



2º Trabalho¹

Acesso a Dados

Quinta-feira, 12 de novembro de 2020.

O objetivo desse trabalho é criar uma camada de acesso a dados composta por **interfaces e classes genéricas** que forneçam, dentre outros, serviços de criação, consulta, atualização e destruição de dados em arquivos ou banco de dados relacionais.

Essas quatro operações básicas são conhecidas como CRUD, sigla em inglês para *Create* (criar, inserir), *Read* (ler, recuperar), *Update* (atualizar) e *Delete* (excluir, apagar).

É fundamental que os serviços sejam especificados para funcionar independente do tipo de implementação usada para acessar a fonte de dados (arquivos ou banco de dados). Isso requer que os métodos, especialmente das interfaces, devem ter uma assinatura e valor de retorno que atenda a qualquer fonte de dados. Veja o exemplo abaixo.

```
// Obtém uma conexão ao abrir o arquivo cursos.dat.
```

```
Arquivo arquivo = Conexao.iniciarConexao("cursos.dat");
```

```
// Inicia uma conexão para acessar o banco de dados cursos.
```

```
BD bd = Conexao.iniciarConexao("jdbc:postgresql:cursos", usuario, senha);
```

As operações que devem ser fornecidas por essa camada de acesso a dados são:

1. Iniciar ou abrir uma conexão com a fonte de dados.
2. Encerrar ou fechar uma conexão com a fonte de dados.
3. Inserir um objeto na fonte de dados.
4. Pesquisar um objeto usando o método de comparação natural de sua classe para identificá-lo.
5. Atualizar um objeto usando o método de comparação natural de sua classe para identificá-lo.
6. Excluir um objeto usando o método de comparação natural de sua classe para identificá-lo.
7. Recuperar um objeto usando o método de comparação natural de sua classe para identificá-lo.
8. Obter o número de objetos armazenados na fonte de dados.
9. Obter uma lista de objetos com todos os objetos armazenados na fonte de dados.

Sinta-se à vontade para adicionar qualquer método na camada de acesso a dados que possa enriquecer sua funcionalidade.

Para testar a camada de acesso a dados use a classe `ArquivoObjeto` (pacote `tsi.too.cap13.arquivo.serializacao`) do Projeto Java 15 - Arquivos para criar uma classe abstrata genérica de implementação dos métodos acima. O objetivo dessa classe é testar a especificação dos

¹ Atualizado em 13 de novembro de 2020.

serviços (métodos) de acesso a uma fonte de dados usando serialização de objetos. No entanto, é necessário que a classe `ArquivoObjeto` seja reescrita como uma classe genérica.

Após desenvolver a classe de implementação, use qualquer objeto, por exemplo, o objeto `Produto` do 1º Trabalho Gestão de Produção para serializar os seus atributos em um arquivo.

Observe que pela descrição acima, o projeto Java desse trabalho deve ser composto por quatro tipos de dados:

- a) A interface genérica que especifica os serviços de acesso a dados das nove operações descritas acima;
- b) Uma classe abstrata de implementação da interface, ambas genéricas, para fornecer acesso a uma fonte de dados de serialização de objetos por meio da classe genérica `ArquivoObjeto`;
- c) A refatoração da classe `ArquivoObjeto` transformando-a em uma classe genérica;
- d) Uma classe de teste dos serviços de acesso a dados da classe de implementação para mostrar o seu funcionamento.

- Critérios de Avaliação

1. O trabalho será avaliado considerando:
 - a. a validação dos dados fornecidos pelo usuário;
 - b. o uso de tratamento de exceções;
 - c. a lógica empregada na solução do problema;
 - d. o funcionamento do programa;
 - e. a usabilidade² do programa;
 - f. o conhecimento da linguagem de programação Java;
 - g. a implementação dos conceitos de orientação a objetos;
 - h. o uso do princípio do menor privilégio;
 - i. código fonte sem erros e sem advertências do compilador;
 - j. código fonte legível, indentado, organizado e comentado;
 - k. identificadores significativos para aprimorar a inteligibilidade do código fonte;
 - l. pacotes significativos para organização de tipos de dados;
 - m. a documentação do programa gerada com *javadoc*.
2. O programa deve ser desenvolvido integralmente usando apenas os recursos da linguagem Java 8 ou superior. Programas desenvolvidos em outras linguagens, mesmo que parcialmente, receberão nota zero.
3. O ambiente de desenvolvimento deve ser o *Eclipse IDE for Java Developers Release 2020-03* (versão 4.15) ou 2020-09 (versão 4.17).
4. Para que o programa seja avaliado o código deve executar com sucesso. Programas que apresentarem erros de compilação receberão nota zero.

² A usabilidade está diretamente ligada a interface e a capacidade do *software* em permitir que o usuário alcance suas metas de interação com o sistema. Ser de fácil aprendizagem, permitir uma utilização eficiente e apresentar poucos erros, são os aspectos fundamentais para a percepção da boa usabilidade por parte do usuário. **Fonte:** <http://pt.wikipedia.org/wiki/Usabilidade>

5. Trabalhos com plágio, ou seja, programas com código fonte copiados de outra pessoa (cópia integral ou parcial) receberão nota zero.
6. O desenvolvimento do trabalho é individual.

- Artefatos de Software

Os seguintes artefatos do programa devem ser entregues:

- a) o projeto Java desenvolvido na IDE Eclipse;
- b) o arquivo JAR executável;
- c) a documentação em HTML gerada com *javadoc*.

Compacte todos os artefatos criando um arquivo *.Zip* com o seu nome e sobrenome e poste no SIGAA na tarefa correspondente a esse trabalho.

- Data de entrega

Segunda-feira, 23 de novembro de 2020.

- Valor do trabalho

20,0 pontos.

Prof. Márlon Oliveira da Silva
marlon.silva@ifsudestemg.edu.br