Faculdade de Ciências e Tecnologias
Universidade de Coimbra
2016/2017

Programação Orientada a Objetos

Ana Catarina Jesus Gonçalves nº2013167088
Ana Rita Ferreira Alfaro nº 2013150362

Introdução

Este projecto foi nos proposto no âmbito da disciplina de Programação Orientada aos Objectos, em que o objectivo é criar uma aplicação de gestão de exames, com a capacidade de ajudar o utilizador a gerir as suas disciplinas, exames e/ou vigilâncias.

O utilizador pode ser de três tipos diferentes, Aluno, Docente ou Não Docente e cada um deles tem diferentes funções.

Foi nos então pedido que o programa tivesse as seguintes funcionalidades:

- Criar exames e associar docentes;
- Configurar a sala do exame e assegurar que está livre.
- Convocar vigilantes e funcionários;
- Inscrever alunos em exame assegurando que têm essa disciplina e têm acesso à época de exame em causa;
- Lançar notas de um exame;
- Listar exames;
- Listar alunos inscritos e suas classificações;
- Listar exames em que o aluno está inscrito e suas classificações;
- Listar docentes e funcionários associados a um exame;
- Listar exames em que funcionários estão envolvidos;
- Listar notas de um exame.

Para a realização deste trabalho utilizamos a linguagem Java com o apoio do IDE NetBeans.

Descrição das Classes

Criámos 14 classes que considerámos importantes para a criação deste trabalho. Sendo elas:

- **ProjectoNovo:** É nesta classe que se encontra o main. E por essa razão, esta classe é muito importante para a execução do programa. Foi criado um menu para facilitar o uso do programa, é no menu que estão a ser executados todas os métodos do programa.

- **Dei:** Uma classe muito importante, visto que é nela que estão a maioria dos métodos. Esta classe contém todos os ArrayLists globais, ou seja, que contêm todos os exames, docentes, não docentes, etc.
-
- **Pessoa:**Classe *abstract* visto que não vai ser criado nenhum objeto do tipo Pessoa, mas que é importante, servindo de superclasse para a classe Funcionário e Aluno.
- **Funcionário:** Herda todos os métodos e variáveis da superclasse Pessoa. Além disso, esta classe, é ela mesma uma superclasse para as classes Docente e NaoDocente. Isto para facilitar a criação de subclasses e evitando a repetição de código.
- **Docente:** Uma classe que *extend* de Funcionário, usando hereditariedade. Este Docente tem a função de criar exames, lançar notas, listar as suas vigilâncias, etc. Para isso cada Docente tem um número mecanográfico diferente, podendo assim ser distinguido dos outros docentes. Além disso, este Docente pode ser Docente Responsável de uma disciplina podendo portanto lançar as notas dessa disciplina.
- **NaoDocente:** Uma classe que *extend* de Funcionário. Este NaoDocente pode ser convocado para vários exames ao mesmo tempo.
- **Aluno:** Uma classe que *extend* de Pessoa. Este Aluno tem várias opções na utilização do programa. Em primeiro lugar para se inscrever num Exame, tem que obrigatoriamente estar inscrito na disciplina desse exame. Estando inscrito na disciplina, pode se inscrever em exame (Normal ou Recurso), tendo limitações quando se tenta inscrever em Exame Especial. Além disso pode se desinscrever a exame e consegue também listar os seus exames e as suas disciplinas.
- **Disciplina:** Esta classe é bastante importante no decorrer do programa. O aluno precisa de estar inscrito numa disciplina para conseguir se inscrever no exame desta.
- **Sala:** Esta classe embora bastante simples, tem o seu grau de importância, pois contém a informação da sala onde se irão realizar os exames criados.

  Tem um método que verifica a disponibilidade da sala, baseando-se num horário onde tem a informação dos tempos em que esta se encontra ocupada, e este é usado quando se cria um exame.

- **Exame:** Uma das classes mais importantes para o funcionamento do programa. Tem como atributos o tipo de exame (i.e. Norma, Recurso ou Especial), a disciplina de que se trata, data de realização, duração da prova, o docente responsável da disciplina, os docentes que leccionam a disciplina e portanto, os vigilantes do exame, os não docentes que também contribuem para a vigilância do exame, os alunos inscritos e as notas obtidas.
- **Curso:** Esta classe tem como atributos o nome do curso e as disciplinas que o constituem. Apesar de ser uma informação importante no histórico do Aluno, não conseguimos tirar tanto partido dela como esperávamos.
- **Date:** Esta classe tem como atributos o dia, o mês, o ano, a hora e o minuto. Esta classe é utilizada na classe Exame e na classe Sala. É bastante útil, pois como o auxílio desta que conseguimos calcular se um Docente tem vigilância sobreposta com outras e se uma determinada Sala está livre numa certa hora

  para a realização de um Exame.

Funcionamento do programa

O programa começa por listar o menu inicial, um menu simples que conta por escolher que tipo de utilizador quer utilizar o programa. Pode escolher entre: Docente, Não Docente, Aluno ou Secretaria.

- **Menu Docente**: Este menu tem várias opções em que só o docente pode fazer, e isso acontece com o número mecanográfico. Qualquer opção desde criar exames até listar vigilâncias pede o número mecanográfico do docente. Caso seja responsável por uma disciplina pode também atribuir notas a cada um dos alunos inscritos no exame que pretende lançar notas.
- **Menu NaoDocente:** Este menu é bastante simples. Os Não Docentes têm apenas a opção de listar as vigilâncias em que estão associados, podendo ser várias ao mesmo tempo.
- **Menu Aluno:** Os alunos têm várias opções no menu, passando por se inscrever ou desinscrever a uma disciplina. Depois de inscritos na disciplina podem se inscrevem ou desinscrever em exame (Normal, Recurso, Especial). Listar as suas disciplinas e exames em que estão inscritos e suas classificações.
- **Menu Secretaria:** Serve essencialmente para obter informações gerais sobre disciplinas, docentes, alunos, não docentes e exames. E serve também para criar estas entidades.

Ficheiros

Por último, uma parte bastante importante, os ficheiros.

Criámos 7 ficheiros como pequenas bases de dados. Estes são atualizados sempre que surge uma alteração de informação nos ArrayList globais e sempre que o programa inicia é carregada toda a informação contida nos ficheiros para os mesmos ArrayList.

Os ficheiros criados são os seguintes:

- **“docentes.dat”**
- **“alunos.dat”**
- **“naoDocentes.dat”**
- **“exames.dat”**
- **“sala.dat”**
- **“disciplinas.dat”**
- **“cursos.dat”**

Nestes ficheiros estão guardadas todas as informações dos ArrayList dos Objectos Docente, Aluno, NaoDocente, Exame, Sala, Disciplina, Curso, respectivamente.

UML inicial