

Universidade Presbiteriana Mackenzie
Faculdade de Computação e Informática

Lista de Exercícios 1

NOTAS DE AULA - Teoria 11

Linguagem de Programação II
2º semestre de 2015
Versão 1.0



PARTE 1 - Controle de Gastos

Você precisa realizar diversas tarefas como parte do desenvolvimento de um sistema de controle de gastos.

Exercício 1

As informações dos gastos do mês são armazenadas em um arquivo-texto chamado **gastos-mes-atual.txt**, onde cada linha corresponde a um gasto e possui o formato:

[descrição]:[categoria]:[valor em reais]

Exemplo:

```
Gasolina posto Esso:CARRO:120
Almoço no quilo do Zé:ALIMENTAÇÃO:23
Filme Cinderela no Eldorado:LAZER:50
Aluguel:MORADIA:1500
```

Escreva um programa em Java que leia o arquivo e apresente na tela do computador todos os gastos no formato:

[categoria] - R\$ [valor em reais] - [descrição]

Exemplo:

```
CARRO - R$ 120 - Gasolina posto Esso
ALIMENTAÇÃO - R$ 23 - Almoço no quilo do Zé
LAZER - R$ 50 - Filme Cinderela no Eldorado
MORADIA - R$ 1500 - Aluguel
```

Ao final da listagem, o programa deverá apresentar o valor total dos gastos.

Exercício 2

Continuando a aplicação de controle de gastos, desenvolva um programa em Java que leia o arquivo especificado no exercício 1 e salve os dados dos gastos utilizando o recurso de **serialização**.

Para isso:

- a) Declare a classe **Gasto** com os atributos, métodos e o que mais for necessário para que suas instâncias possam ser serializadas.
- b) Escreva um programa em Java que leia o arquivo **gastos-mes-atual.txt** (conforme especificado no Exercício 1), crie as instâncias de **Gasto** correspondentes e serialize estes objetos para o arquivo **gastos-mes-atual.dat**.

Exercício 3

- a) Escreva um programa em Java que permita a inserção das informações de um novo gasto no arquivo **gastos-mes-atual.txt**. O programa deverá solicitar as informações do gasto (descrição, categoria e valor) e adicionar uma nova linha no arquivo com os dados fornecidos.
- b) Escreva um programa em Java que permita a inserção das informações de um novo gasto no arquivo **gastos-mes-atual.dat** (definido no Exercício 2). O programa deverá solicitar as informações do gasto (descrição, categoria e valor), criar uma instância de **Gasto** (conforme definida no Exercício 2) e serializar o novo objeto para o arquivo.

Exercício 4

Escreva um programa em Java que permita a leitura dos gastos que estão serializados no arquivo **gastos-mes-atual.dat** (definido no Exercício 2). O programa deverá ler os objetos serializados e apresentar na tela do computador todos os gastos no formato:

[categoria] - R\$ [valor em reais] - [descrição]

Exemplo:

CARRO - R\$ 120 - Gasolina posto Esso
ALIMENTAÇÃO - R\$ 23 - Almoço no quilo do Zé
LAZER - R\$ 50 - Filme Cinderela no Eldorado
MORADIA - R\$ 1500 - Aluguel

Ao final da listagem, o programa deverá apresentar o valor total dos gastos.

Exercício 5

Haverá também a necessidade de ler e escrever as informações dos gastos em uma tabela na base de dados. O programa abaixo permite a leitura dos dados da tabela criada para armazenar os gastos:

```
public class AppSelectGastos {  
    public static void main(String[] args)  
        throws ClassNotFoundException, SQLException {  
        Class.forName("org.apache.derby.jdbc.ClientDriver");  
        Connection conexao;  
        String url = "jdbc:derby://127.0.0.1:1527/banco";  
        String usuario = "app";  
        String senha = "app";  
        conexao = DriverManager.getConnection(url, usuario, senha);  
        PreparedStatement st;  
        String sql = "SELECT id, descricao, categoria, valor FROM gastos";  
        st = conexao.prepareStatement(sql);  
        ResultSet resultados = st.executeQuery();  
        System.out.println("Dados dos gastos:");  
        while (resultados.next()) {  
            System.out.println("id: " + resultados.getLong("id"));  
            System.out.println("Descrição: " + resultados.getString("descricao"));  
            System.out.println("Categoria: " + resultados.getString("categoria"));  
            System.out.println("Valor: R$ " + resultados.getLong("valor") + "\n");  
        }  
        conexao.close();  
    }  
}
```

Exercício 5 (cont.)

Escreva um programa que permita inserir um novo registro na tabela de gastos. Seu programa deverá solicitar que o usuário forneça as informações do novo gasto e, a seguir, criar um novo registro na tabela com as informações fornecidas.

Exercício 6

Vamos encapsular as operações de persistência. Para isso, vamos definir a interface **GastoDaoInterface**:

```
public interface GastoDaoInterface {  
    List<Gasto> lerTodos();  
    void adicionar(Gasto gasto);  
}
```

Com base no que foi realizado nos exercícios anteriores:

- a) Implemente esta interface em uma classe que realize as operações persistindo as informações no arquivo-texto **gastos-mes-atual.txt**.
- b) Implemente esta interface em uma classe que realize a serialização das instâncias de **Gasto** no arquivo **gastos-mes-atual.dat**.
- c) Implemente esta interface em uma classe que realize as operações de persistência na tabela **gastos** da base de dados.

Exercício 7

- a) Altere a classe **AppSelectGastos** (cujo código foi fornecido no Exercício 5) de forma a utilizar o DAO desenvolvido no item (c) do exercício 6. Com esta alteração, a classe **AppSelectGastos** não deverá mencionar explicitamente nenhuma classe da API JDBC.
- b) Que modificações seriam necessárias nesta nova versão de **AppSelectGastos** se quiséssemos utilizar a persistência no arquivo-texto **gastos-mes-atual.txt** em lugar da persistência na base de dados?
- c) Que modificações seriam necessárias nesta nova versão de **AppSelectGastos** se quiséssemos utilizar a persistência no arquivo serializado **gastos-mes-atual.dat** em lugar da persistência na base de dados?

Exercício 8

Você precisa desenvolver uma aplicação para efetuar donwloads de arquivos.

Sua aplicação deverá apresentar um menu inicial com as seguintes opções:

MENU INICIAL:

- (1) Fornecer informações do download.
- (2) Finalizar

Se o usuário escolher a opção (1), sua aplicação deverá solicitar a URL do arquivo a ser baixado e o nome da pasta local onde ele deverá ser salvo:

INFORMAÇÕES DO DOWNLOAD

- URL do arquivo a ser baixado:
- Pasta local onde o arquivo deve ser salvo:

Logo em seguida, seu programa deve iniciar o download em uma *thread* dedicada e imediatamente voltar a apresentar o MENU INICIAL.

Se o usuário escolher a opção (2), sua aplicação deverá encerrar a aplicação logo após todos os downloads terem sido finalizados.

continuação do enunciado do Exercício 8

Um colega da sua equipe já desenvolver a classe **RunnableDownload**, que é uma implementação da interface *Runnable* que faz o download do arquivo.

```
+-----+
|           RunnableDownload           |
+-----+
|                                     |
+-----+
| RunnableDownload(url:String, pasta:String) |
| run(): void                             |
+-----+
```

Escreva o código da sua aplicação utilizando a classe desenvolvida pelo seu colega.

Bom estudo!

As notas de aula são material de apoio para a aula e não têm o objetivo de apresentar o assunto de maneira exaustiva. Não deixe de ler o material de referência da disciplina.

