

Pontifícia Universidade Católica – Escola Politécnica
Curso de Engenharia de Software – Semestre 2
Disciplina de Programação Orientada a Objetos
Professor Marcelo Hideki Yamaguti

RELATÓRIO DE PROJETO – ACMETech

Autores do projeto:

- Carlos Ramiro Trejes Dornelles
- Thiago Tarantino
- Isadora Spode Cardoso

1 . Diagrama de Classes

O diagrama de classes do sistema pode ser encontrado no arquivo “Class Diagram0-POOTF.asta” (sem aspas) que está incluído dentro da pasta resources do projeto, pasta a qual foi utilizada para guardar tudo que é vinculado ao sistema e não é um arquivo Java. Nessa pasta colocamos o diagrama de classes(Class Diagram0-POOTF.asta), alguns assets que colocamos para a execução, arquivos de texto padrão para leitura e arquivos de carregamento e/ou salvos via sistema.

2 . Coleções de dados

Ao longo do desenvolvimento do software, vimos a necessidade de utilizar coleções da biblioteca padrão do Java para organizar, armazenar e manipular os dados da aplicação. Com base em conhecimentos aprendidos em sala de aula na disciplina de Programação Orientada a Objetos e na disciplina de Algoritmo e Estrutura de Dados I, escolhemos as seguintes estruturas de dados:

2.1 . TreeSet<>

Todos os dados cadastrados das entidades foram cadastrados em um TreeSet<> por dois motivos: preservar ordenação dos elementos por código e evitar duplicatas de código, utilizando os métodos compareTo() das classes de cada entidade.

2.2 . HashMap<>

Um HashMap<> foi utilizado para consultar comprador com maior vendas. O método que o implementa precisa contar quantas vezes cada objeto Comprador aparece na lista de vendas. Devido a isso, vimos a necessidade de utilizar o HashMap<> por ele associar uma entidade única via chave a um valor acumulado, fazendo, assim, a contagem de vendas.

3 . Armazenamento (Persistência) de Dados – Salvar e Carregar

A aplicação inicia carregando dados de arquivos .csv de alguns exemplos que o professor nos encaminhou no .PDF do trabalho. Além disso, a aplicação disponibiliza Salvar dados via uma janela que pede ao usuário o nome do arquivo que o mesmo deseja salvar. A aplicação, então, cria ou subscreve o arquivo de nome que o usuário digitou (ex.: NOMEDOARQUIVODOUSUARIO.json) em formato .JSON na pasta resources. Caso o arquivo ainda não exista, o mesmo é criado ao salvar.

O processo de carregar dados acontece de maneira similar ao de salvar dados: o usuário digita o nome do arquivo que deseja carregar da pasta resources. A aplicação abre o arquivo com o nome digitado (ex.: NOMEDOARQUIVODOUSUARIO.json) em formato .JSON na pasta resources. Caso o arquivo não exista, informa ao usuário com uma mensagem de erro.

Como dito previamente, os arquivos de persistência ficam armazenados no diretório resources do projeto, facilitando a localização e permitindo que o sistema carregue dados

automaticamente de arquivos base ao iniciar. Se o usuário desejar, pode carregar dados de um outro arquivo e, quando o faz, todos os dados prévios são apagados e substituídos pelos dados desse arquivo que o usuário carregou.