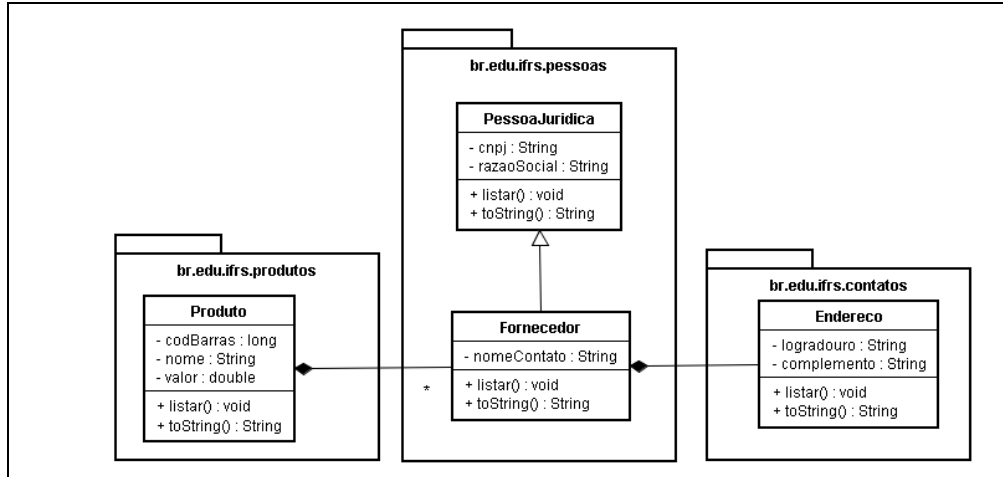


<b>Disciplina:</b> Programação para Web I	<b>Semestre:</b> 3º
<b>Turma:</b> Noite	<b>Revisão Prova</b>
<b>Data:</b> 02/05/2017	<b>Professora:</b> Silvia Bertagnolli

## REVISÃO PROVA

### 1) Analise o esquema abaixo e crie todas as classes usando a linguagem de programação em Java.



- 1.1 A classe Fornecedor tem um endereço
- 1.2 A classe Fornecedor é subclasse de PessoaJuridica
- 1.3 A classe Produto tem vários fornecedores - use uma LinkedList para implementar isso
- 1.4 Use, sempre que possível, this e super

### 2) Monte uma classe de testes para criar e listar dois produtos. Crie o primeiro objeto usando o construtor com parâmetros e o segundo objeto usando o construtor sem parâmetros.

### 3) Reescreva as linhas abaixo utilizando o menor número de linhas e variáveis possível:

#### 3.1

```

FileSystem fs = FileSystems.getDefault();
Path diretorio = fs.getPath("C:", "Diretorio", "Teste");
Path arquivo = diretorio.resolve("Teste.txt");
  
```

#### 3.2

```

File arquivo = new File("Q1.txt");
FileOutputStream saida = new FileOutputStream( arquivo );
BufferedOutputStream escrita = new BufferedOutputStream(saida);
escrita.write( "questao1_prova".getBytes());
  
```

#### 3.3

```

File arquivo = new File("Q1.txt");
FileReader entrada = new FileReader(arquivo);
BufferedReader leitura = new BufferedReader(fr);
  
```

**4) Analise a saída gerada (pilha de execução) quando um programa Java é executado. Agora, responda os itens 4.1 a 4.5.**

```
Exception in thread "main" java.lang.IndexOutOfBoundsException: Index: 3,
Size: 2
    at java.util.LinkedList.checkPositionIndex(LinkedList.java:560)
    at java.util.LinkedList.add(LinkedList.java:507)
    at excecoes.Questao1.main(Questao1.java:18)
Java Result: 1
```

- 4.1 Qual o nome da classe em que foi gerada a exceção?
- 4.2 Qual o nome do método em que a exceção foi gerada?
- 4.3 Qual a linha que gerou a exceção?
- 4.4 Qual exceção foi gerada?
- 4.5 Qual a causa da exceção gerada?

**5) Faça o tratamento das exceções do código abaixo usando try/catch e finally:**

```
public class Questao5 {
    public static void main(String[] args) {
        FileReader in = new FileReader("teste.txt"); //caminho do arquivo
        BufferedReader buff = new BufferedReader(in, 1024);

        StringBuilder builder = new StringBuilder();
        String s = null;
        while ((s = buff.readLine()) != null) {
            builder.append(s).append("\n");
        }
        System.out.println("Conteudo do arquivo:\n\n"+builder);
        buff.close();
        in.close();
    }
}
```

**6) Faça as correções necessárias, bem como o tratamento das exceções do código abaixo. Use try com recursos no tratamento das exceções:**

```
public class Questao6 {
    public static void main(String[] args) throws SQLException, Exception {
        Class.forName("com.mysql.jdbc.Driver");
        String query = "select nome, cpf from clientes";
        String urlDB = "jdbc:mysql://localhost:3306/testbd";

        Connection con = DriverManager.getConnection(urlDB, "user", "user");
        Statement stmt = con.createStatement();
        ResultSet rs = stmt.executeQuery(query);
        while (rs.next()) {
            String nome = rs.getString("nome");
            String cpf = rs.getString("cpf");
            System.out.printf("Nome:%s\t Cpf:%s \n", nome, cpf);
        }
    }
}
```

**7) Considerando o conteúdo de tratamento de exceções análise o código abaixo. Agora responda os itens 7.1 a 7.5:**

```
1. public class Questao7 {
2.     public static void main(String[] args) {
3.         System.out.println("Início do main()");
4.         metodo1();
5.         System.out.println("Fim do main()");
6.     }
7.     private static void metodo1() {
8.         System.out.println("Início do metodo1()");
9.         metodo2();
10.        System.out.println("Fim do metodo1()");
11.    }
12.    private static void metodo2() {
13.        System.out.println("Início do metodo2()");
14.        int[] array = new int[5];
15.        for (int i = 5; i > 0; i--) {
16.            array[i] = i;
17.            System.out.println(i);
18.        }
19.        System.out.println("Fim do metodo2()");
20.    }
21. }
```

7.1 Execute o código acima. Agora, determine qual exceção foi gerada.

7.2 Adicione um try/catch em volta do for, dentro do metodo2(). O que o código imprime?

7.3 Agora, adicione o try/catch em torno de todo o método2(). O que o código vai imprimir? Qual a diferença entre a saída do item 7.2 e esta?

7.4 Adicione o try/catch dentro do for no metodo2(). O que o código vai imprimir?

7.5 Retire o try/catch e coloque ele em torno da chamada do metodo2. Qual a saída gerada? Qual a diferença entre a saída dos itens 7.1, 7.2 e esta?

**8) Criar a interface OperacoesMapa, genérica, conforme as regras abaixo:**

8.1 Lista está definida no pacote package br.edu.ifrs.util;

8.2 método void adicionar(Object chave, Object valor)

8.3 Set getChaves

8.4 Object getValor(Object chave)

8.5 Object remover(Object chave)

8.6 boolean substituir(Object chave, Object valorVelho, Object valorNovo)

8.7 String toString()

- 9) **Crie a classe MeuMapa que deve ser definida como genérica e utilizar a classe HashMap. Para resolver essa questão você ainda deve usar a interface definida na questão anterior e tratamento de exceções.**
- 10) **Agora, determine:**
  - 10.1 Qual código deveria ser incluído/excluído na classe anterior se o método adicionar propagasse a exceção NullPointerException?
  - 10.2 Qual código deveria ser incluído/excluído na classe anterior se o método substituir causasse a exceção NullPointerException?
- 11) **Monte uma classe de testes para a classe MeuMapa após as modificações realizadas na Questão 10.**
- 12) **Redefina a classe Produto de modo que ela possa ser armazenada em um TreeSet. Lembre-se que para fazer isso é necessário implementar interfaces que possibilitam ordenar os seus objetos. Use o código de barras para a sua ordenação.**
- 13) **Agora, monte uma classe de testes de modo que: 5 objetos sejam adicionados ao TreeSet e que o conjunto seja impresso, sem nenhuma chamada explícita a métodos de ordenação.**
- 14) **Usando os códigos desenvolvidos em aula monte as classes que possibilitam cadastrar e listar todos os dados, e ainda, excluir, alterar e pesquisar um Fornecedor no banco de dados. Use os padrões de projeto abordados em aula e enumerações para definição das consultas SQL.**