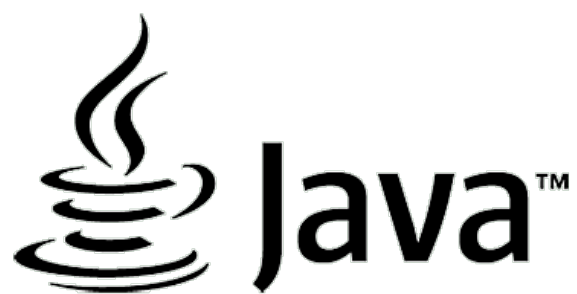


RELATÓRIO
Programação Orientada a Objetos
Escola Secundária
Grupo: 19

Autores:
Leonardo Gonçalves- 48710
Rui Soares- 46022
Professora: Paula Prata



1. Introdução

Foi-nos proposto pela Professora Paula Prata no Trabalho Final da disciplina de **Programação Orientada a Objetos**, a implementação do Sistema de uma Escola em Java, a linguagem de programação lecionada durante a Unidade Curricular. Dos vários tipos de Escola disponíveis, optámos pela **Escola Secundária**, pois, apesar de, à partida, ser mais trabalhosa, o grupo estava mais familiarizado com a estrutura base desta em relação às outras.

2. Interface da Aplicação

Passando agora à descrição da Aplicação criada, sem entrar ainda em código. O Utilizador terá o **Menu Inicial** disponível com uma mensagem de boas vindas à **Escola Secundária da Covilhã** (nome fictício), onde poderá, nesta ordem:

- Criar um Curso novo para a Escola;
- Listar os Cursos da Escola;
- Consultar um Curso através do nome ou número do mesmo;

E dentro desta opção terá várias opções relacionadas aos Alunos, Professores e Aulas do Curso, tais como: Criar, Listar, Consultar, Alterar, Apagar. Além destas, terá uma opção de ver Estatísticas, afim de poder ver vários dados disponíveis sobre o Curso (nº alunos, percentagens por género, entre muitas outras).

- Alterar os dados do Curso;
- Remover o Curso;

Em todas as opções terá a devida indicação dos dados que o Utilizador deve inserir, de modo a tudo funcionar corretamente.

3. Estrutura base

Agora sim, falaremos de todo o esqueleto em Java que foi desenvolvido, passando ao código propriamente dito. Iremos neste ponto descrever sucintamente todas as classes e funções presentes na aplicação, assim como os atributos e o funcionamento de cada um. Com o objetivo de uma compreensão mais acessível, estruturaremos a descrição referida por tópicos.

3.1. Pessoa

A superclasse **Pessoa** possui como atributos:

- Primeiro nome (*p_nome*): o primeiro nome da Pessoa.
- Último nome (*u_nome*): o último nome da Pessoa.
- Número de identificação fiscal (*nif*): o número de identificação fiscal da Pessoa.
- Data de nascimento (*data_nasc*): a data de nascimento da Pessoa.
- Sexo (*sexo*): o sexo da Pessoa.

Possui métodos *getters*, *setters*, *toString()*, *equals()*, e suporte para formatação de data, tudo devidamente implementado.

3.2. Aluno

A classe **Aluno**, além de herdar atributos da superclasse **Pessoa**, possui também:

- Número de aluno (*num_aluno*);
- Aulas (*aulas*): as aulas em que o aluno está inscrito.
- Turma (*turma*);
- Horário (*horário*).

Também possui métodos *getters*, *setters*, *toString()*, *equals()*, e além destes. Métodos para adicionar e remover aulas.

3.3. Prof

A classe **Prof**, além de herdar atributos da superclasse **Pessoa**, possui também:

- Número de professor (*num_Prof*);
- Disciplina lecionada (*cadeira*);
- Lista de aulas ministradas (*profaula*).

A classe também possui métodos *getters*, *setters*, *toString()*, *equals()*, e métodos adicionais para adicionar e remover aulas. A estrutura segue a mesma lógica da classe **Aluno**, promovendo coesão e reutilização de código.

3.4. Aula

Atributos:

- Hora (*hora*): a hora em que a aula começa.
- Minuto (*minuto*): o minuto em que a aula começa.
- Nome (*nome*): o nome da aula.
- Inscritos (*inscritos*): o número de alunos inscritos na aula.
- Professor (*professor*): o professor que irá lecionar a aula.
- Dia da semana (*dia*): o dia da semana em que a aula acontece.
- Código (*codigo*): o código atribuído à aula.

3.5 Curso

Atributos:

- Último (*ultimo*): o número do último curso criado (incrementando consoante a criação de um novo curso).
- Número (*num*): o número do curso.
- Nome (*nome*): o nome do curso.
- Turmas (*turmas*): as turmas do curso.
- Aulas (*aulas*): as aulas do curso.
- Alunos (*alunos*): os alunos do curso.
- Pessoas (*pessoas*): o total de pessoas (alunos + Profs) que o Curso possui.
- Prof (*professores*): os professores do curso.

3.6 Classes e Respetivas Funções

3.6.1 Classe FuncAula

- *criarAula*: lê e valida informações sobre uma nova aula a ser adicionada a um curso especificado e adiciona a aula à lista de aulas do curso.
- *listarAulas*: percorre a lista de aulas e imprime informações sobre cada aula.
- *consultarAula*: lê informações sobre uma aula específica e pesquisa a lista de aulas do curso em busca de uma aula com o mesmo código ou nome. Se encontrada, imprime informações sobre a aula.

- *inscreverAula*: lê e valida informações sobre um aluno que deseja inscrever-se em uma aula específica e adiciona o aluno à lista de alunos inscritos na aula.
- *atribuirAula*: esta função permite atribuir um Professor a uma aula.
- *alterarAula*: esta função permite alterar os atributos de uma aula como as horas, o código, o nome e o dia de semana.
- *apagarAula*: apaga uma aula, bem como os alunos inscritos e o Professor atribuídos à mesma.
- *removerAluno*: lê e valida informações sobre um aluno inscrito numa aula e remove o aluno da lista de alunos inscritos na aula.
- *removerProfessor*: esta função remove um Professor de uma dada aula, com isto também, claro, remove a aula da lista de aulas do Professor.

3.6.2 Classe FuncCurso

- *criarCurso*: lê informações sobre um novo curso a ser adicionado à lista de cursos e adiciona o curso à lista.
- *consultarCurso*: lê informações sobre um curso específico e pesquisa a lista de cursos em busca de um curso com o mesmo nome ou número. Se encontrado, imprime informações sobre o curso.
- *alterarCurso*: permite a atualização de informações sobre um dos cursos da respetiva lista.
- *apagarCurso*: permite a remoção de um curso da lista de cursos.
- *listarCurso*: percorre a lista de cursos e imprime informações sobre cada curso.

3.6.3 FuncAluno

- *criarAluno*: lê informações sobre um novo aluno a ser inscrito na Escola, e por consequência no Curso. **O Curso que o aluno escolhe e o ano de nascimento dele guiarão o aluno automaticamente a uma turma e respetivo horário.**
- *listarAluno*: percorre a lista de alunos de um curso e imprime informações sobre cada um deles.
- *consultarAluno*: lê informações sobre um aluno da lista de alunos, em busca de um com o mesmo nome ou número de aluno. Se encontrado, imprime informações sobre o aluno.
- *alterarAluno*: lê informações sobre um aluno e permite alterar as seguintes informações: nome, NIF, data de nascimento e sexo. As alterações são guardadas na lista de alunos.
- *apagarAluno*: permite a remoção de um aluno da Lista de Alunos do curso.

3.6.4 FuncProf

- *criarProf*: lê informações sobre um novo Professor a ser adicionado à lista de Professores e adiciona-o à mesma (de forma semelhante à criação do aluno).
- *alterarProf*: lê informações sobre um Professor e permite alterar os atributos do mesmo, como o nome, o NIF, a data de nascimento, o sexo e a disciplina que leciona. As alterações são então guardadas na lista de Professores do Curso.
- *listarProf*: percorre a lista de Professores e imprime informações sobre cada um.
- *consultarProf*: lê informações sobre um Professor específico e pesquisa o mesmo na lista de Professores, através do seu nome e/ou número. Se encontrado, imprime as informações sobre o Professor.
- *apagarProf*: permite a remoção de um Professor da respetiva lista.

3.6.5 FuncEstatisticas

- *numAlunos*: imprime o número total de alunos do Curso em questão.
- *alunosInscAula*: imprime o número de alunos inscritos em aulas de um curso e a percentagem dos inscritos em relação ao total de alunos.
- *alVerIdade*: imprime o número de aluno com X anos (X é o input do utilizador).
- *alunosGenero*: imprime o número de alunos do sexo masculino e do sexo feminino, com as respetivas percentagens.
- *numProfessores*: imprime o número de Professores.
- *idadeProfessores*: imprime o número de Professores com uma idade, introduzida pelo utilizador e a sua respetiva percentagem em relação ao total de Professores registados.
- *aulasProfessores*: indica o número de Professores que leciona uma aula ou mais.
- *generoProfessores*: indica o número de Professores do sexo masculino e feminino e a sua respetiva percentagem.
- *aulaDias*: imprime no ecrã o número de aulas por cada dia da semana e a sua respetiva percentagem.
- *aulasTotal*: imprime no ecrã o número de aulas existente (caso não existam, imprime que não existem).

4. Conclusão

Mesmo com as dificuldades normais da elaboração de um trabalho deste nível, acrescidas também ao facto de sermos apenas dois membros no grupo, a solução apresentada foi desenvolvida com relativo sucesso na nossa perspectiva. Dos pontos não tão positivos, apenas apontamos o facto de os Horários das Turmas não estarem totalmente mecanizados com a marcação das Aulas. Em tudo o resto, sentimos que cumprimos com as nossas expetativas, pois além da solução estar bem desenvolvida, cumprimos estritamente todas as condições propostas no Enunciado, e esse rigor foi a chave para a correta construção da aplicação. Saímos da realização deste trabalho mais conhecedores da **Programação Orientada a Objetos** e dos princípios que a estruturam.



Departamento de
Informática