How To Java Tutorials

Lesson 10 Java Classes

Author: Ramy Hakam

TABLE OF CONTENTS

Class Declaration

The class body (the area between the braces) contains all the Code that provides for the life cycle of the objects created From the class: constructors for initializing new objects, Declarations for the fields that provide the state of the class And its objects, and methods to implement the behavior of The class and its objects.

```
  class MyClass {
      // field, constructor, and
      // method declarations
  }
```

The preceding class declaration is a minimal one. It contains Only those components of a class declaration that are Required. You can provide more information about the class, Such as the name of its superclass, whether it implements Any interfaces,

```
class MyClass extends MySuperClass implements
YourInterface {
    // field, constructor, and
    // method declarations
}
```

Define Classes

In general, class declarations can include these components, In order:

1-Modifiers
public or private
2-The class name
with the initial letter capitalized by convention. .
3-The name of the class's parent (superclass),
if any, preceded by the keyword extends. A class can only extend (subclass) one parent..
4-A comma-separated list of interfaces implemented by The class,
5-The class body, surrounded by braces, {}..

CODEING

Creating ClassDemo Examples

Nested Classes

The Java programming language allows you to define a class Within another class. Such a class is called a nested class And is illustrated here:

```
class OuterClass {
    ...
    class NestedClass {
        ...
    }
}
```

A nested class is a member of its enclosing class. Non-static Nested classes (inner classes) have access to other members of The enclosing class, even if they are declared private. Static Nested classes do not have access to other members of the Enclosing class. As a member of the OuterClass, a nested Cass can be declared private, public, protected.

Why Use Nested Classes?

1-It is a way of logically grouping classes that are only Used in one place

2-It increases encapsulation

3-It can lead to more readable and maintainable code

Static Nested Classes

As with class methods and variables, a static nested class is Associated with its outer class. And like static class methods, A static nested class cannot refer directly to instance Variables or methods defined in its enclosing class: it can use Them only through an object reference.

Inner Classes

As with instance methods and variables, an inner class is associated with an instance of its enclosing class and has direct access to that object's methods and fields. Also, because an inner class is associated with an instance, it cannot define any static members itself.

Shadowing

If a declaration of a type (such as a member variable or a Parameter name) in a particular scope (such as an inner Class or a method definition) has the same name as Another declaration in the enclosing scope, then the Declaration shadows the declaration of the enclosing scope. You cannot refer to a shadowed declaration by its name Alone.

CODEING

Creating NastedClassDemo Examples

Local Classes

Local classes are classes that are defined in a block, which is A Group of zero or more statements between balanced braces. You Typically find local classes defined in the body of a Method.

You can define a local class inside any block For example, You can define a local class in a method body, a for loop, or An if clause.

Accessing Members of Enclosing Class

A local class has access to the members of its enclosing class In addition, a local class has access to local variables. However, a local class can only access local variables that Are declared final.
However, starting in Java SE 8, a local class can access local variables and parameters of the enclosing block that are final or effectively final. A variable or parameter whose value is never changed after it is initialized is effectively final. Starting in Java SE 8, if you declare the local class in a method, it can access the method's parameters.
Note:
Local Class can not declare static members except it will declare as constant variable "final"
Also can not define interfaces in local class

CODEING

Creating LocalClassDemo Examples

Anonymous Classes

Anonymous classes enable you to make your code more concise. They enable you to declare and instantiate a class at the same time. They are like local classes except that they do not have a name. Use them if you need to use a local class only once
While local classes are class declarations, anonymous classes are expressions, which means that you define the class in another expression.

Syntax of Anonymous Classes

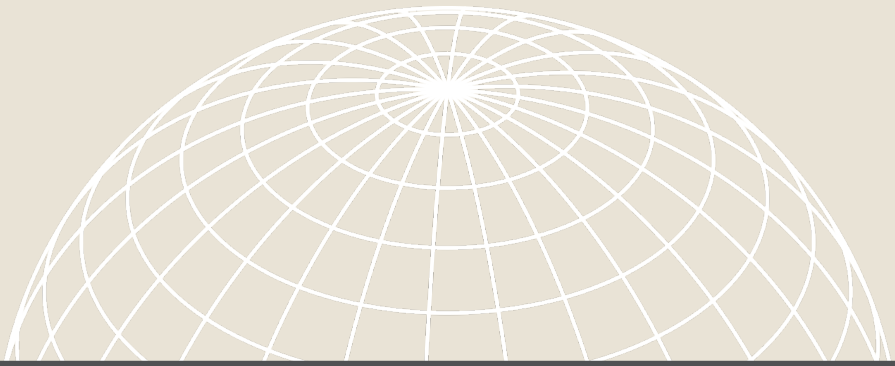
As mentioned previously, an anonymous class is an expression. The syntax of an anonymous class expression is like the invocation of a constructor, except that there is a class definition contained in a block of code.
The anonymous class expression consists of the following:
1-The new operator
2-The name of an interface to implement or a class to extend.
3-A body, which is a class declaration body. More specifically, in the body, method declarations are allowed but statements are not.

NOTE:

Accessing members for anonymous class is the same as local class and can not define static members or interface

CODEING

Creating AnonyClassDemo Examples



Thank You For Watching

I am still not Better, But I always try .

See You Next