**Javafx**

How to set up:

* All javafx classes **MUST** extend **application**
* There needs to a **launch** statement in the **main** method
* The **launch** statement calls on the **start** method
* Before anything is displayed you need a layout which could be a stackPane and then add children to the layout
* Also you then needs to set up your scene by setting the size and the layout
* Then with the primary stage you need to set up the scene
* Then set the primary stage to show

Terms

* The entire **window** is called the **stage**
* The **content** on the window is called the **scene(Buttons, text)**
* **When something interactive happens its called an event**

Handle User Events

* If you ever want to deal with an event you need to Implements EventHandler<>
* There are many different types of eventhandelrs but for a simple button click you need a actionevent
* For an eventhandler you need a method called handle and the argument is (ActionEvent event)
* Then in the start method you need to **set** the **on action** of the **button** which can be set to **this** if the handler method is in the class.
* In the handle event you need to get the source
* **Or**
* Instead of implementing a eventhandler you can do this while setting the on action as you pass the same argument EventHandler<ActionEvent>()\_
* Then use the same handle method same rules.
* This is recommended as you don’t have to get the source
* **Or**
* when setting the on action of the button use the argumetns e-> then what you want it to do e.g. sysou

Layout

* Vbox – children are laid out in a vertical column
* StackPane
* Grid – you can add padding which means it will have a margin, and you can use v and h gap to set the gap. So everything is grouped together which makes it look much better. And two set things in columns you use set constraints(child, numbers)

Children

* Button – allows actions
* Label – displays text
* Text field – user can type stuff in

Switching scenes

* when switching scenes you use the same stage so you don’t need a new stage for each just two new scenes
* to change the scene on set the on action of the button to stage.setscene(name of scene)
* also set the stage to the first scene so it knows where to start with and then show

Alert Box doesn’t let you use the other window till you have dealt with the alertbox e.g. do you want to save?

Alert box

* initModality(modality.APPLICATION\_MODE) sets the application mode so that the user can’t interact with anything outside this scene.
* This is what scenes an alertbox from other windows
* Also this is different than the last option as its switching stage not scene.
* Also instead of show use show and wait as this blocks any user interaction until this is closed.

Communicating between stages

* Instead on the display method being void it is turned into a return method.
* Then to store the results you need to do the type and name then it then equals the button action.
* Which is stage name then the method display

Closing a program properly (saving)

* Set on close request then set it to the close method.
* Also you can use a consume statement to control the request

Embedding layouts

* You can use more than one layouts by embedding them.
* Make layouts then use a boarder, stack and grid pane to embed them.