

Design Pattern Assignment-2-

Heba Saeed Hamdan

202010678

- Part2:Simple factory assignment:
- **Client class:**

```
public class ButtonDesign {  
    simpleButtonFactory factory;  
    public ButtonDesign(simpleButtonFactory factory){  
        this.factory=factory;  
    }  
    public Button drawButton(String type){  
        Button b=factory.createDesign(type);  
        b.draw();  
        return b;  
    }  
}
```

- **Simple factory class:**

```
public class simpleButtonFactory {  
    public Button createDesign(String type){  
        Button button=null;  
        if(type.equals("rounded")){  
            button=new RoundedButton();  
        }else if(type.equals("square")){  
            button=new SquareButton();  
        }  
        return button;  
    }  
}
```

- **Product abstract class "Button":**

```
public abstract class Button {  
    public abstract void draw();  
}
```

- **Concrete products classes:**

```
public class RoundedButton extends Button { //first one when choose rounded button Design  
    @Override  
    public void draw() {  
        System.out.println("Rounded Button Design");  
    }  
}
```

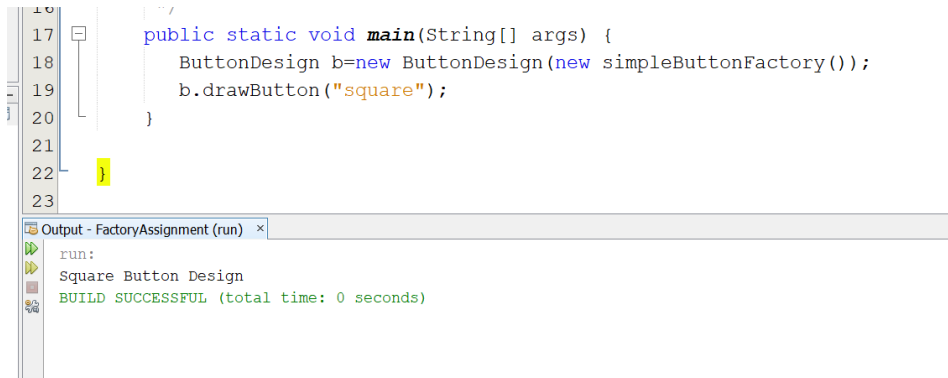
public class SquareButton extends Button{//the second when choose square Button design

@Override

```
public void draw() {  
    System.out.println("Square Button Design");  
}
```

- **Test class:**

```
Public class main{  
    public static void main(String[] args) {  
        ButtonDesign b=new ButtonDesign(new simpleButtonFactory());  
        b.drawButton("square");  
    }  
}
```

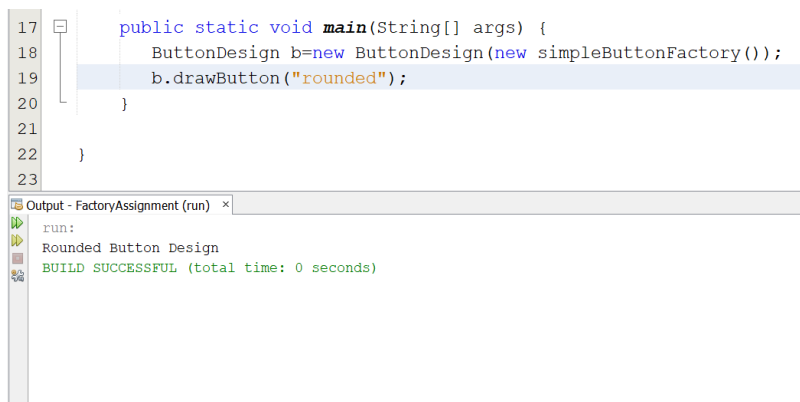


The screenshot shows an IDE with a code editor and an output window. The code editor displays the following code:

```
17 public static void main(String[] args) {  
18     ButtonDesign b=new ButtonDesign(new simpleButtonFactory());  
19     b.drawButton("square");  
20 }  
21  
22 }  
23
```

The output window, titled "Output - FactoryAssignment (run)", shows the following output:

```
run:  
Square Button Design  
BUILD SUCCESSFUL (total time: 0 seconds)
```



The screenshot shows an IDE with a code editor and an output window. The code editor displays the following code:

```
17 public static void main(String[] args) {  
18     ButtonDesign b=new ButtonDesign(new simpleButtonFactory());  
19     b.drawButton("rounded");  
20 }  
21  
22 }  
23
```

The output window, titled "Output - FactoryAssignment (run)", shows the following output:

```
run:  
Rounded Button Design  
BUILD SUCCESSFUL (total time: 0 seconds)
```

-Part2:Factory Method :

- **Creator abstract class :**

```
public abstract class ButtonDesign {  
    public Button drawButton(String type){  
        Button button=createDesign(type);  
        button.draw();  
        return button;  
    }  
    public abstract Button createDesign(String type);  
}
```

- **Concrete creator classes:**

```
public class MacButton extends ButtonDesign {  
    @Override  
    public Button createDesign(String type) {  
        Button button=null;  
        if(type.equals("square")){  
            button=new MacButtonSquare();  
        }else if(type.equals("rounded")){  
            button=new MacButtonRounded();  
        }  
        return button;  
    }  
}
```

```

public class WindowsButton extends ButtonDesign{

    @Override

    public Button createDesign(String type) {

        Button button =null;

        if(type.equals("square")){

            button=new WindowsButtonSquare();

        }else if(type.equals("rounded")){

            button=new WindowsButtonRounded();

        }

        return button;

    }

}

```

- **Product abstract class “Button”:**

```

public abstract class Button {

    public abstract void draw();

}

```

- **Concrete products classes:**

```

public class MacButtonRounded extends Button {

    //for Mac OS the button design will be Rounded

    @Override

    public void draw() {

        System.out.println("Mac Rounded Button Design");

    }

}

```

```

public class MacButtonSquare extends Button {
    //for Mac OS the button design will be Square

    @Override
    public void draw() {
        System.out.println("Mac Square Button Design ");
    }

}

public class WindowsButtonRounded extends Button {
    //for Windows OS the button design will be Rounded

    @Override
    public void draw() {
        System.out.println("Windows Rounded Button Design");
    }

}

public class WindowsButtonSquare extends Button {
    //for Windows OS the button design will be Square

    @Override
    public void draw() {
        System.out.println("Windows Square Button Design");
    }

}

```

- **Test class:**

```

public class Main{
    public static void main(String[] args) {
        ButtonDesign b=new MacButton();//for mac os and choosing rounded design
        b.drawButton("rounded");
    }}

```

```
16 | */
17 | public static void main(String[] args) {
18 |     ButtonDesign b=new MacButton();
19 |     b.drawButton("rounded");
20 | }
21 |
22 | }
23 |
```

Output - parttwoassignment (run) x

run:
Mac Rounded Button Design
BUILD SUCCESSFUL (total time: 0 seconds)

```
16 | */
17 | public static void main(String[] args) {
18 |     ButtonDesign b=new WindowsButton();
19 |     b.drawButton("rounded");
20 | }
21 |
22 | }
23 |
```

Output - parttwoassignment (run) x

run:
Windows Rounded Button Design
BUILD SUCCESSFUL (total time: 0 seconds)

```
15 | * @param args the command line arguments
16 | */
17 | public static void main(String[] args) {
18 |     ButtonDesign b=new WindowsButton();
19 |     b.drawButton("square");
20 | }
21 |
22 | }
23 |
```

Output - parttwoassignment (run) x

run:
Windows Square Button Design
BUILD SUCCESSFUL (total time: 0 seconds)

```
16 | */
17 | public static void main(String[] args) {
18 |     ButtonDesign b=new MacButton();
19 |     b.drawButton("square");
20 | }
21 |
22 | }
23 |
```

Output - parttwoassignment (run) x

run:
Mac Square Button Design
BUILD SUCCESSFUL (total time: 0 seconds)