

Programação Orientada por Objetos 2021/2022

Ficha de Laboratório #14

Objetivos

- Introdução ao uso de JavaFX

Programa

- Utilização de Propriedades avançadas
- Utilização de controlos

Regras de implementação

- Criar a aplicação utilizando o IDE BlueJ.
- Implementar o código necessário e testar no fim de cada nível.
- Use as convenções de codificação adotadas para a linguagem Java (ver **Notas**).

Implementação

Nível 1:

- Crie um novo projeto JavaFX Application
- Crie um package `model` e importe para o package as classes `Product` e `Products`
- A classe `Product` refere um produto tendo já implementados os métodos e propriedades necessárias ao projeto.
- A classe `Products` é uma coleção de produtos. Tem dois métodos de classe (um para ler objetos de um ficheiro e um outro para criar dados de exemplo).

Nível 2

- Pretende-se a partir deste nível criar a interface com o utilizador apresentada na imagem : `App01.png`

- Crie a classe `MyProductsMenu` que estende a classe `MenuBar`
- Crie o método privado `draw()` que é chamada no construtor e onde deve efetuar a criação dos seguintes menus:
 - `Home` - Menu com o título "Home" e que deverá conter o `MenuItem` "Exit". Esta opção deverá permitir sair da aplicação. Para sair da aplicação utiliza a instrução `Platform.exit();`
 - `Settings` - Menu com o título "Settings" e que deverá conter o `MenuItem` "Create Sample Data". Ao escolher esta opção deverão ser criados registos de teste chamando o método `CreateSampleData` da classe `Products`.
- Na aplicação principal (`MyProducts`) crie um `Pane` do tipo `BorderPane` que deverá chamar `root`.
- Adicione um `Pane` do tipo `VBox` ao topo do `BorderPane`.
- Adicione o menu criado à `VBox`.
- Teste as funcionalidades criadas.

Nível 3

Pretende-se neste nível criar a lista de produtos existentes.

- Crie a classe `ProductsPane` que estende a classe `ListView<Product>` com o seguinte atributo:
 - `productsList` - `ObservableList<Product>`, lista observável que será utilizada no objeto do tipo `ListView`
- Crie um construtor que recebe como parametro um objeto do tipo `Products`
- Inicie o atributo `productsList` com a lista de productos recebida
- Defina a lista observável como os items a utilizar na `ListView`
- Cada linha `ListView` pode ser considerada um `Node`, assim para implementar o conteúdo de cada uma das linhas defina o método:

```
private Node drawCell(Product item)
```

- Dentro do método `drawCell` implemente dentro de uma `VBox` as Labels necessárias para mostrar a informação. Note-se que deve utilizar as propriedades da classe

`Product` para atualizar as labels criadas

- No construtor da classe `ProductsPane` implemente o seguinte código que irá permitir customização do conteúdo da `ListView`:

```
setPadding(new Insets(10, 10, 10, 10));

setCellFactory((ListView<Product> listview) -> {
    ListCell<Product> cell = new ListCell<Product>() {
        @Override
        public void updateItem(Product item, boolean empty) {
            super.updateItem(item, empty);
            if (empty) {
                setGraphic(null);
            }
            if (item != null) {
                setGraphic(drawCell(item));
            }
        }
    };
    return cell;
});
```

- - Adicione ainda à classe o método seletor da lista criada
- Na classe principal da aplicação adicione o objeto `ProductsPane` ao centro do `BorderPane`. Utilize como Lista de produtos a informação de exemplo fornecida e que pode ser acedida por ficheiro.

Nível 4

A partir deste nível pretende-se criar a lógica de gestão da lista de produtos.

- Cria a classe `ProductPane` que estende a classe `BorderPane`
- Adicione à classe os seguintes atributos:
 - `product` - Objeto do tipo `Product` que servirá para identificar o objeto a alterar
 - `products` - A lista observável de produtos já criada anteriormente
 - `previousPane` - Objeto do tipo `Pane` que servirá para guardar a informação do painel anterior na cena.
- Crie um construtor com todos os atributos existentes
- Implemente o método `draw()` que deverá ser chamado no construtor. Pretende-se criar a interface apresentada em: App02.png
- Os valores dos objetos criados devem responder ao produto que foi passado no construtor. Para um produto novo o valor do atributo passado no construtor deverá ser `null`

Nível 5

- Após criada a interface no nível 4 deverá agora ser implementado:
 - Os botões de Add, Edit e Remove para colocar no ecrã principal
 - Os botões de Save e Cancel na página de edição
- Para as opções de edição e remoção de uma linha e de modo a aceder ao produto selecionado na `ListView` utilize a expressão:

```
Product p = productsPane.getSelectionModel().getSelectedItem(); //Acede ao valor
selecionado da lista
if (p != null)
{
    top.getScene().setRoot(new ProductPane(root, productsPane.getProductsList(), p));
//altera o painel existente na cena que existe no palco
}
```

****.** ******Insira uma imagem (logotipo do IPS Setúbal fornecido) no canto superior direito da janela .

Notas:

Para os identificadores siga as convenções adotadas normalmente, em particular:

1. A notação **camelCase** para o nome das variáveis locais e identificadores de atributos e métodos.
2. A notação **PascalCase** para os nomes das classes.
3. Não utilize o símbolo '_', nem abreviaturas nos identificadores.