Documentação – 2º Trabalho de Estrutura de Dados

https://github.com/f-rxfxel/EstrutDados_TR2

Classe Produto

A classe Produto possui três atributos privados: String produto, int quantidade e float valor representando as informações do produto, e seus respectivos métodos de encapsulamento.

```
public class Produto {
	private String produto;
	private int quantidade;
	private float valor;

	//getters and setters;
}
```

A classe No possui dois atributos: Produto produto e ArrayList<No> filhos, além de seus construtores.

```
public class No {
	Produto produto;
	ArrayList<No> filhos = new ArrayList<No>();

	//constructors
}
```

Classe Arvore

O método "`inserir()`" possui uma versão na qual é passada apenas um parâmetro indicando qual o filho a ser adicionado, e outra onde é indicado a qual pai um filho é adicionado.

```
public void inserir(No filho) {
	if (raiz == null) {
		raiz = filho;
		return;
	}
	pai.filhos.add(filho);
	System.out.println("Filho adicionado!\n");
}

public void inserir(No filho, No pai) {
	pai.filhos.add(filho);
	this.pai = pai;
	System.out.println("Filho adicionado!\n");
}
```

O método "`percorrer()`" é responsável pela navegação na estrutura da árvore usando recursividade. Também encarregado de imprimir todos os nós caso seja necessário.

```
public void percorrer(No raiz) {
	Iterator<No> iterator = raiz.filhos.iterator();
	while (iterator.hasNext()) {
		No n = iterator.next();
```

```
                System.out.println(n.produto.getProduto());
                if (!n.filhos.isEmpty()) {
                      percorrer(n);
                }
          }
    }
```

Os métodos “setFolhas()” e “setValor( )” são métodos baseados no mesmo princípio de navegação recursiva usada anteriormente, mas desta vez armazenando os nós sem filhos em um ArrayList<No> folhas. E autoincrementando uma variável float valor que é usada para somar os valores de cada nó multiplicado pela sua quantidade.

```
public void setFolhas(No raiz) {
   Iterator<No> iterator = raiz.filhos.iterator();
   while (iterator.hasNext()) {
         No n = iterator.next();
         if (n.filhos.isEmpty()) {
               folhas.add(n);
         }
   setFolhas(n);
   }
}

public void setValor(No raiz) {
      Iterator<No> iterator = raiz.filhos.iterator();
         while (iterator.hasNext()) {
               No n = iterator.next();
            valor += n.produto.getValor() *
      n.produto.getQuantidade();
               if (!n.filhos.isEmpty()) {
               setValor(n);
         }
      }
   }
```

Os métodos “getFolhas()” e “getValor( )” são formas de obter as informações das lógicas realizadas e gravadas de métodos anteriores, como informações de quais nós são folhas ou por exemplo qual é valor total do produto.

```
public void getFolhas() {
   for (No folha : folhas) {
   System.out.println(folha.produto.getProduto());
   }
}

public void getValor() {
   System.out.printf("R$ %.2f", valor);
}
```

Classe Main

A classe Main possue os seguintes atributos static:

```
ArrayList<String> produtos = new ArrayList<String>();
ArrayList<Integer> quantidades = new ArrayList<Integer>();
ArrayList<Float> valores = new ArrayList<Float>();
```

Que irão conter os dados que serão passados como parâmetros para os construtores na criação dos nós. Estes ArrayLists serão preenchidos com o seguinte método:

```
   private static void getParametros(String next) {
//       "(Produto), (Quantidade), (Valor)".
         String[] parametrosBrutos = next.split(";");
         for (String parametroBruto : parametrosBrutos) {
               String[] parametros = parametroBruto.split(",");
               int index = 0;
//             Cria os parâmetros para o construtor dos nós nos
respectivos tipos,
//             separando-os novamente e adicionando-os aos
respectivos ArrayLists.
               produtos.add(parametros[index]);
               index++;

   quantidades.add(Integer.parseInt(parametros[index]));
               index++;
               valores.add(Float.parseFloat(parametros[index]));
               index++;
         }
}
```