitar a gestão de dados para o utilizador, portanto implementamos uma *interface* de consola de simples utilização, recorrendo apenas ao teclado.

Com esta conseguimos gerir tanto enunciados, como ocorrências e, dentro destas conseguimos gerir os grupos.

Como sugerido na descrição do trabalho, implementamos três tipos de enunciados: pesquisa, análise e desenvolvimento. Cada um destes possui um repositório de dados, que no caso do nosso trabalho são apenas *strings*.

Para cada ocorrência interessa-nos guardas, apenas, o ano letivo e um vetor de grupos de trabalho.

Em cada grupo de trabalho resolvemos guardar um professor/monitor responsável por ajudar os alunos na resolução de um enunciado. Guardamos, ainda dentro de cada grupo um vetor com os alunos pertencentes a esse grupo.

A classe *Person* tem duas subclasses: *Student* e *Professor*.

Cada *student* tem dois vetores que partilham índices, ou seja, o primeiro contém títulos de enunciados e o segundo, na mesma posição, a nota que ele obteve.

No que concerne o *Professor* apenas nos interessa guardar o “básico” que já é guardado em *Person*, ou seja, nome.

Desta forma, no início do nosso programa são carregados todos os projetos, pessoas (estudantes ou professores) e ocorrências para os respectivos vetores. De seguida é mostrado o menu principal que se encontra dividido em quatro opções: gerir enunciados, gerir pessoas ou criação de novas ocorrência. Esta última parte consideramos a mais importante uma vez que é esta responsável pela implementação do algoritmo que considera enunciados anteriores e que os sugere para uma nova ocorrência da unidade curricular.

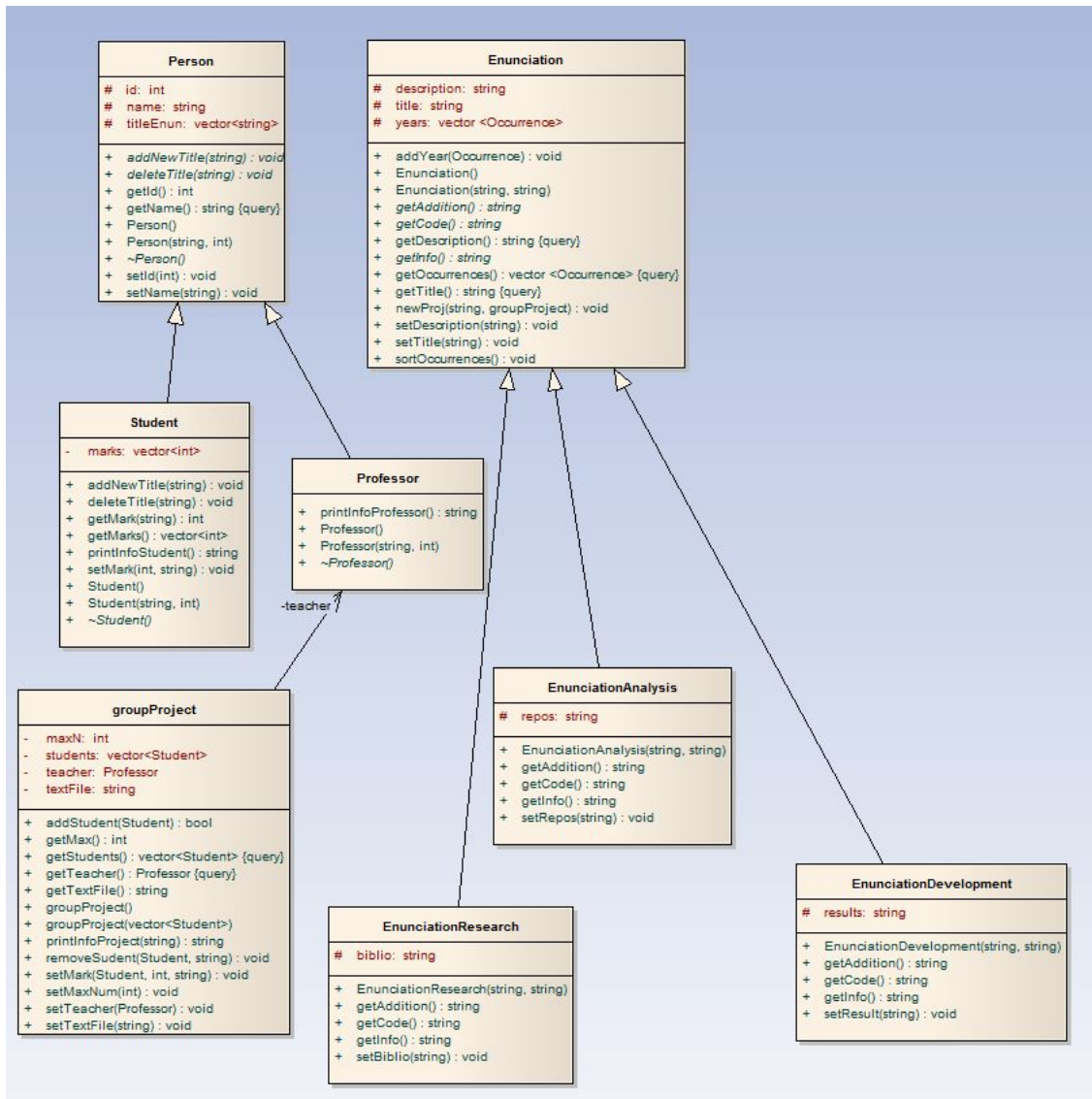
À medida que os trabalhos são feitas são armazenados na árvore binária de pesquisa ordenada por tipo, enunciado e ano.

Os trabalhos que necessitam de ser avaliados estão numa fila de prioridade e quando o utilizador deseja avaliar trabalhos, acedemos a esta estrutura até estar vazia, ou ele desejar parar de avaliar trabalhos.

Os trabalhos mais antigos (mais de 5 anos) são guardados numa tabela de dispersão, ordenada.

No final, ou seja, quando o utilizador decide sair do programa, são guardados todos os dados do programa, gerados de novo ou anteriormente, e colocamos essa informação em diversos ficheiros de texto para futuras execuções do código.

Diagrama UML



Lista de Casos de Utilização

- Criar novos enunciados de trabalhos
- Aceder à lista de todos os enunciados já criados
- Filtrar os enunciados pelo tipo
- Aceder à informação de um enunciado
- Acrescentar ocorrências aos enunciados já criados
- Dois algoritmos que filtram os enunciados apresentando, apenas aqueles que são recomendados a ser reutilizados no ano presente
- Aceder à lista de todos os estudantes
- Aceder à lista de todos os professores/monitores
- Acrescentar novos estudantes
- Acrescentar novos professores/monitores
- Aceder à informação de um estudante em específico, através do nome ou ID
- Aceder à informação de um professor/monitor em específico, através do nome ou ID
- Guardar os trabalhos realizados
- Saber os trabalhos mais recentes de um enunciado
- Obter listagens de enunciados agrupados por tipo

Relato de Dificuldades

A principal dificuldade encontrada foi quanto à organização de classes/subclasses.

Inicialmente não foi claro qual deveria ser a hierarquia de classes, o que resultou em algumas alterações do esquema do projeto. Acabamos por adotar esta estrutura e relação entre classes, no entanto, sem grande certeza se terá sido a melhor hipótese.

Com a mesma importância da dificuldade anterior foi também a parte de ler e escrever nos ficheiros, pois é sempre uma parte com alguma dificuldade. Mesmo depois de passarmos à fase de testes, estas funções continuaram a gerar erros, tomando grande parte de *debugging*.

Dado que na tabela de dispersão utilizamos apontadores para o tipo de dados tivemos que adaptar os operadores, tarefa que não nos lembramos logo de início, tendo perdido algum tempo.

Esforço dedicado

Todos os membros cumpriram com o que lhes tinha sido proposto, contribuindo também com ideias para o projeto.