

Декларативный пользовательский интерфейс

Декларативность означает, что описывается желаемый результат, а не последовательность действий для его достижения.

Использование декларативного языка описания UI помогает отделить интерфейс от остального кода приложения

Для декларативного описания интерфейса используются XML-технологии

Что такое XML

- Расширяемый язык разметки
- Язык с простым формальным синтаксисом, удобный для создания и обработки документов программам и одновременно удобный для чтения и создания документов человеком
- Для определения структуры используется разметка
- Разработчик волен создать разметку в соответствии с потребностями к конкретной области, будучи ограниченным лишь синтаксическими правилами языка

Синтаксис

Синтаксис элемента:

<имя_элемента [список атрибутов] > открывающий тег

содержимое элемента

<имя_элемента / > закрывающий тег

Синтаксис атрибута:

Атрибут = "значение"

Логическая структура

Пролог

Объявления (`<?xml version="1.1" encoding="UTF-8" ?>`)

инструкции обработки

Комментарии (`<!-- это комментарий -->`)

Корневой элемент — обязательная часть документа

Элементы + атрибуты

`<title>` Заголовок `</title>`

`<ingredient amount="3" unit="стакан">Мука</ingredient>`

FXML

Язык разметки, основанный на XML

Описывает элементы сцены и задает их параметры

Иерархическая структура узлов сцены
соответствует структуре XML-документа

Отдельный fxml-файл с описанием сцены

Каждому fxml-файлу соответствует отдельный
класс-контроллер

Документация

[docs.oracle.com/javase/8/javafx/api/javafx/fxml/doc-
files/introduction_to_fxml.html](https://docs.oracle.com/javase/8/javafx/api/javafx/fxml/doc-files/introduction_to_fxml.html)

Пролог

Объявление

```
<?xml version="1.0" encoding="UTF-8"?>
```

Инструкции импорта

```
<?import javafx.scene.control.*?>
```

```
<?import javafx.scene.control.Label?>
```

Корневой элемент

Корневой элемент fxml-документа соответствует корневому элементу сцены

Панель-контейнер для расположения узлов сцены

```
<GridPane xmlns:fx="http://javafx.com/fxml" >
```

xmlns:fx – обязательный атрибут, определяющий пространство имен

Основные элементы

Экземпляр класса

имя тега начинается с прописной буквы, совпадает с именем класса

объект создается, если есть соответствующие инструкция импорта и элемент в корне документа

Свойства класса

имя тега начинается со строчной буквы, совпадает с именем свойства, для которого есть метод `set`, или свойства-списка, для которого есть только метод `get`

Статическое свойство класса

имя тега состоит из префикса (имя класса), точки и имени свойства

Пример

```
<Label text="Organizations List" minHeight="16" minWidth="69">  
  <font>  
    <Font name="Verdana" size="24.0" />  
  </font>  
</Label>  
<TableView fx:id="table">  
  <columns>  
    <TableColumn prefWidth="100.0" text="Name"  
      fx:id="nameCol" />  
    <TableColumn prefWidth="100.0" text="Boss Name"  
      fx:id="bossNameCol" />  
    <TableColumn prefWidth="100.0" text="Personnel"  
      fx:id="personnelCol" />  
  </columns>  
</TableView>
```

Атрибуты

Свойство класса

Статическое свойство

Обработчик событий

Свойства с префиксом fx: —

fx:id задает имя создаваемого объекта

fx:controller связывает fxml-документ с
соответствующим классом контроллером

Примеры атрибутов

Свойства

```
<Label text="Name" GridPane.columnIndex="0"  
      GridPane.rowIndex="0"/>
```

Идентификатор

```
<TextField fx:id="addName" />
```

Контроллер

```
<GridPane fx:controller =  
      "tablexml.OrganizationAddDialogController">
```

Обработчик события

```
<Button onAction="#handleOk" />
```

Связывание свойств элементов сцены

Префикс \$ указывает ссылку на объект

Значения свойств одинакового типа можно связать с помощью имени объекта и символов \${}

Метка будет автоматически отображать текст, введенный в поле ввода:

```
<TextField fx:id="textField"/>
```

```
<Label text="${textField.text}"/>
```

Контроллер

Доступ к объектам fxml-документа через декларацию @FXML

@FXML

```
private TextField addName;
```

Обработчики событий

@FXML

```
private void handleCancel() {  
    okClicked = false;  
    dialogStage.close();  
}
```

Загрузка fxml-файла

Класс FXMLLoader

Конструктор

FXMLLoader()

Методы

static <T> T load(URL location) (T – корневой элемент сцены)

void setLocation(URL location)

<T> T load() (T – корневой элемент сцены)

<T> T getController() (T – контроллер связанный с fxml-файлом)

Загрузка fxml-документа в главном классе приложения

```
Parent root = FXMLLoader.load  
(getClass().getResource("FXMLDocument.fxml"));
```

```
Scene scene = new Scene(root, 500, 300);
```

```
FXMLLoader loader = new FXMLLoader();
```

```
loader.setLocation(TableFXML.class.getResource  
("FXMLDocument.fxml"));
```

```
Parent root = loader.load();
```

```
FXMLDocumentController controller = loader.getController();
```

```
controller.setMainApp(this);
```

```
Scene scene = new Scene(root);
```


Загрузка fxml-документа для дополнительного окна приложения

```
FXMLLoader loader = new FXMLLoader();  
loader.setLocation(TableFXML.class.  
getResource("OrganizationAddDialog.fxml"));
```

```
Parent root = loader.load();  
Stage dialogStage = new Stage();  
Scene scene = new Scene(root);  
dialogStage.setScene(scene);
```

```
OrganizationAddDialogController controller =  
    loader.getController();  
controller.setDialogStage(dialogStage);  
dialogStage.showAndWait();
```