

Nesting classes (or types) within another class (or type)

A class can contain other types within the class body, such as other classes, interfaces, enums, and records.

These are called nested types, or nested classes.

You might want to use nested classes when your classes are tightly coupled, meaning their functionality is interwoven.

Nested Classes

The four different types of nested classes you can use in Java are: the static nested class, the inner class, and the local and anonymous classes.

Type	Description
static nested class	declared in class body. Much like a static field, access to this class is through the Class name identifier
instance or inner class	declared in class body. This type of class can only be accessed through an instance of the outer class.
local class	declared within a method body.
anonymous class	unnamed class, declared and instantiated in same statement.

Important Restrictions for nested classes were removed in JDK16

Before JDK16, only static nested classes were allowed to have static methods.

As of JDK16, all four types of nested classes can have static members of any type, including static methods.

In the next video, we'll start out talking about static nested classes.

There's a lot to cover, so let's get started.