

Forum:[Hack Forums](#)

Thread: [How do you write clean GUI code?](#)

3mogamer - January 24, 2015, 6:11 pm

Hello Guys,

I been programming for a while now and all my code looks neat till you get to the GUIs. For example, I have four buttons, each with its own listeners and secondary features like designs and it looks pretty horrendous.

ex. Code */*Initialization of Buttons&Textfield*/*

```
Button button1= new Button("Button 1");
Button button2= new Button("Button 2");
Button button3= new Button("Button 3");
Button button4= new Button("Button 4");
TextField field1 = new TextField("");
```

*/*Addition of Listeners*/*

```
button1.setOnAction(new EventHandler<ActionEvent>() {
    @Override
    public void handle(ActionEvent arg0) {
        /*Omitted or else this code will look crappier*/
    }
});
button2.setOnAction(new EventHandler<ActionEvent>() {
    @Override
    public void handle(ActionEvent arg0) {
        /*Omitted or else this code will look crappier*/
    }
});
button3.setOnAction(new EventHandler<ActionEvent>() {
    @Override
    public void handle(ActionEvent arg0) {
        /*Omitted or else this code will look crappier*/
    }
});
button4.setOnAction(new EventHandler<ActionEvent>() {
    @Override
    public void handle(ActionEvent arg0) {
        /*Omitted or else this code will look crappier*/
    }
});
```

Wow, doesn't that code look beautiful. Luckily, I removed the bloat and kept the bare minimum. Usually, I would have several more Controls/Buttons and that would aesthetically ruin the code. Anyway my question here is ... "In what way do you guys write GUIs? (Just plain out instantiating or is there some sort of trick to make it neater?)

Corios - January 24, 2015, 6:48 pm

Coming from .net I like the way it's done there. GUI goes in a separate file that links an event to a

method that you would declare on the class you want to use it from, looks clean and keeps the ugly part away from the nice part.

ObscureCoder - January 24, 2015, 6:53 pm

I don't tend to do much AWT or Swing GUI programming in Java but I do it more like this:

```
Code import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JPanel;
import javax.swing.SwingUtilities;
import javax.swing.UIManager;

public class Main extends JFrame {

    public Main() {
        super("Sexy Title");
        setSize(500, 400);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setLocationRelativeTo(null);

        JPanel panel = new JPanel();
        getContentPane().add(panel);

        JButton button = new JButton("Swag 4 lyf");
        panel.add(button);

        setVisible(true);

        button.addActionListener(listener -> {
            System.out.println("Sexy msg");
        });
    }

    public static void main(String[] args) {
        try {
            UIManager.setLookAndFeel(UIManager.getSystemLookAndFeelClassName());
        } catch (Exception e) {
            e.printStackTrace();
        }

        SwingUtilities.invokeLater(() -> new Main());
    }
}
```

Some people dislike doing it that way, I prefer it for Swing. I do JavaFX mostly now.

Trisomy 21 - January 24, 2015, 8:35 pm

Definitely keep any of your output code/GUI away from the actual logic and calculations of your program, but apart from that I agree with you GUI code in Java makes me want to cut myself.

ObscureCoder - January 25, 2015, 4:38 am

(Jan 24, 2015 09:35 PM)Trisomy 21 Wrote: [\[CITE\]](#) Definitely keep any of your output code/GUI away from the actual logic and calculations of your program, but apart from that I agree with you GUI code in Java makes me want to cut myself.

Then use dat dank JavaFX:

Spoiler ([Click to View](#))

??Austin?? - January 25, 2015, 9:01 am

Some people Recccomend MVC it looks neat.

Watch this and you will get it.

<https://www.youtube.com/watch?v=dTVVa2gfht8>

300th post

Corios - January 25, 2015, 11:18 am

(Jan 25, 2015 10:01 AM)?---Abyte ---? Wrote: [\[CITE\]](#) Some people Recccomend MVC it looks neat.

Watch this and you will get it.

<https://www.youtube.com/watch?v=dTVVa2gfht8>

300th post

ok, MVC concept isn't needed in a desktop application, that's overkill lol <.<

nOrr - January 25, 2015, 2:13 pm

(Jan 25, 2015 12:18 PM)Corios Wrote: [\[CITE\]](#) ok, MVC concept isn't needed in a desktop application, that's overkill lol <.<

Uhm... no it's not. In fact it's perfectly suited for a desktop application, and once you get the hang of it, following the MVC design pattern is actually going to get you in a better mood. Heck, I've heard the MVC design pattern can cure depression*.

*) Does not actually cure depression.

3mogamer - January 26, 2015, 8:46 pm

Thanks, Ill check out MVC out. Is there any other way that we can improve GUI code or is it just God's Wish that GUI-code should make us cut ourselves.

nOrr - January 26, 2015, 9:03 pm

(Jan 26, 2015 09:46 PM)3mogamer Wrote: [\[CITE\]](#) Thanks, Ill check out MVC out. Is there any other way that we can improve GUI code or is it just God's Wish that GUI-code should make us cut ourselves.

Sure there is. Read the book Clean Code by uncle Bob and you should get an idea of how to do it.

MasterBugPatch - January 27, 2015, 12:35 am

They do have online beautifiers for various programming languages. I'm sure you can find one that interest you.

ObscureCoder - January 27, 2015, 12:37 am

Wait, so why do you think GUI code is unclean? I find it incredibly straightforward and well structured.

3mogamer - January 27, 2015, 8:03 pm

(Jan 27, 2015 01:37 AM)ObscureCoder Wrote: [\[CITE\]](#) Wait, so why do you think GUI code is unclean? I find it incredibly straightforward and well structured.

It feels like how I first started writing Java Code(All in one Class in a single Main method) and looking back at that now is quite different to how I currently write code which is much more OOP. Yes, I also read your example... it was indeed straightforward but the structuring just gives off a feel

that I am back at where I started and it doesn't feel right.

I will also give the book Clean Code a read but I feel like that focuses overall rather than what I am just looking for right now, GUIs.

I'll do more research and ect. Thanks guys for your responses!

edit: Actually MVC looks great! Thanks for the suggestion!

nOrr - January 27, 2015, 9:21 pm

(Jan 27, 2015 09:03 PM)3mogamer Wrote: [\[CITE\]](#) It feels like how I first started writing Java Code(All in one Class in a single Main method) and looking back at that now is quite different to how I currently write code which is much more OOP. Yes, I also read your example... it was indeed straightforward but the structuring just gives off a feel that I am back at where I started and it doesn't feel right.

I will also give the book Clean Code a read but I feel like that focuses overall rather than what I am just looking for right now, GUIs.

I'll do more research and ect. Thanks guys for your responses!

edit: Actually MVC looks great! Thanks for the suggestion!

Why do you think GUI code is much different than code not related to GUI? Imagine that you need three buttons grouped together. Create a class that extends JPanel or similar and either make your class a generic "GroupedButtons" class, or a specific "StartMenuButtonGroup". Then you add buttons to that class, and the class will handle structuring the buttons for that particular area of the GUI. In your boiler plate code you stich together the bigger pieces, and voila: one pretty GUI.

Also, in general, wherever in your code you feel like writing a single line comment to describe what happens at a particular place such as "//Set values to JLabels", create a method "setJLabelValues" or similar. Single Responsibility Principle. It's all in the book. Just not specifically using Swing components...

Also MVC is wonderful. I love the idea of building up a skeleton and thinking "This method should do this, but I will do the implementation in the future, when it's all structured". :P

3mogamer - January 28, 2015, 8:55 pm

(Jan 27, 2015 10:21 PM)nOrr Wrote: [\[CITE\]](#) -snip-

I guess it was just myself structuring it incorrectly. We are two different people and, the way I did it was ugly. Anyway, Thanks for the examples, tips, book suggestion, and everything else!

nOrr - January 28, 2015, 8:59 pm

(Jan 28, 2015 09:55 PM)3mogamer Wrote: [\[CITE\]](#) I guess it was just myself structuring it incorrectly. We are two different people and, the way I did it was ugly. Anyway, Thanks for the examples, tips, book suggestion, and everything else!

Well isn't that why we're discussing it in the first place? :P

I'm not attacking you... I'm just saying that the book is fine even for code that will set up visual components in a GUI.

Doug Dimmadome - January 29, 2015, 7:04 am

This is how I format my GUI code.

```
Code // button1
```

```
Button button1 = new Button("Button 1");
button1.setOnAction(new EventHandler<ActionEvent>() {
    @Override
    public void handle(ActionEvent arg0) {
        // TODO: Code
    }
});
```

```
// button2
Button button2= new Button("Button 2");
button2.setOnAction(new EventHandler<ActionEvent>() {
    @Override
    public void handle(ActionEvent arg0) {
        // TODO: Code
    }
});
```

```
// button3
Button button3 = new Button("Button 3");
button3.setOnAction(new EventHandler<ActionEvent>() {
    @Override
    public void handle(ActionEvent arg0) {
        // TODO: Code
    }
});
```

```
// button4
Button button4 = new Button("Button 4");
button4.setOnAction(new EventHandler<ActionEvent>() {
    @Override
    public void handle(ActionEvent arg0) {
        // TODO: Code
    }
});
```

```
// field1
TextField field1 = new TextField();
However. I have been using Java8 for current projects and things get a little cleaner still.
```

Lambdas usually make life a little cleaner/easier. Just don't go overboard with them ;)

```
Code // button1
Button button1 = new Button("Button 1");
button1.setOnAction((event) -> {
    // TODO: Code
});
```

```
// button2
Button button2= new Button("Button 2");
button2.setOnAction((event) -> {
    // TODO: Code
});
```

```
// button3
Button button3 = new Button("Button 3");
button3.setOnAction((event) -> {
    // TODO: Code
});
```

```
// button4
Button button4 = new Button("Button 4");
button4.setOnAction((event) -> {
    // TODO: Code
});
```

```
// field1
TextField field1 = new TextField();
```

I might also suggest you take a look at JavaFX. It takes some getting used to, but it really makes life easier as it more separates GUI code from backend code with FXML files.

Kind Regards,
Massive

3mogamer - January 29, 2015, 8:25 pm

(Jan 28, 2015 09:59 PM)nOrr Wrote: [\[CITE\]](#) Well isn't that why we're discussing it in the first place?
:P

I'm not attacking you... I'm just saying that the book is fine even for code that will set up visual components in a GUI.

Yes, of course it was what we were discussing :D. Ill give the book Clean Code by Martin a read, since it was the only one I found online.

(Jan 29, 2015 08:04 AM)Massive Wrote: [\[CITE\]](#) -snip-

Oh I am using Javafx right now. Also, Wow! I didn't know you could use (->) in java.

Doug Dimmadome - January 30, 2015, 8:30 am

(Jan 29, 2015 09:25 PM)3mogamer Wrote: [\[CITE\]](#) Yes, of course it was what we were discussing :D. Ill give the book Clean Code by Martin a read, since it was the only one I found online.

Oh I am using Javafx right now. Also, Wow! I didn't know you could use (->) in java.

Yes, the "->" is only lambdas though. You cannot declare a normal function like that. Also, you can only use it in Java 8, it will not work in Java 7 or below.

ObscureCoder - January 30, 2015, 2:07 pm

OP, you should stick with JavaFX; it's bae.

??????M1kep?????? - February 1, 2015, 12:05 am

(Jan 27, 2015 10:21 PM)nOrr Wrote: [\[CITE\]](#) Why do you think GUI code is much different than code not related to GUI? Imagine that you need three buttons grouped together. Create a class that extends JPanel or similar and either make your class a generic "GroupedButtons" class, or a specific "StartMenuButtonGroup". Then you add buttons to that class, and the class will handle structuring the buttons for that particular area of the GUI. In your boiler plate code you stich together the bigger pieces, and voila: one pretty GUI.

Also, in general, wherever in your code you feel like writing a single line comment to describe what happens at a particular place such as "//Set values to JLabels", create a method "setJlabelValues" or similar. Single Responsibility Principle. It's all in the book. Just not specifically using Swing

components...

Also MVC is wonderful. I love the idea of building up a skeleton and thinking "This method should do this, but I will do the implementation in the future, when it's all structured". :P

Can you explain/give an example of why you're talking about at the beginning? At the moment [my gui in my program](#) is like five hundred lines and I just feel like I could be doing something differently....

nOrr - February 1, 2015, 1:56 am

You could experiment with dividing your GUI code into sections. Each section should be a class of its own. It's just an idea I have, and not sometime I've read in a book by a famous author... the idea is that you should be able to add a "MyTopActionBar" to your main JFrame (let's pretend you have an action bar in the top) which will give you, say, a JPanel with components in, instead of you adding all the components and styling by hand.

Bibi - February 1, 2015, 3:50 am

yeah lol sometimes i look at old code and try think back to what i was on at the time

Immortal - February 1, 2015, 10:05 pm

Yo brah why not do something like this to make it neater?

PHP Code

```
<?php
public class YourGui extends JFrame
{
    public JButton jBtn1, jBtn2;
    public YourGui(){
        this.setSize(150,250); //ex
        this.setDefaultCloseOperation(DISPOSE_ON_CLOSE);
        this.setVisible(true);
        ActionListener al = new MultiListener();
        jBtn1 = new JButton("Dont Click me bro");
        jBtn1.addActionListener(al);
        jBtn2 = new JButton("Click me bro");
        jBtn2.addActionListener(al);
    }
    // GUI SHIT BRAH

    public static void main(String[] args){
        YourGui yg = new YourGui();
    }

    private class MultiListener implements ActionListener{
        public void actionPerformed(ActionEvent e){
            if(e.getSource() == jBtn1){
                //btn 1 click
                System.out.println("I SAID DONT CLICK ME BRO!");
            }
            if(e.getSource() == jBtn2){
```

```

        //btn 2 click
        System.out.println("brahhhhhh");
    }
}
}

```

I wrote this off the top of my head so it might not be right, but you should get the point of the how to check for multiple buttons in one listener instead of giving a listener to each button.

ObscureCoder - February 2, 2015, 3:17 pm

(Feb 1, 2015 11:05 PM)Imm?rtal Wrote: [CITE](#) Yo brah why not do something like this to make it neater?

PHP Code

```

<?php
public class YourGui extends JFrame
{
    public JButton jBtn1, jBtn2;
    public YourGui(){
        this.setSize(150,250); //ex
        this.setDefaultCloseOperation(DISPOSE_ON_CLOSE);
        this.setVisible(true);
        ActionListener al = new MultiListener();
        jBtn1 = new JButton("Dont Click me bro");
        jBtn1.addActionListener(al);
        jBtn2 = new JButton("Click me bro");
        jBtn2.addActionListener(al);
    }
    // GUI SHIT BRAH

    public static void main(String[] args){
        YourGui yg = new YourGui();
    }

    private class MultiListener implements ActionListener{
        public void actionPerformed(ActionEvent e){
            if(e.getSource() == jBtn1){
                //btn 1 click
                System.out.println("I SAID DONT CLICK ME BRO!");
            }
            if(e.getSource() == jBtn2){
                //btn 2 click
                System.out.println("brahhhhhh");
            }
        }
    }
}

```

I wrote this off the top of my head so it might not be right, but you should get the point of the how to check for multiple buttons in one listener instead of giving a listener to each button.

PHP Code

```

<?php
button.addActionListener(listener -> {
    System.out.println("Pressed");
}

```

```
});
```

??????M1kep?????? - February 2, 2015, 5:17 pm

(Feb 2, 2015 04:17 PM)ObscureCoder Wrote: [\[CITE\]](#) PHP Code

```
<?php  
button.addActionListener(listener -> {  
    System.out.println("Pressed");  
});
```

I think this is only in Java 8 and up, correct?

ObscureCoder - February 2, 2015, 11:40 pm

(Feb 2, 2015 06:17 PM)??????M1kep?????? Wrote: [\[CITE\]](#) I think this is only in Java 8 and up, correct?

Yes, I tend to write the latest java version because I like to use newer classes etc. for doing things. Also, lambda is bae, makes me feel boss af.

3mogamer - February 4, 2015, 8:47 pm

(Feb 1, 2015 01:05 AM)??????M1kep?????? Wrote: [\[CITE\]](#) Can you explain/give an example of why your talking about at the beginning? At the moment [my gui in my program](#) is like five hundred lines and I just feel like I could be doing something differently....

Funny thing was that I was making a Manga Program too. ?