**Inner Classes – Part-04**

* **Normal Java Class VS Anonymous Inner Class**

A normal Java class can extend only one class at a time, of course anonymous inner-class also can extend only one class at a time.

A normal Java class can implement any number of interfaces simultaneously. But anonymous inner class can implement only one interface at a time.

A normal Java class can extend a class and implement any number of interfaces simultaneously. But anonymous can extend a class or can implement an interface, but not both simultaneously.

In normal Java class we can write any number of constructors, but in anonymous inner classes, we can’t write any constructor explicitly (because the name of the class and name of the constructor must be same), but anonymous inner classes not having any name.

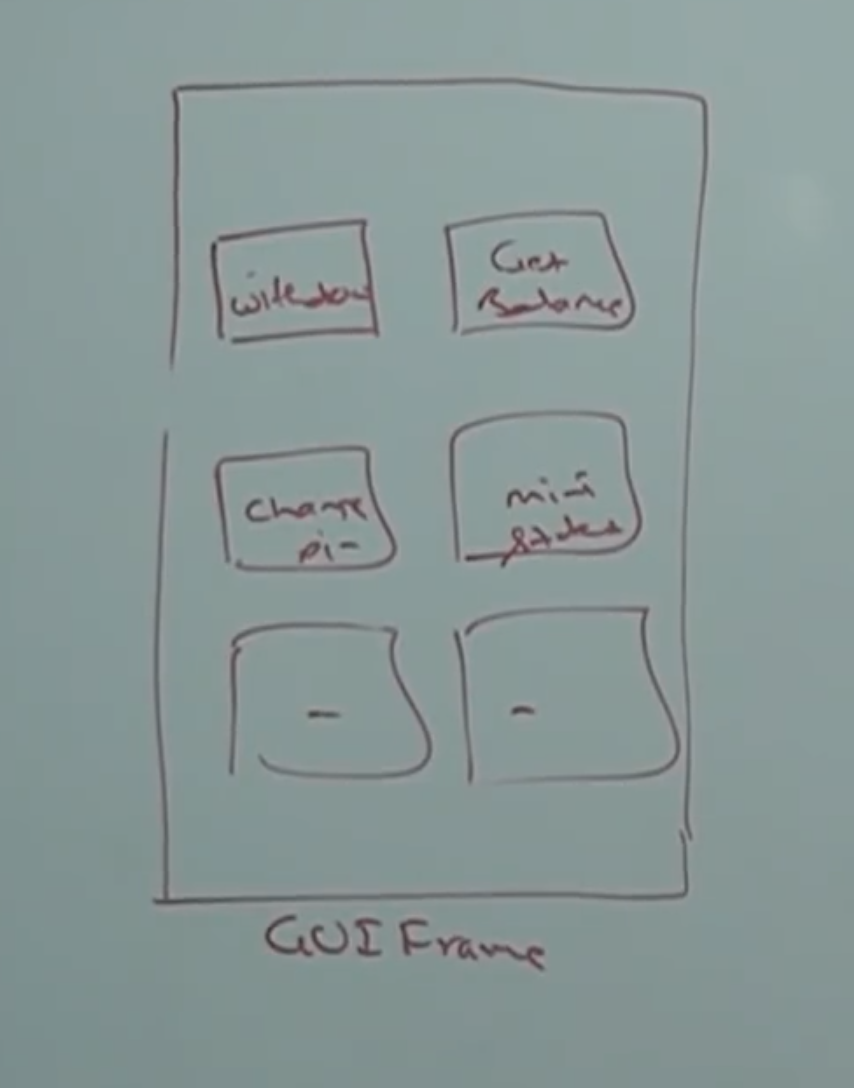
Note:

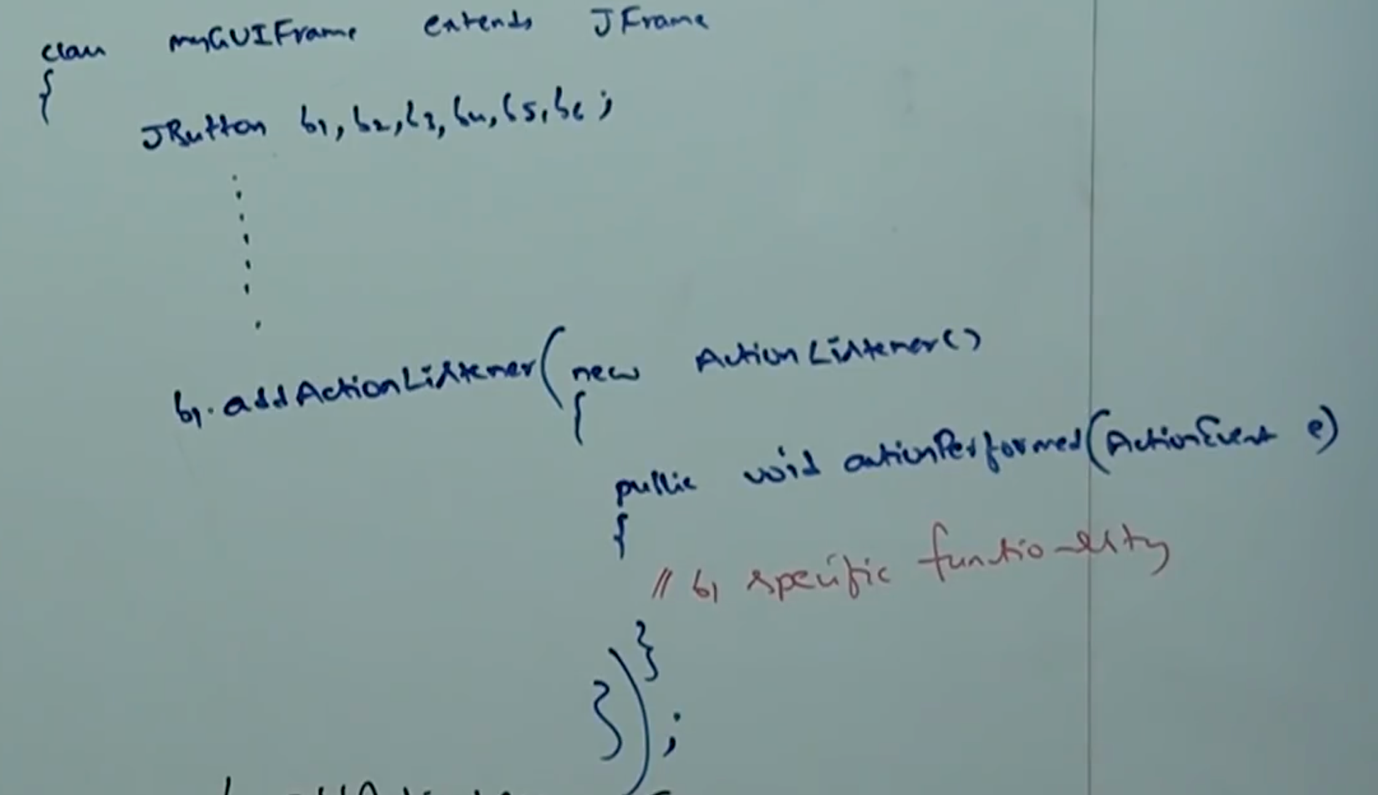
If the requirement is standard and required several times then we should go for normal top-level class.

If the requirement is temporary and required only once (instant use) then we should go for anonymous inner-class.

* **Where anonymous inner classes are best suitable?**

We can use anonymous inner-classes frequently in GUI based applications, to implement event handling.





* **Static Nested Classes:**

Sometimes we can declare inner class with static modifier such type of inner-classes is called static nested classes.

In the case of normal or regular inner class without existing outer class object there is no chance of existing inner class object. That is inner class object is strongly associated with outer class object.

But in the case of static nested classes without existing outer class object there may be chance of existing nested class object, hence static nested class object is not strongly associated with outer class object.

Note:

class Test{

int x;

static int y;

}

To access the instance variable x, we need the object of Test class.

But, to access the value of y, we don’t need the object of Test class.

Apply the same analogy with the static nested classes

**Example:**

class Outer{

static class Nested{

public void m1(){

System.out.println(“Static nested class method”);

}

}

public static void main(String[] args){

Nested n =new Nested();

n.m1();

}

}

Note:

If you want to create nested class object from outside of Outer class, then we can create as follows.

Outer.Nested n = new Outer.Nested();

In normal or regular inner-classes we can’t declare any static members.

But in static nested classes we can declare static members including main method. Hence, we can invoke static nested class directly from command prompt.

Example:

class Test{

static class Nested{

public static void main(String[] args){

System.out.println(“Static nested class main”);

}

}

public static void main(String[] args){

System.out.println(“Ouer class main method”);

}

}

javac Test.java

java Test

Outer nested class main

java Test$Nested

Static nested class main

From normal or regular classes, we can access both static and non-static members of outer class directly. But static nested classes, we can access static members of outer class directly and we can’t access non-static members.

**Example:**

class Test{

int x = 10;

static int y = 20;

static class Nested{

public void m1(){

System.out.println(x);

System.out.println(y);

}

}

}

**CE:** non-static variable x cannot be referenced from a static context.

* **Difference between normal or regular inner class and static nested class.**

|  |  |  |
| --- | --- | --- |
|  | Normal or Regular Inner Class | Static Nested Classes |
| 1 | Without existing outer class object, there is no chance of existing inner class object. That is inner class object is strongly associated with outer class object. | Without existing outer class object there may be a chance of existing static nested class object. That is static nested class object is not strongly associated with outer class object. |
| 2 | In normal or regular inner classes, we can’t declare static members. | In static nested classes we can declare static members. |
| 3 | In normal or regular inner class, we can’t declare main method and hence we can’t invoke inner class directly from command prompt. | In static nested classes we can declare main method and hence we can invoke nested class directly from command prompt. |
| 4 | From normal or regular inner-classes we can access both static or non-static members of outer class directly. | From static nested classes we can access only static members of outer class. |