Relatório do trabalho de Orientação a Objetos

UFJF

Deiverson Mourão Alves Pedroso (201965123A)
Deyvison Gregório Dias (201835017)
Pedro Henrique Almeida Cardoso Reis (201835039)

Juiz de Fora, 17 de Março de 2021

Contents

1 Introdução

Neste relatório será apresentando o projeto final do trabalho de Orientação a Objetos(DCC025). Para esse trabalho foi utilizado a linguagem Java e o projeto final consiste em um sistema para auxílio na administração de uma rede de academias.

2 Informações Relevantes

• Para fazer a compilação do projeto, foi utilizado o Maven.Para isso, entre na pasta SistemaDeAcademia e depois execute o seguinte comando:
java -jar target-1.0-SNAPSHOT-jar-with-dependencies.jar

• Para acessar o repositório do GitHub acesse o seguinte link: (`https://github.com/PHenriqueCEC/Trabalho-oo`)

• Após a compilação do projeto é aberta a tela do Login. Com essa tela tanto o administrador quanto o instrutor e cliente poderão efetuar Login ao inserir o nome de usuário e senha. Após a realização do Login cada usuário poderá realizar diferentes tarefas. Por exemplo, o cliente pode conferir o status de pagamento, o instrutor pode cadastrar uma nova ficha e o administrador pode cadastrar um novo cliente.

• O sistema foi desenvolvido pensando em quatro tipos de usuários:

• *Administrador:* Responsável por cadastrar e remover um funcionário(recepcionista) e um instrutor no sistema;
• *Recepcionista:* Responsável por cadastrar um novo cliente(aluno) e receber o pagamento do cliente;
• *Instrutor:* Responsável por criar a ficha do cliente(aluno) e alterá-la;
• *Cliente(aluno):* Pode acessar o sistema para visualizar a ficha de treino e verificar vencimento/status da mensalidade;

• Foram feitas algumas reuniões e pesquisas sobre quais estruturas de dados seriam melhores para o projeto. Com base nas referências encontradas, optamos por estruturar nosso código por meio de

várias classes onde definimos em cada uma delas sua funcionalidade. Para o nosso sistema foi utilizado um ArrayList para armazenar os dados dos clientes, aparelhos e fichas. Embora a inserção em uma ArrayList seja lenta, optamos por usar essa estrutura de dados pois o acesso ao seus elementos é feito de maneira rápida.

3 Organização das Classes e Pacotes

Após algumas reuniões foi decidido organizar as classes e pacotes de forma bem definida para melhor entendimento e organização do código. Diante disso desenvolvemos o nosso projeto utilizando três pacotes: *Interfaces e Objetos*

3.1 Pacote Interfaces

O pacote *Interfaces* foi utilizado para incluir todas as interfaces utilizadas no sistema. As interfaces são:

- *CadastroDeCliente*- Interface utilizada para cadastrar o cliente;
- *ControleDeAdms* - Interface utilizada para adicionar e remover um administrador;
- *CriarFicha* - Interface utilizada para criar uma nova ficha;
- *FichaDoCliente* - Interface que permite o cliente visualizar sua ficha;
- *Instrutor* - Interface que permite o instrutor alterar a ficha de determinado cliente;
- *JanelaDeAdmn* - Interface que permite o administrador cadastrar um novo cliente;
- *JanelaDeCliente* - Interface que permite o cliente acessar a ficha de treino, verificar o valor da mensalidade e olhar a data de vencimento;
- *Login* - Interface utilizada para o Administrador, Instrutor e Cliente fazer Login;
- *NovoAdm* - Interface utilizada para cadastrar um novo Administrador;
- *Pagamento* - Interface que permite procurar determinado Cliente e receber o pagamento;
- *Recepcao* - Interface que permite o Administardor acessar a Inter-

face de pagamento e de cadastro;
• *RemoverFunc* - Interface utilizada para remover um funcionário;
• *RemoverAdm* - Interface utilizada para remover um Administrador;
• *Treinador* - Interface que permite o treinador editar a ficha do cliente;
Vale ressaltar ainda que essas interfaces foram colocadas dentro de pastas, nomeando-as com nomes bem intuitivos.
Na pasta *Administrador* ficarão as interfaces *ControleDeFunc, novoFunc e removerFunc.* Já na pasta *Clientes* temos a interface *JanelaDeCliente e FichaDoCliente.*E, por fim, na pasta *Funcionários* teremos duas sub-pastas que são: *Instrutor e Recepcionista.* Na primeira nós teremos as interfaces de *Instrutor e CriarFicha* e na segunda teremos as interfaces de *Recepcao, Pagamento e CadastroCliente*

3.2 Pacote Objetos

O pacote *Objetos* contém as principais classes desenvolvidas para o sistema, que são:

• *Administrador* - Classe responsável por criar o Administrador do sistema;
• *Cliente* - Classe responsável por criar o cliente. Além disso, a classe Cliente é uma superclasse das classes ClienteAnual, ClienteSemestral e ClienteTrimestral;
• *Ficha* - Classe responsável por criar a Ficha do Cliente;
• *Instrutor* - Classe responsável por criar um instrutor para a academia;
• *Pagamento* - Classe utilizada para que o Cliente possa fazer o pagamento da mensalidade;
• *Recepcionista* - Classe responsável por criar um recepcionista, além de receber o pagamento do cliente;
• *Main* - Classe principal do sistema. Após a compilação do projeto esta classe abre a tela de Login, dando início a execução do sistema.

4 Conceitos utilizados

Para ter um melhor aproveitamento no desenvolvimento do trabalho, foram utilizados no desenvolvimento do projeto alguns conceitos abordados no decorrer da disciplina(herança, polimorfismo, classes

abstratas, interfaces, coleções, tratamento de exceções, leitura e escrita em arquivos, e interface gráfica). Para ter um maior entendimento de como o código foi desenvolvido criamos algumas subseções para explicar a forma de como usamos os conceitos trabalhados na disciplina em nosso projeto.

4.1 Herança

A herança permite criar novas classes a partir de classes já existentes, aproveitando-se das características existentes na classe a ser estendida. Com base nisso as classes *ClienteAnual, ClienteSemestral e ClienteTrimestral* herdam os atributos da classe *Cliente*.

4.2 Polimorfismo

É o princípio pelo qual duas ou mais classes derivadas de uma mesma superclasse podem invocar métodos que tem a mesma identificação (assinatura) mas com comportamentos distintos. Na classe *Cliente* podemos instanciar um novo cliente definindo ele como anual, semestral ou trimestral.

4.3 Interface

Interface é um tipo que define um conjunto de operações que uma classe deve implementar. A interface estabelece um contrato que a classe deve cumprir.
Em nosso sistema nós usamos a interface em Pagamento, onde cada tipo de Cliente implementa a interface relativa ao seu método de pagamento.

4.4 Coleções

Uma coleção é uma estrutura de dados que permite armazenar vários objetos. Em Java, as principais coleções são as *Lista, Coleções e Dicionário*
No sistema desenvolvido usamos as seguintes Coleções:

- *List* para a classe *Clientes*;
- *List* para a classe *Ficha*;
- *Map* para a classe *Main*;

4.5 Tratamento de Exceções

O tratamento de exceção é responsável pelo tratamento da ocorrência de condições que alteram o fluxo normal da execução de determinado programa. Na linguagem Java, os comandos *try* e *catch* são responsáveis por tratar as exeções. Para o desenvolvimento do nosso sistema utilizamos os seguintes tipos de exeções:

• *IOException:* para tratar erros de entrada e saída;
• *ParseException:* sinaliza que um erro foi alcançado inesperadamente durante a análise.

4.6 Leitura e Escrita em Arquivos

A leitura e escrita de arquivos por um programa pode ser muito útil pois ela serve simplesmente como uma entrada e saída de dados. Em nosso projeto, trabalhamos a leitura e escrita na Classe *Ficha*, nos tipos de *Clientes* , e nos *funcionarios*. A classe *Ficha* é responsável por fazer a escrita da ficha do aluno dentro de um arquivo txt além de escrever todos os dados dos *Funcionarios* e *Clientes*. Além disso, os dados das classes *Funcionario* e *Cliente* são lidas no nosso sistema a partir de um arquivo txt.

4.7 Interface Gráfica

Interface Gráfica é um conceito da forma de interação entre o usuário do computador e um programa por meio de uma tela. Em nosso projeto as interfaces estão no Pacote *Interfaces*. Nesse pacote encontra-se todas as telas de interação do usuário com o sistema