也是Quick Start Guide to JavaFx读书笔记

Design pattern(structure): MVC (Model-view-Controller)

UI --- independent--- logic

View: fxml

You can design it without the need to know the underlying data of the related java files.

Controller: .java file coupled with the .fxml file is the Controller

Model : the supporting Java files in your project that contain logic or data that is displayed in the view are the “models”.

{

* 在**VIEW**上显示它
* 用户使用**CONTROLLER**
* 操作数据(更新，修改，删除，..)，**MODEL**上的数据已更改。
* 在**VIEW**上显示**MODEL**的数据。

}

Event-driven programming: *a Graphical Application is a big loop ...*

An event can be anything. A key pressed, a move

of the mouse, a finger swiping a touch screen,

somebody jumping in front of a webcam ...

Anything that can be translated into an electrical

signal reaching the computer.

A JavaFX application derives from the Application class

in the JavaFx package. It means that it automatically

inherits standard attributes and methods.

1.

JavaFx will also automatically call a function called init().

By default, this function does nothing. You can write

your own version, and connect to a network or a

database, or read a parameter file.

2.

What you must write is a function called "start()" that

takes a "Stage" (the name given to windows in JavaFx)

as parameter. The function adds the widgets to the

window and defines how it looks, and how widgets will

react.

3. You must write the event handlers you need, and

nothing else – JavaFx will run the application until it

calls an exit routine (perhaps associated with a "Quit"

button) or it receives the event "Window destroyed".

4.

Create an instance of the Application class

Call the init() method

Call the start(javafx.stage.Stage) method

Wait for the application to finish:

the application calls Platform.exit()

or window closed

C a ll t h e s t o p ( ) m e t h o d

(It will then call a stop()

method where you can undo(disconnect) what you have done in init()

– disconnect for instance from a database or network.

Like with init(), rewriting stop() is optional.)