Universidade de Brasília – 2.2018
117889 - Técnicas de Programação 1 – Turma A
Professor Christian Riaño

Trabalho Prático 1

José Antonio Siqueira de Cerqueira : 18/0111566
Danilo : 17/0140008

Relatório:

Realizamos a implementação do sistema de **Cinemas** proposto. Inicialmente geramos os diagramas do modelo conceitual. São eles: Diagrama de Classes, Diagrama de Objetos, Diagrama de Caso de Uso e Diagrama de Sequência, para o sistema de Cinemas.

Para a implementação utilizamos a IDE Eclipse 4.9.0 em um Sistema Operacional Debian. Também fizemos uso de um sistema de versionamento chamado Git, junto com o repositório remoto Github, para fins de colaboração entre a dupla. Tal link para o repositório é :
https://github.com/josesiqueira/ControleCinema

Criamos os pacotes dados, domain, excecoes, run e transacoes:

- No pacote dados estarão as classes e interfaces que tratam de acesso aos dados, que em nosso sistema é feito através de arquivos de texto.
- No pacote domain estão as classes que representam as entidades do diagrama de classes, são elas Filme, Sala e Sessao.
- No pacote excecoes, que estão as classes que implementam os erros que são percebidos na execução do código. Aqui foi utilizado o conceito de extends, quando a classe AcessoDadosEx extends Exception, por exemplo.
- No pacote run há a classe executora CinemaTeste, que funciona como a main do nosso programa. Nela há um menu onde o usuário – Funcionário de um Cinema - poderá escolher qual tipo de serviço, aqueles descritos no Diagrama de Caso de Uso, irá usar.

Aqui cabe frisar um conceito visto em aula:

```
private static final CatalogoSessoes catalogoSession = new CatalogoSessoesImpl();
```

é uma variável privada e estática, do tipo CatalogoSessoes, ou seja, eh do tipo desta classe, com o nome catalogoSession, e ela será um construtor da classe CatalogoSessoesImpl. Ou seja, a variável catalogoSession é do tipo CatalogoSessoes e é um objeto da classe CatalogoSessoesImpl.

- No pacote transações há as interfaces CatalogoFilmes, CatalogoSalas, e CatalogoSessoes. Estas interfaces possuem métodos abstratos, ou seja, sem corpo. Estes métodos devem ser implementados nas classes CatalogoFilmesImpl, CatalogoSalasImpl, CatalogoSessoesImpl. O conceito utilizado aqui foi de herança, polimorfismo, entre outros. Onde as classes que possuem apenas métodos abstratos são interfaces (superclasse), e as classes que implementam seus métodos são as subclasses, e são essas as classes que geram instâncias, ou seja, objetos. As classes concretas podem implementar métodos da superclasse (interface) com o mesmo nome do método através do conceito de polimorfismo.

Superclasse interface que possui apenas métodos abstratos

```
public interface CatalogoFilmes {
```

Subclasse que implementa a interface:

```
public class CatalogoFilmesImpl implements CatalogoFilmes {
```

Para trabalho futuro: tratamento de exceções mais específicias, tratamento dos dados de forma persistente através de um banco de dados, novas funcionalidades como a classe Ingresso.