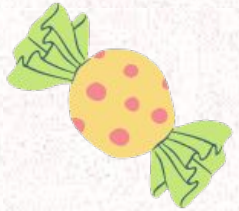



InnerClass

SAM





Inner Class in Java | Use, Types, Example

- A class declared inside another class is known as **nested classes in java**. The scope of a nested class is tied by the scope of its enclosing class (outer class).
- A nested class in java has access to the members (including private members) of its enclosing class. But, the enclosing class does not have access to the members of the nested class.
- The class, which is a member of another class, can be either static or non-static.
- The member class, which is declared with a static modifier is known as **static nested class in java**.
- The member class, which is non-static is known as inner class or non-static nested class. It is the most important type of nested class.

Inner class in Java

- An inner class in java is a class that is declared inside of another class without static modifier. It is also commonly known as a non-static nested class in Java. It can access all members (variables and methods) of its outer class.
- An inner class cannot have any kind of static members. The members of the java inner class may be:
  1. Instance variables
  2. Instance methods
  3. Constructors
  4. Initializer block
  5. Inner class

Features of Inner class

- There are several important features of an inner class that is as follows:
  1. An inner class cannot have the same name as the outer class. But, it is possible to use the same names for members of both outer and inner classes.
  2. The scope of inner class is bounded by the scope of its outer class.
  3. Without existing an outer class object or instance, there will be no chance of an existing inner class object.

Features of Inner class

4. An inner class can directly access all the variables and methods of the outer class including private.

5. Since the inner class is a regular member of the outer class just like declared variables and methods in the class.

That's why, we can apply all java access modifiers such as public, default, protected, and private to inner class similar to the other members of a class. But the outer or normal class cannot be private or protected.

6. If the variable name of inner class is the same as the variable name of outer class, we can access the outer class variable like this. OuterClassName.this.VariableName; here this represents current outer class object.

7. Java inner class is hidden from another class in its enclosing class. Therefore, it provides a safety mechanism in the application program and decreases readability (understanding) of the program.

8. An object of inner class is often created in its outer class and cannot be created from other classes.

9. Both outer class and inner class objects are created in separate memory locations.

Example

```
public class OuterClass
 {
      int number = 6;
      public void display(){
      System.out.println("Hey there");
      }
public class InnerClass {
      int number = 8;
      public void info(){
      System.out.println("What's goin' on in the innerClass");
      }
}
```

```
public class MainClass{
      public static void main (String[] args)
      {
      OuterClass outer = new OuterClass();
      outer.display();
      }
}
```

How to create non-static InnerClass object?

```
public class MainClass{
        public static void main (String[] args)
        {
        OuterClass outer = new OuterClass();
        outer.display();

        //InnerClass Object
        OuterClass.InnerClass inner = outer.new InnerClass();
        inner.info();



        }
}
```

How to create static InnerClass object?

```
public class OuterClass
 {
      int number = 6;
      public void display(){
      System.out.println("Hey there");
      }
public static class InnerClass {
      int number = 8;
      public void info(){
      System.out.println("What's goin' on in the innerClass");
      }
}
```

```
public class MainClass{
      public static void main (String[] args)
      {
      OuterClass outer = new OuterClass();
      outer.display();

      //InnerClass Object
      OuterClass.InnerClass inner = new OuterClass.InnerClass();
      inner.info();

      }
}
```

How to create Method Local InnerClass object?

```
public class OuterClass
 {
        int number = 6;
        public void display(){
        System.out.println("Hey there");
        class LocalInnerClass{
        String info = "Hey sam";
        public void info(){
                System.out.println(info);
        }
        }
        LocalInnerClass lic = new LocalInnerClass();
        lic.info();
        }
}
```

```
public class MainClass{
        public static void main (String[] args)
        {
        OuterClass outer = new OuterClass();
        outer.display();
        }
}
```

Realtime Example of Inner class

Suppose there is a university. University contains several departments such as electronics, computer science, electrical, mechanical, etc.

Assume t ity due to its illegal losed, all departme

That's wh i.e. their functiona university object, th

```
class University {  // University is outer class.
  . . . . . .
class Department { // Department is inner class.
  . . . . . .
 }
}
```

Since the department is always a part of the university class. Hence, we must declare the department class inside the university class.

Realtime Example of Inner class

Assume that there is a car. Within a car, there are several important individual
component of the car i

The engine
xisting car object, the

Since an e
, the class engine mu

```
class Car {
  .........
  class Engine  {
     ..........
  }
}
```

Use of Inner class in Java

- There are the following important uses of inner class in Java.

  1. The inner class is a valuable feature because it allows us to group classes and interfaces in one place that logically belongs together.

  2. It helps to control the visibility of one within the other so that it can be more readable and maintainable.

  3. The best use of java inner class is when its functionality is tied to its outer class. That is, without an existing outer class object, there is no chance of existing inner class object. Understand the real-time examples.

  4. Java inner class is useful for providing security for the important code. For example, if we declare inner class as private, it is not available to other classes. This means that an object to the inner class cannot be created in any other classes.

Resources

https://www.scientecheasy.com/2020/06/inner-class-in-java.html/#:~:text=Java%20inner%20class%20is%20useful,help%20of%20realtime%20example%20program.