IPS Instituto Politécnico de Setúbal
Escola Superior de Tecnologia de Setúbal

Introdução à Programação Orientada por Objetos

Projeto – 1ª Fase e 2ª fase

MyCalendar

Sistema de gestão de agenda escolar

Horas	Segunda	Terça	Quarta	Quinta	Sexta	Sábado
08:00 - 08:30						
08:30 - 09:00	IPOO (PL) 1ºINF-03 ESTS F152 CG	MI (PL) 1ºINF-03-04 ESTS E314 MRi			LC (PL) 1ºINF-03-04 ESTS F155 AGo+PFe	
09:00 - 09:30			LC (TP) 1ºINF-1-3 ESTS F304 AGo+PFe	MI (TP) 1ºINF-03-04 ESTS F253 CAl		
09:30 - 10:00						
10:00 - 10:30						
10:30 - 11:00	IPOO (TP) 1ºINF-1-3 ESTS F251 JASP	EG (TP) 1ºINF-03-04 ESTS F253 EFe	IPOO (TP) 1ºINF-1-3 ESTS F251 JASP	EG (TP) 1ºINF-03-04 ESTS F253 FVa		
11:00 - 11:30						
11:30 - 12:00						
12:00 - 12:30						
12:30 - 13:00						
13:00 - 13:30	MEC (TP) 1ºINF-1-3 ESTS F251 ECL					
13:30 - 14:00			MEC (TP) 1ºINF-1-3 ESTS F251 ECL			
14:00 - 14:30						
14:30 - 15:00		MI (TP) 1ºINF-03-04 ESTS F253 CAl				
15:00 - 15:30						
15:30 - 16:00						
16:00 - 16:30						
16:30 - 17:00						
17:00 - 17:30						

Ano Letivo: 2018/2019

Época Normal e de Recurso

Índice

Introdução

O objetivo deste projeto é desenvolver uma aplicação, utilizando a linguagem Java e a Programação Orientada por Objetos (POO), com o intuito de armazenar e visualizar a agenda escolar de alunos da ESTSetúbal/IPS.

No problema a modelar, destacam-se as seguintes entidades:

- A agenda de um aluno que corresponde à um semestre de aula e na qual um aluno pode registar os vários compromissos académicos que este irá ter durante o semestre,
- Os alunos que criam e visualizam as suas agendas para gerir a organização de um semestre,
- Os docentes que lecionam, regem unidades curriculares, e atendem alunos,
- Os cursos nos quais os alunos estão inscritos,
- As unidades curriculares nas quais são lecionados tópicos específicos e realizadas avaliações
- Os compromissos que um aluno tem ao longo de um semestre. Estes compromissos podem ser divididos em dois tipos: aulas e avaliações (como por exemplo os exames ou as entregas de projetos e trabalhos),
- As salas onde ocorrem as aulas ou as avaliações e onde os docentes atendem os alunos.
- Os grupos de alunos que se juntam para fazer trabalhos no âmbito de determinadas unidades curriculares,

Além das entidades da agenda será ainda necessário ter um visualizador que irá permitir visualizar as entidades descritas. Com isto será possível criar e analisar um "semestre de aulas".

Na secção seguinte são apresentados os detalhes a considerar na implementação do projeto.

Primeira Fase

Detalhes de Implementação

Características e funcionalidades

Aluno

Um aluno é caracterizado por:

- Um nome, uma cadeia de caracteres
- Um número, uma cadeia de caracteres,
- Uma data de nascimento, uma data
- Um género, um caracter
- Uma referência para o curso no qual está inscrito
- Uma lista de agendas, onde estão guardadas as agendas de cada um dos semestres do curso.

As principais funcionalidades a implementar nesta entidade são as seguintes:

- Deve ser possível adicionar novas agendas
- Deve ser possível inserir as unidades curriculares de um semestre e os respetivos compromissos,
- É importante saber qual a agenda ativa em função da data atual, quando é feita a consulta.

Agenda

Um aluno possuí uma agenda para cada semestre. Cada agenda deverá permitir guardar a seguinte informação:

- A data de início do semestre,
- A data de fim do semestre,
- A indicação do semestre ser par ou ímpar,
- Um grupo de unidades curriculares nas quais o aluno está inscrito nesse semestre
- Um grupo de docentes que lecionam para este aluno, neste semestre
- Uma lista de grupos aos quais o aluno pertence nas várias unidades curriculares do semestre

As principais funcionalidades a implementar nesta entidade são as seguintes:

- Deve ser possível adicionar ou alterar os grupos aos quais pertence o aluno.
- Deve ser possível consultar a lista de grupos de um aluno para um semestre.
- Deve ser possível extrair da agenda informações tais como
    - o total de créditos das unidades curriculares nas quais o aluno está inscrito,
    - a lista de docentes do aluno nesse semestre,
    - o total das horas das disciplinas do semestre
    - a lista das avaliações das disciplinas do semestre

Curso

Os cursos são caracterizados pelo:

- Nome, uma cadeia de caracteres com a sigla do curso (e.g. Engenharia do Ambiente),
- Tipo: existem três tipos de cursos: Licenciaturas, Pós-graduações e Mestrados (tipo enumerado),
- Código, uma cadeia de caracteres com a sigla do curso
- Créditos: o número de créditos necessários para completar o curso (e.g. 180 para uma licenciatura).

Docente

Os docentes são necessários para lecionar e reger unidades curriculares, tendo de pertencer ao corpo docente de um departamento.

Como principais características há a referir:

- O nome (uma cadeia de caracteres),

- O número mecanográfico do docente (um número inteiro),
- Os iniciais que permitem representar de forma abreviada o docente (uma cadeia de caracteres),
- Uma referência para o departamento ao qual pertence.
- O e-mail do docente (uma cadeia de caracteres),
- O horário de atendimento, uma referência para um grupo de horários. O total dos vários horários de atendimento deve atingir no mínimo 2h horas semanas.
- Uma referência para a sala onde o docente recebe os alunos.

As principais funcionalidades a implementar nesta entidade são as seguintes:

- Deve ser possível alterar o horário de atendimento de um docente.
- Deve ser possível alterar a sala do docente.
- Qualquer pessoa deve poder consultar o horário e a sala de atendimento de um docente.

Departamento

Um departamento é a entidade que agrega os docentes de uma mesma área científica. Um Departamento tem um nome, uma lista de cursos dos quais é responsável e uma lista de docentes.

Aulas

As aulas podem ser de três tipo: teóricas, práticas ou laboratoriais. As aulas podem ter uma duração de uma hora, uma hora e meia, duas horas ou quatro horas. As aulas decorrem numa determinada sala e são lecionadas por um determinado docente. Deverá, portanto, ser necessário guardar as referências para cada um destes dados. A outra característica importante a cerca de uma aula é o horário de agendamento desta aula. Quando se insere uma aula, deve ser possível inserir o número de semanas em que uma aula será repetida. Cada tipo de aula de uma disciplina deve, em princípio, ser repetida durante 15 semanas sucessíveis. Contudo, é necessário precisar o número de repetições de uma aula porque existem aula de compensação ou aulas extras.

Unidades curriculares

As aulas pertencem a uma determinada unidade curricular (UC) que permite representar

Como principais características há a referir:

- O nome;
- O código da unidade curricular guardado como um conjunto de caracteres cujas três primeiras letras é a sigla do curso em que se inseres (E.G. INF32145);
- O semestre em que ocorre a UC (um valor inteiro),
- O número de crédito que esta unidade curricular representa;
- Uma lista de aulas onde são inseridas todas as aulas de todos os tipos;
- Uma lista de avaliações onde são guardadas todas as avaliações da UC;
- Uma referência para o departamento responsável por esta unidade curricular.
- Uma referência para o docente responsável da Unidade curricular

As principais funcionalidades a implementar nesta entidade são as seguintes:

- Deve ser possível obter a partir da unidade curricular as informações tais como:
    - a lista de docentes da UC,
    - a lista das aulas e das aulas que faltam até o final do semestre
    - a lista das avaliações da disciplina e das avaliações que faltam até o final do semestre
    - o docente responsável

Grupo

Um grupo é formado por um conjunto de alunos que pode ir de 1 até 10 elementos. Um grupo pode ser formado por apenas um aluno quando outros alunos desistem do grupo ou quando um docente autoriza excecionalmente que um grupo seja formado por um único aluno. Desta forma, um grupo é caracterizado por:

- Uma lista de alunos
- Uma referência para a unidade curricular a qual diz respeito o grupo.

Como principais funcionalidades há a referir:

- Adicionar e retirar elementos de um grupo.
- Obter a lista de alunos que fazem parte do grupo.

Horário

Um horário permite representar a altura em que acontece um determinado compromisso ou uma determinada avaliação. Um horário deverá permitir guardar a data e hora de início, a data e hora de fim e a duração do compromisso.

Avaliações

As avaliações são de vários tipos. Podem ser testes, exames, apresentações ou entregas de projeto. Consoante o tipo de avaliação, estas podem ocorrer numa determinada sala e numa determinada data. Deverá ser possível adicionar uma descrição à avaliação, por exemplo para inserir se é necessário levar folhas de teste ou folha A4 com matéria resumida.

Salas

As salas são os locais onde ocorrem as aulas, as avaliações e o atendimento aos alunos. Cada sala é caracterizada por:

- A zona do edifício, um caráter que guarda a letra que caracteriza a zona do edifício;
- O andar do edifício, um valor inteiro entre 0 e 2;
- O número da sala neste andar;
- A capacidade da sala, um valor inteiro, que armazena o número de pessoas que pode conter a sala.

A visualização de uma sala é feita sob a forma de uma cadeia de caracteres, por exemplo F258 para a sala número 58, da Zona F no andar 2.

Visualizador

Esta entidade deverá permitir visualizar a informação de uma agenda, e deverá receber as listas de todos os elementos necessários instanciados:

- Lista de docentes.
- Lista de alunos com as agendas.
- Lista de salas.
- Lista de avaliações

Deverá ser possível visualizar:

- a lista de grupos de que faz parte um aluno num determinado semestre
- a agenda de uma determinada semana do semestre, de 2ª feira a sábado,
- a agenda de um determinado dia do semestre
- as avaliações que faltam até o fim do semestre para uma determinada unidade curricular
- o horário de atendimento dos docentes de uma determinada unidade curricular
- a lista das aulas que ocorrem numa determinada sala
- a lista de disciplinas de um semestre com os respetivos responsáveis.

Regras de Desenvolvimento e Entrega da 1ª fase do Projeto

Entrega

A data de entrega da 1ª fase do projeto é **22 de dezembro de 2018, até as 22h00**.

Implementação e codificação

O programa deve ser desenvolvido utilizando a linguagem Java, colocando em prática os conceitos fundamentais do paradigma de Programação Orientada por Objetos.

Em relação às regras de codificação, siga as convenções adotadas normalmente para a linguagem Java:

- A notação *camelCase* para os nomes de métodos e dos seus parâmetros, de variáveis locais e de atributos;
- A notação *PascalCase* para os nomes das classes;
- Não utilize o símbolo '_' nos identificadores, nem abreviaturas.

É necessário que o projeto cumpra o que é pedido no seu enunciado, sendo deixado ao critério do programador qualquer pormenor de implementação que não seja referido, o qual deverá ser devidamente documentado.

Nas funcionalidades desenvolvidas, deverão ser incluídas todas as validações necessárias para impedir um comportamento incorreto do sistema.

Sempre que fizer sentido, os métodos deverão ser responsáveis por apresentar no ecrã mensagens de informação e/ou erro, indicando o processamento que foi feito ao objeto.

Nas situações em que for adequado, devem ser facultadas diferentes assinaturas para os métodos implementados.

Os nomes das classes, atributos e métodos deverão ser definidos em língua inglesa. As mensagens apresentadas pela aplicação podem ser apresentadas em português e/ou inglês.

Constituição de grupos

Cada projeto deverá ser elaborado em grupos de dois alunos, podendo ser desenvolvido individualmente em casos pontuais devidamente justificados. Não serão permitidos, **em nenhum caso**, grupos com mais do que dois alunos.

Os grupos já se encontram determinados através da metodologia de *pair programming* que está a ser utilizada nos laboratórios. Caso existam alunos que não têm o grupo escolhido, deverão contactar o respetivo docente de laboratório para regularizar a situação.

Entrega do projeto

- O projeto deverá ser entregue até à data limite especificada por **via exclusivamente eletrónica, utilizando a área dos trabalhos na plataforma Moodle.** Todos os ficheiros que compõem o projeto deverão estar guardados num único ficheiro compactado em **formato ZIP**. Em caso de dificuldades no acesso à plataforma Moodle, o envio dos ficheiros poderá ser feito por correio eletrónico para o respetivo docente de laboratório, dentro do prazo acima indicado.

- **Não serão aceites quaisquer projetos entregues fora do prazo!**

- Todos os materiais do projeto devem ser devidamente identificados com nome, número e endereço de correio eletrónico dos alunos.

Os materiais do projeto deverão incluir:

- A documentação do programa, explicando de forma simples as classes criadas, juntamente com os seus atributos e métodos, bem como qualquer detalhe de implementação que necessite de explicações adicionais.

- O código fonte do programa na forma de projeto em BlueJ ou Netbeans, com todas as funcionalidades implementadas. **Não existe necessidade de implementação de uma interface com o utilizador na primeira fase.** Deverá apenas testar o seu código, criando objetos com o BlueJ e executando os respetivos métodos. Poderá gravar as instruções de teste num ficheiro e executá-las a qualquer altura usando o *Scratchpad*.

- Todos os ficheiros que compõem o projeto deverão estar guardados num único ficheiro compactado em formato ZIP cujo nome deverá ter a seguinte nomenclatura: **<*curso*>_<*numAluno1*>_<*numAluno2*>.zip (exemplo: EI_12345678_87654321). O incumprimento das normas de entrega está sujeito a uma penalização de até 1 valor.**

Regras e Critérios de Avaliação do Projeto

Regras de Avaliação

A avaliação do projeto está sujeita às seguintes regras:

- Não serão aceites quaisquer projetos entregues fora do prazo.
- A classificação do programa terá em conta a qualidade da programação e a estrutura do código criado segundo os princípios da Programação Orientada por Objetos.
- Serão premiadas a imaginação e a criatividade.
- Todos os projetos serão submetidos a um sistema automático de deteção de cópias. Os projetos que forem identificados como possíveis cópias, e verificando-se serem efetivamente cópias, serão anulados.

Critérios de Avaliação

Este primeiro projeto será avaliado segundo os seguintes critérios:

Entidades	**50%**
Docentes	5%
Alunos, cursos	5%
Horários	5%
Salas	5%
Visualizador	15%
Compromissos: Avaliações, Aulas, Entregas de trabalhos	10%
Grupos	5%

Implementação	**40%**
Estrutura do projeto, definição e relação entre as classes	10%
Estrutura e funcionamento interno das classes	10%
Conhecimento e boa utilização da linguagem	10%
Bom estilo (identificadores, comentários, indentação)	10%

Avaliação qualitativa	**10%**
Qualidade geral, detalhes de implementação e funcionalidades adicionais	10%

Bom trabalho!

Segunda fase

Código e Estrutura de Classes

Nesta segunda fase é levantada a restrição imposta anteriormente de uso exclusivo de *arrays* para representação de coleções. Assim, **onde se justificar**, os *arrays* devem ser alterados para uma representação com recurso a uma das coleções do Java que foram aprendidas. É importante que o tipo de coleções escolhido seja adequado à coleção que representa. Na escolha dos tipos de coleção tenha em consideração as novas funcionalidades que são pedidas.

Não existem restrições à modificação de código entregue na primeira fase. Neste caso, pode aproveitar para melhorar a qualidade da estrutura de classes e do código de acordo com as boas práticas de escrita, nomeadamente: um acoplamento fraco, uma boa coesão e um desenho orientado por responsabilidades.

Documentação e testes

Deverá ser criada uma bateria de testes adequada às entidades criadas, recorrendo à *framework* **JUnit**, conforme abordado na disciplina. Deste modo, os testes deverão ser suficientemente extensivos para que toda a lógica fundamental do projeto seja validada.

Pretende-se igualmente a criação de documentação adequada às diversas entidades implementadas, neste caso com recurso a **JavaDoc**.

Novos Requisitos

Adicionalmente aos requisitos expostos na primeira fase, o sistema deverá ter em conta as seguintes necessidades:

Menu de interação com o utilizador de estatísticas

No início da execução do programa, será necessário mostrar ao utilizador as opções de utilização da aplicação: inserção de docentes, de salas, de alunos, de unidades curriculares, de docente e compromissos. Além das opções de inserção de dados, será também necessário mostrar os modos de visualização disponíveis: diário e semanal

Preparação de estatísticas

Nesta segunda fase, além da visualização das agendas, o programa deve permitir verificar se existe um docente livre naquele momento. Também deve permitir visualizar um conjunto de estatísticas sobre os compromissos. Sinta-se livre para criar o código auxiliar que achar necessário para guardar os dados pretendidos. Sempre que possível, utilize o processamento funcional de coleções na obtenção e visualização dos dados. Estatísticas:

- Número total aulas de aulas, organizadas por tipo de aulas;
- Número total de horas de avaliações de um semestre;

- Número total de avaliações ainda por fazer, considerando todas as disciplinas do semestre;
- Número total de horas de aulas até o fim do semestre, considerando todas as disciplinas do semestre;
- Número total de créditos das disciplinas do semestre,
- Média de horas semanais, considerando todas as disciplinas do semestre.
- Número médio de aulas por semestre de um aluno, i.e., média de horas de aulas por semestre de todas as agendas de um aluno.

Regras de Desenvolvimento e Entrega da 2^{aa} fase do Projeto

Entrega

A data de entrega da 2ª fase do projeto é **27 de janeiro de 2018 as 22h00**.

Implementação e codificação

As regras de implementação e de codificação da segunda fase são idênticas às regras da primeira fase.

Constituição de grupos

Os grupos para a 2ª fase são os mesmos do que os grupos da 1ª fase.

Entrega do projeto

As instruções para a entrega da segunda fase são idênticas as instruções às da primeira fase.

Regras e Critérios de Avaliação do Projeto

Regras de Avaliação

A avaliação da segunda fase do projeto está sujeita às seguintes regras:

- Não serão aceites quaisquer projetos entregues fora do prazo.
- A classificação do programa terá em conta a qualidade da programação e a estrutura do código criado segundo os princípios da Programação Orientada por Objetos.
- Serão premiadas a imaginação e a criatividade.
- O projeto terá uma componente de **avaliação oral obrigatória** com classificação individual dos elementos do grupo.
- Os alunos que não comparecerem à discussão serão classificados com zero na fase respetiva. Nesta discussão será apurada a capacidade do aluno de produzir o código apresentado. Nos casos em que essa capacidade não for demonstrada, a nota atribuída será zero.
- A avaliação oral é realizada pelo respetivo professor de laboratório e irá ser feita uma marcação prévia para cada grupo de trabalho.

- Todos os projetos serão submetidos a um sistema automático de deteção de cópias. Os projetos que forem identificados como possíveis cópias, e verificando-se serem efetivamente cópias, serão anulados.

Critérios de Avaliação da 2ª fase

Esta segunda fase será avaliada segundo os seguintes critérios:

Funcionalidades	**30%**
Visualização das agendas e atendimentos	15%
Qualidade e detalhe da estratégia de visualização	15%

Implementação	**60%**
Restruturação e estrutura do projeto (não inclui alterações das coleções usadas)	10%
Estrutura e funcionamento interno das classes	10%
Utilização de coleções (**`ArrayList`**, **`HashSet`** e **`HashMap`**)	10%
Bom estilo (identificadores, comentários, indentação)	10%
Testes	10%
Documentação (JavaDoc e relatório)	10%

Avaliação qualitativa	**10%**
Qualidade geral, detalhes de implementação e funcionalidades adicionais	10%

A nota final será calculada da seguinte forma:

```
Nota final = 50% fase1 + 50%fase2
```