

Programação Orientada por Objetos 2022/2023

Ficha de Laboratório #12

Objetivos

- Introdução ao uso de JavaFX

Programa

- Utilização de Propriedades

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 utilizando o template **Ant**
- Crie um classe **Player** e defina os seguintes atributos:
- **name** - Propriedade do tipo **StringProperty** que servirá para guardar o nome do jogador
- **birthDate** - Propriedade do tipo **ObjectProperty**. Será utilizado para guardar um objeto do tipo **LocalDate** para a data de nascimento do player
- **score** - Propriedade do tipo **DoubleProperty** que servirá para guardar a pontuação obtida pelo jogador
- Implemente um construtor sem argumentos onde deve inicializar os atributos.
- Implemente um construtor para receber todos os atributos como argumentos
- Implemente todos os métodos modificadores e seletores dos atributos, tendo em conta o seu tipo primitivo
- Implemente os métodos seletores das propriedades criadas e dos seus valores

Nível 2

- Crie a classe **PlayerPane** que estende a classe **VBox**
e defina o atributo:
- **player**- atributo do tipo **Player**

- Defina um construtor que recebe como parametro um objeto do tipo **Player**
- No construtor lance uma exceção do tipo **NullPointerException** caso o atributo **player** seja **null**
- Implemente um método privado **draw** , que deverá ser chamado no construtor. Neste método implemente o seguinte:
- Defina a margem da **VBox** para 10 através do método **setPadding**
- Defina Labels para os atributos do objeto player e adicione à **VBox**
- Para a label **name** defina o tipo de fonte "Verdana" com tamanho "20"
- Para a label **birth date** defina o tipo de fonte "Verdana" com tamanho 15
- Para a label **score** defina o tipo de fonte "Verdana" com tamanho 15 e formato "Bold". Para o formato "Bold" user a seguinte instrução:

```
lblScore.setStyle("-fx-font-weight: bold");
```

- Implemente o código necessário, nesta classe, para que sempre que uma das propriedades de um objeto Player tenha alterações as labels sejam automaticamente atualizadas
- Na classe principal do programa:
 - No método **start** remova todo o código e adicione

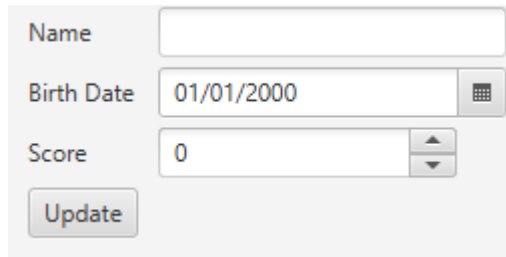
```
@Override
public void start(Stage primaryStage)
{
    Scene scene = new Scene(root, 600, 400);
    primaryStage.setTitle("Testar propriedades");
    primaryStage.setScene(scene);
    primaryStage.show();
}
```

- Adicione um painel do tipo **BorderPane** e adicione o painel à cena
- Criei um objeto do tipo **Player**
- Adicione o painel do tipo **PlayerPane** à coluna da esquerda do **BorderPane**

Nível 3

- Crie a classe **PlayerEditPane** que estende a classe **GridPane**
e defina o atributo:
- **player** - atributo do tipo **Player**
- Defina um construtor que recebe como parametro um objeto do tipo **Player**

- No construtor lance uma exceção do tipo `NullPointerException` caso o atributo `player` seja `null`
- Implemente um método privado **draw**, que deverá ser chamado no construtor. Neste método implemente o seguinte:
- Defina o `Gap` vertical da grid para 5
- Defina o `Gap` horizontal da grid para 10
- Defina a margem da `GridPane` para 10 através do método `setPadding`
- Deverá implementar os seguintes controlos para permitir ao utilizador a alteração dos dados do **Player**



- `name`- Label e TextField na linha 1 da grid
 - `birthDate`- Label e DatePicker na linha 2 da grid
 - `score` - Label e Spinner na linha 3 da grid. Defina para este controlo um valor minimo de 0, maximo de 20 e intervalos de 0.5
- Na classe principal do programa:
 - Adicione o painel do tipo `PlayerEditPane` à coluna da direita do `BorderPane`

Nível 4

- Na classe `PlayerEditPane`, método `draw` adicione um controlo do tipo `Button` para efetuar o `Update` da informação
- Implemente o código necessário no botão de modo a atualizar a informação do `player`

Nível 5

- Implemente o código necessário para que a sincronização entre o que é inserido nos controlos sejam imediatamente atualizado nos controlos da classe `PlayerPane`

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.