

# Polymorphism



**Mosh Hamedani**  
[codewithmosh.com](https://codewithmosh.com)

1: Project

7: Structure

2: Favorites

Main.java × CheckBox.java × UIControl.java × TextBox.java ×

```
1 package com.codewithmosh;
2
3 public class UIControl {
4     private boolean isEnabled = true;
5
6     // public UIControl(boolean isEnabled) {
7     //     this.isEnabled = isEnabled;
8     // }
9
10    public void render() {
11        |
12    }
13
14    public void enable() {
```

Ant Build Maven

1: Project

Z: Structure

2: Favorites

Main.java

CheckBox.java

UIControl.java

TextBox.java

```
1 package com.codewithmosh;
2
3 public class CheckBox extends UIControl {
4     @Override
5     public void render() {
6         System.out.println("Render CheckBox");
7     }
8 }
9
```

Ant Build

Maven

1: Project

Z: Structure

★ 2: Favorites

Main.java ×

CheckBox.java ×

UIControl.java ×

TextBox.java ×

```
5
6 // public TextBox() {
7 //     super(true);
8 // }
9
10
11 @Override
12 public void render() {
13     System.out.println("Render TextBox");
14 }
15
16 @Override
17 public String toString() {
18     return text;
```

Ant Build

Maven

Main.java ×

CheckBox.java ×

UIControl.java ×

TextBox.java ×

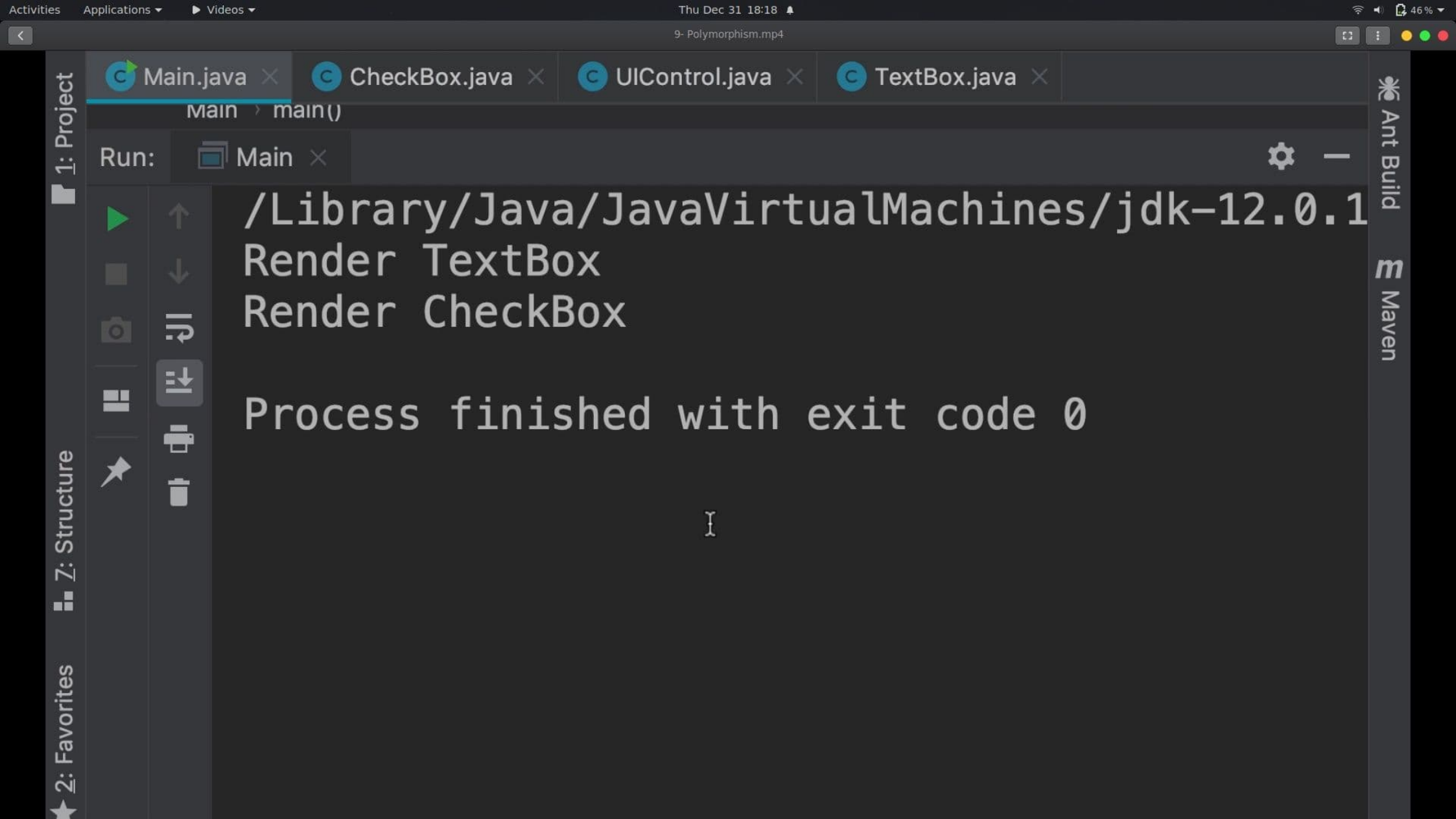
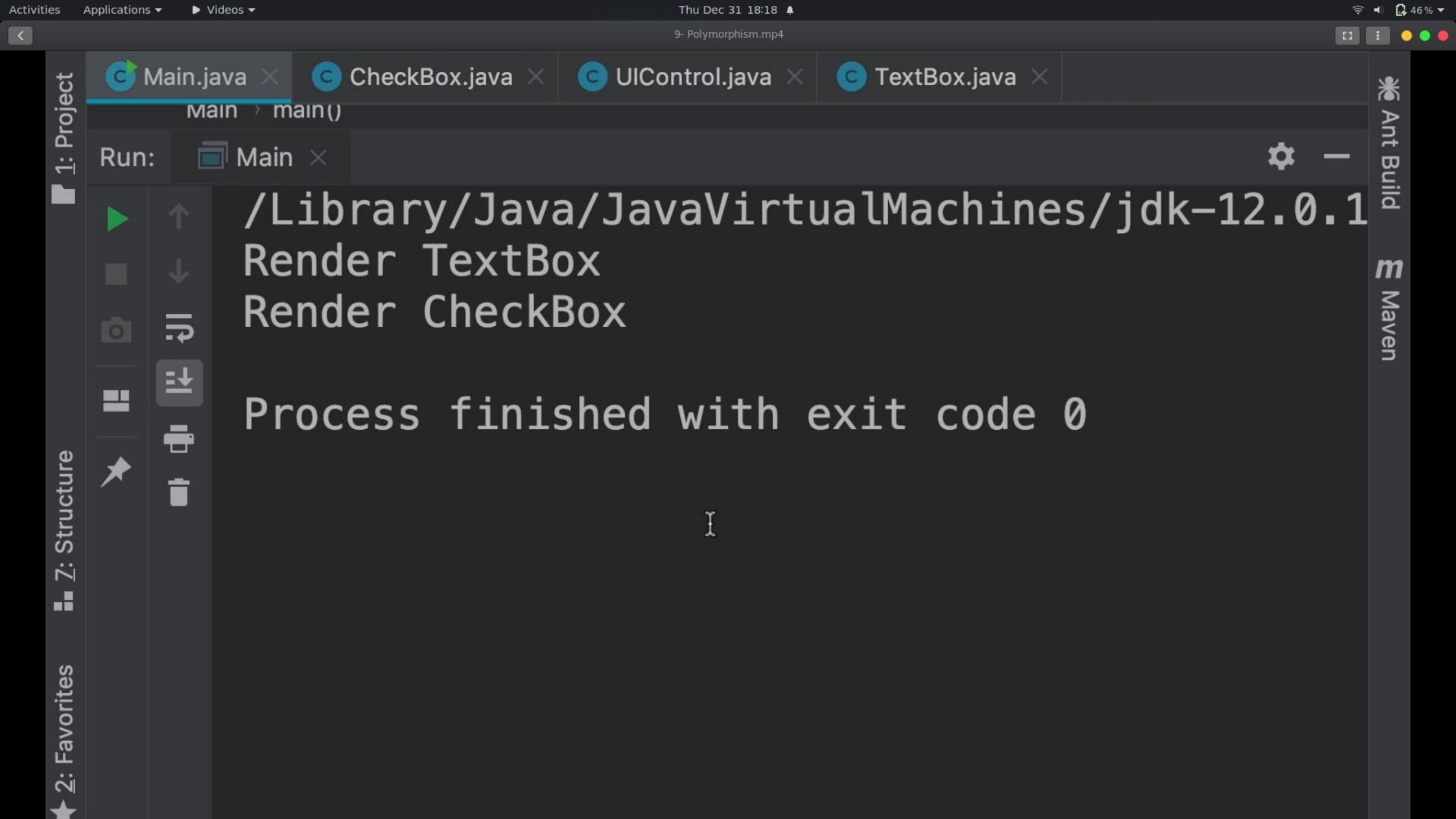


Ant Build



Maven

```
1  kage com.codewithmosh;
2
3  lic class Main {
4
5  public static void main(String[] args) {
6      UIControl[] controls = { new TextBox(), new CheckBox
7  for (var control : controls)
8      control.render();
9  }
```



# Abstract Classes and Methods



**Mosh Hamedani**  
[codewithmosh.com](https://codewithmosh.com)



Main.java ×

UIControl.java ×

CheckBox.java ×

TextBox.java ×

```
1 package com.codewithmosh;
2
3 public abstract class UIControl {
4     private boolean isEnabled = true;
5
6     // public UIControl(boolean isEnabled) {
7     //     this.isEnabled = isEnabled;
8     // }
9
10    public void render() {
11
12    }
13
14    public void enable() {
```



1: Project

Z: Structure

2: Favorites

Main.java ×

UIControl.java ×

CheckBox.java ×

TextBox.java ×

Ant Build

Maven

```
1 package com.codewithmosh;
2
3 public class Main {
4
5     public static void main(String[] args) {
6         UIControl[] controls = { new UIControl(), new Te
7         for (var control : controls)
8             control.render();
9     }
10 }
11
```

Main.java

UIControl.java

CheckBox.java

TextBox.java



Ant Build



Maven

```
1 package com.codewithmosh;
2
3 public abstract class UIControl {
4     private boolean isEnabled = true;
5
6     // public UIControl(boolean isEnabled) {
7     //     this.isEnabled = isEnabled;
8     // }
9
10    public abstract void render();
11
12    public void enable() {
13        isEnabled = true;
14    }
```

1: Project

Z: Structure

2: Favorites

Main.java x UIControl.java x CheckBox.java x TextBox.java x

```
1 package com.codewithmosh;
2
3 public class CheckBox extends UIControl {
4     // Override
5
6     // System.out.println("Render CheckBox");
7     // }
8 }
9
```

Class 'CheckBox' must either be declared abstract or implement abstract method 'render()' in 'UIControl'

1: Project

Z: Structure

2: Favorites

Main.java x UIControl.java x CheckBox.java x TextBox.java x

```
1 package com.codewithmosh;
2
3 public abstract class CheckBox extends UIControl {
4     // @Override
5     // public void render() {
6     //     System.out.println("Render CheckBox");
7     // }
8 }
9
```

Ant Build Maven

1: Project

Z: Structure

2: Favorites

Main.java x UIControl.java x CheckBox.java x TextBox.java x

```
1 package com.codewithmosh;
2
3 public abstract class CheckBox extends UIControl {
4     // @Override
5     // public void render() {
6     //     System.out.println("Render CheckBox");
7     // }
8 }
9
```

Ant Build Maven

# Final Classes and Methods



**Mosh Hamedani**  
[codewithmosh.com](https://codewithmosh.com)



Main.java × UIControl.java × CheckBox.java × TextBox.java ×

```
1 package com.codewithmosh;
2
3 public final class CheckBox extends UIControl
4     @Override
5     public void render() {
6         System.out.println("Render CheckBox");
7     }
8 }
9
10 public class MyCheckBox extends CheckBox {}
11
```

Cannot inherit from final 'com.codewithmosh.CheckBox'