Note:

This is a case where the videos are easier to understand than the pdfs alone.

The videos comment the slides step by step, this helps build your understanding.

Don’t read and rush. Watch the video instead!

Where should I write my lines of code????

It is nice to have learned how to write a couple of lines of code. It is nicest to know where we should write these lines, so that what is written in these lines get "performed" / "executed" by the app

1) Lines of codes should get between { }

In programming, you often write code in a place surrounded by curly braces:

```
{
   String myText;
   myText = "welcome to my app";
   Label myLabel = new Label(myText);
   etc…
}
```

Then you put a name to this group of lines of code. The name should be without spaces or special characters, as usual (see next slide).

So, we give a name to this group of lines of codes inside the curly braces:

Just for your information, a block of code with a name is called a function, or a method.

```
pleaseExecuteThisCodeWhenTheAppStarts {
    String myText;
    myText = “welcome to my app”;
    Label = new Label(myText);
    etc…
}
```

Just like when you create objects, you could add parameters to be used each time the block of code is executed. When there is no parameter, just like here, you still have to write the empty parentheses ()

To be precise, you have a bit more weirdness to add when naming this block of code:

Means « don’t do anything special after the block of code has finished executing. »

```
void pleaseExecuteThisCodeWhenTheAppStarts () {
    String myText;
    myText = “welcome to my app”;
    Label = new Label(myText);
    etc…
}
```

Now, a (tiny) bad news and a (very) good news

1. The bad news

The name of the block of code can be extended with several parameters etc., so at the end it gets veeeeery complex:

Wowowowoow??!!!

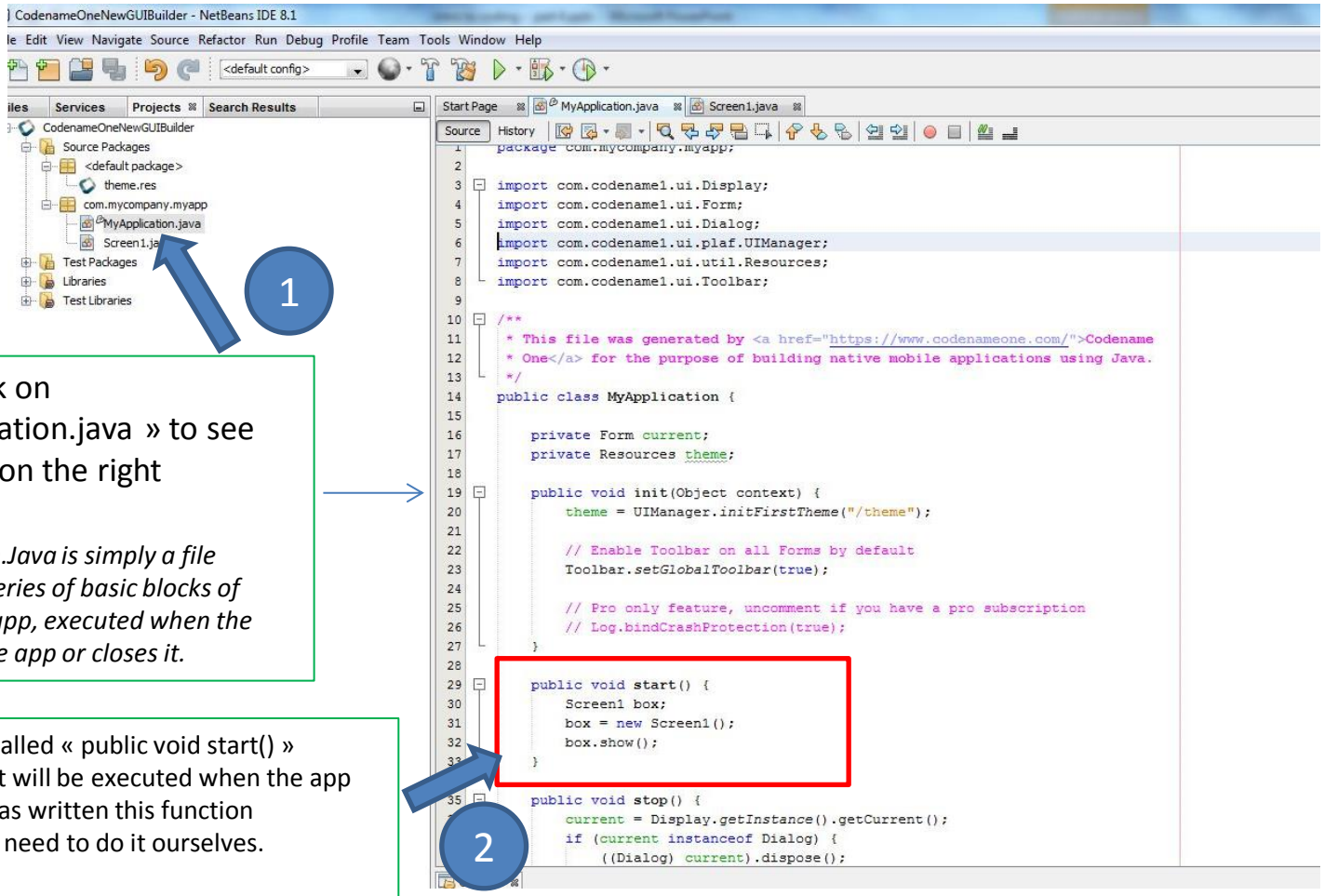
```
public Integer pleaseExecuteThisCodeWhenTheAppStarts (ActionEvent e ) throws IOException {
    String myText;
    myText = “welcome to my app”;
    Label = new Label(myText);
    etc…
}
```

1. **The good news**

We don’t need to write that ourselves. Codename One provides shortcuts that actually get it written for us. We will just concentrate on writing what’s inside the curly braces { }

see next slide for 2 examples

Example 1: a block of code called « public void start () ».



double click on « MyApplication.java » to see its content on the right

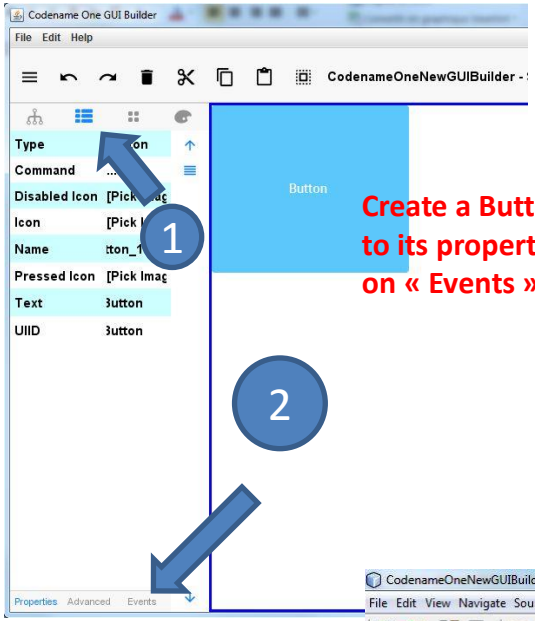
MyApplication.Java is simply a file containing a series of basic blocks of codes for the app, executed when the users starts the app or closes it.

Example: this function called « public void start() » contains everything that will be executed when the app starts. CodenameOne has written this function automatically, we don't need to do it ourselves.

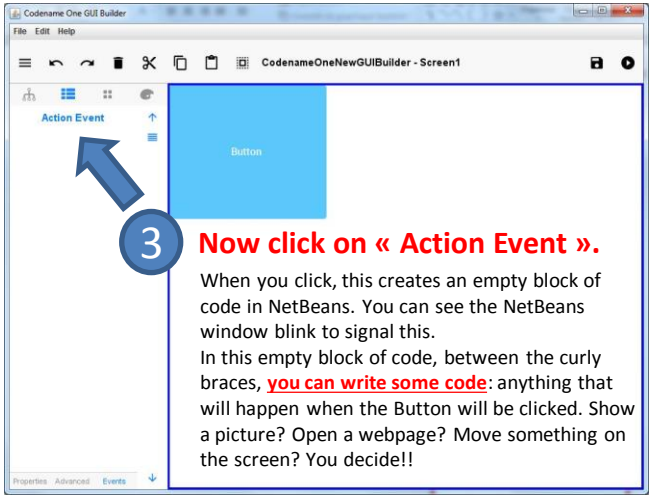
<u>Inside the curly braces of this function, we are free to write any lines of codes that we want.</u>

Here, we have written 3 lines, instructing to show the first screen of the app: a) create a box specialized in containing screens, put a new screen in this box, apply the « show » action to the box.

Example 2: code being executed when a user click on a button in the app

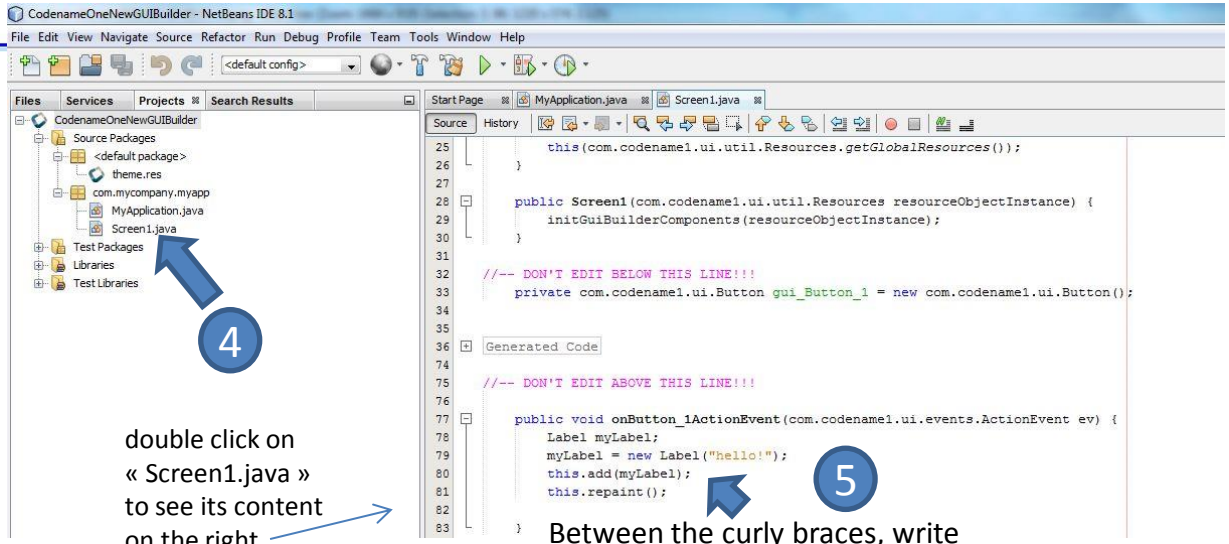


Create a Button, select it and go to its properties (1). Then click on « Events » (2).



Now click on « Action Event ».

When you click, this creates an empty block of code in NetBeans. You can see the NetBeans window blink to signal this.
In this empty block of code, between the curly braces, **you can write some code**: anything that will happen when the Button will be clicked. Show a picture? Open a webpage? Move something on the screen? You decide!!



double click on « Screen1.java » to see its content on the right

Between the curly braces, write anything you want. Here I write:

1. Create a box called « myLabel » specialized in containing Labels.
2. Put a new Label in it, with some text
3. Add the label to the current screen (« this » designates the box containing the screen we are currently working on)
4. Apply the « repaint » action, meaning it is going to refresh the screen & the Label will be shown.

EMLYON business school

CODAPPS