Nome: Breno Guedes da Silva – 10425218

Matheus Tramont Hoffmann – 10425295

Sérgio de Souza Melo – 10425206

Relatório do Projeto de Programação de Sistemas

Classes

Classe Pessoa:

A classe Pessoa foi criada para poder criar os atributos de nomePessoa, nascimento, cep e telefone. Criamos também o construtor Pessoa para que ele possa ser herdado pelas classes herdeiras, utilizando os atributos mencionados. Os métodos getters são utilizados para permitir que os valores dos atributos privados da classe Pessoa sejam acessados.

Subclasse Estudante:

Esta classe herda a classe Pessoa.

Ele importa as classes ArrayList e List do pacote java.util.

Esta classe serve para criar atributos, estes sendo o RA, email, curso, semestre, semestreconclusão, este sendo o atributo extra, e os codigosOferecimento que lista o código das disciplinas em que o estudante está matriculado.

Com esta classe, o método construtor Estudante é criado, em que é inicializado os atributos da classe e também chama a classe pai Pessoa, usando super().

Além disso, ele também tem métodos getters como maneira de retornar os valores dos atributos privados.

Além do mais, ele possui dois métodos para calcular o semestre de conclusão.

- CalculaSemestreConclusão: Calcula o semestre de conclusão baseado no semestre atual, sendo este, o método extra do trabalho.
- CalculaAnoConclusão: Calcula o ano de conclusão baseado no semestre e no ano atuais. Este também é mais um método extra do nosso trabalho.

Além disso, ele possui o método adicionarCodigoOferecimento para verificar se o código que receber já está na lista ou não.

Subclasse Professor:

Esta classe herda a classe Pessoa.

Ele importa as classes ArrayList e List do pacote java.util.

Esta classe também cria atributos, estes sendo o DRT, email, unidadeAcademica e disciplinasOferecimento que lista o código de oferecimento das disciplinas em que o professor ministra a aula.

A classe também cria o construtor Professor, em que inicializa os atributos da classe e também chama a classe pai Pessoa, usando super().

Assim como na subclasse Estudantes, ele cria métodos para retornar os valores dos atributos privados.

Parecido com a subclasse Estudantes, ele também possui um método que adiciona um código de oferecimento na lista de disciplinasOferecimento, validando se o código que receber já está ou não na lista.

Classe Disciplina

Ele importa as classes ArrayList e List do pacote java.util.

Esta classe possui uma classe herdeira, este sendo a classe Oferecimento.

Esta classe cria os atributos codigosdisciplina, nome, unidadeAcademica, curso e oferecimentos, este último sendo uma lista que representa as ofertas da disciplina.

O construtor Disciplina faz com que seja inicializado os atributos da classe.

- codigodisciplina, nome, unidadeAcademica e curso são inicializados com os valores passados como parâmetros.
- oferecimentos é inicializado como uma nova ArrayList vazia.

Ele também possui métodos getters para retornar os valores dos atributos privados.

Ele possui um método para listar os oferecimentos, dando um print para cada detalhe que ele conseguir de getNome, getCodigoDisciplina, getSemestre e getProfessorResponsavel da subclasse Oferecimento.

Por fim, ele possui um método que adiciona um objeto Oferecimento na lista de oferecimentos, verificando se o oferecimento já está na lista ou não.

Subclasse Oferecimento

Esta classe herda a classe Disciplina.

Ele importa as classes ArrayList e List do pacote java.util.

Esta classe cria os atributos codigooferecimento, semestre, professorResponsavel, estudantesMatriculados, sendo o último uma lista do RA dos estudantes matriculados em oferecimento.

Ele cria o construtor Oferecimento que inicializa os seus atributos e também ao utilizar o super(), ele também herda do construtor da classe Disciplina.

Ele também possui métodos getters para retornar os valores dos atributos privados.

Também possui um método chamado matricularEstudante. Este método serve para adicionar um RA de estudante na lista de estudantesMatriculados, verificando se o RA está ou não na lista.

Classe Listas

Ele importa as classes ArrayList e List do pacote java.util.

Ele cria três atributos que são listas para classe Listas:

- Estudantes: Lista de objetos Estudante.
- Professores: Lista de objetos Professor.
- Disciplinas: Lista de objetos Disciplina.

Ele define os atributos Estudantes, Professores e Disciplinas como listas novas e vazias, depois, cria um método chamado adicionarEstudante, assim ele adiciona o objeto Estudante para lista Estudantes. Depois, ele obtém a lista de estudantes criando o método getEstudantes. Isso vale

também para Professores, com o método adicionarProfessor e também para Disciplinas com o método adicionarDisciplina.

Ele cria um método booleana chamada buscaEstudante, em que ele procura um Estudante através do RA. Caso ele encontre, ele vai dar print com os detalhes do estudante e retorna true. Caso contrário, ele dá o print "Estudante não encontrado" e retorna false.

A classe possui um método para listar todos os estudantes, chamado listaEstudantes, em que ele imprime os detalhes de todos os estudantes na lista Estudantes.

A classe possui um método para também listar todos os professores, chamado listaProfessores, em que ele imprime os detalhes de todos os professores na lista Professores.

Ele também possui o método listaEstudantesMatriculados em que ele faz uma listagem de todos os alunos que estão matriculados naquela disciplina, utilizando um for para mostrar cada objeto Disciplina na lista Disciplinas, e ele sempre verifica através do if para saber se o código da disciplina atual de disciplina.getCodigoDisciplina() é igual ao codigoDisciplina. Logo após ele, também há outro método chamado listaDisciplinas(), que lista todas as disciplinas, imprimindo seus detalhes.

A classe também possui mais um método chamado de buscaOferecimento, em que ele busca um oferecimento através do código da disciplina e do código do oferecimento, e logo após isso, ele dá um print dos detalhes que ele encontra.

Ele possui um método de listaOferecimento que lista o oferecimento específico em que os estudantes estão matriculados, dado o código da disciplina e o código de oferecimento.

Por fim, ele possui um método buscaDisciplina que procura uma disciplina na lista Disciplinas com base no código fornecido.

Classe Main

A classe main primeiramente cria uma instância da classe Listas, em que é usado para gerenciar e armazenar listas de estudantes, professores e disciplinas.

Após isso, quatro estudantes e dois professores são criados com seus respectivos dados e adicionados à lista de estudantes e na lista de professores da classe Listas, respectivamente. Depois, quatro disciplinas são criadas e adicionadas na lista de disciplinas da classe Listas. Logo depois disso, três oferecimentos de disciplinas são criados e adicionados em suas respectivas disciplinas e os estudantes são matriculados nos oferecimentos correspondentes.

Depois disso, os códigos de oferecimento são adicionados aos estudantes para poder registrar em quais oferecimentos os estudantes estão matriculados. Em sequência, o oferecimento é adicionado ao professor responsável.

Por fim, ele faz a chamada de vários métodos para listar as informações sobre estudantes, professores, disciplinas e oferecimentos. Essas chamadas demonstram as funcionalidades implementadas das classes:

- buscaEstudante: Busca e exibe informações de um estudante específico.
- buscaProfessor: Busca e exibe informações de um professor específico.
- buscaDisciplina: Busca e exibe informações de uma disciplina específica.
- buscaOferecimento: Busca e exibe informações de um oferecimento específico.
- listaEstudantes: Lista todos os estudantes.

- listaProfessores: Lista todos os professores.
- listaDisciplinas: Lista todas as disciplinas.
- disciplina1.listaOferecimentos: Lista todos os oferecimentos de uma disciplina específica.
- listas.listaOferecimento: Lista um oferecimento específico e os estudantes matriculados nele.

Outras Decisões do Projeto:

Classe Listas

A classe listas é uma classe extra que foi feita para organizar de maneira mais eficiente o código ao agrupar todas as listas extras de objetos e métodos que lidam com ela em uma classe só, dessa maneira, tornando-se desnecessário precisar fazer isso no main.

Atributo e Método Extra na Classe Estudantes

A ideia do método extra foi inventar algo que pudesse ser interessante, desta maneira, uma das ideias que surgiu foi calcular o ano e o semestre de conclusão e foi essa mesma ideia que foi aplicada no projeto. Também criamos um atributo extra para fazer isso funcionar, este sendo semestreconclusão.

Teste de Execução

Em seguida, está o nosso teste de execução do código:

```
Nome: Matheus
Curso: Sistemas de Informação
Semestre em que ingressou: 2023.2
Semestre de conclusão previsto: 2027.1
Ano de nascimento: 2004
CEP: 12345-67
Email: 1@mackenzista.com.br
Número de telefone: (11) 99782-6139
Códigos de oferecimentos em que está matriculado: [1, 2, 3, 4]

Nome: André
Unidade: FCI
Ano de nascimento: 1990
CEP: 12345-67
Email: 1@mackenzie.br
Número de telefone: (11) 12345-6789
Códigos de oferecimentos sob sua responsabilidade: [1]

Disciplina: Inteligência Artificial
Unidade: FCI
Curso: Sistemas de Informação

Disciplina: Banco de Dados
Unidade: FCI
Curso: Sistemas de Informação
Semestre: 2024/01
DRT do professor responsável: 2

Nome: Matheus
RA: 1
Curso: Sistemas de Informação
Nome: Sérgio
RA: 2
Curso: Sistemas de Informação
Nome: Breno
RA: 3
Curso: Sistemas de Informação
Nome: Julio
RA: 4
Curso: Sistemas de Informação

Nome: André
DRT: 1
Nome: Thiago
DRT: 2
```

```
Disciplina: Programação
Unidade: FCI
Código: 1
Disciplina: Banco de Dados
Unidade: FCI
Código: 2
Disciplina: Inteligência Artificial
Unidade: FCI
Código: 3
Disciplina: Redes de Computadores
Unidade: FCI
Código: 4

Disciplina: Programação
Código: 1
Semestre: 2024/01
DRT do professor responsável: 1

Disciplina: Banco de Dados
Semestre: 2024/01
Professor: 2
Alunos matriculados:
Nome: Matheus
RA: 1
Nome: Sérgio
RA: 2
Nome: Breno
RA: 3
Nome: Julio
RA: 4
```