Topic: Inner Class

An inner class refers to the class that is declared inside another class or interface. Declaring a class this way provides a few benefits

1. It makes the code clean and readable.
2. The private methods of the outer class can be accessed, which brings a new dimension and making it closer to the real world.
3. Inner classes can help optimize code

(GeekForGeeks, 2023)

There are three/four types of inner classes:

* Nested Inner Class can access private instance variables from the outer class. It allows for access modifiers.
* Method Local Inner Classes are declared within a method of the outer class.
* Static Nested Classes only have access to static methods and variables of the outer class.
* Anonymous Inner Classes are created without a class name.
* (GeekForGeeks, 2023)

Inner classes take advantage of how Java encapsulation works to create clean, optimal code. This helps reduce code complexity and improve code access control. In the real world you may use the outer class as a template for a series of polymorphic inner classes. This is often easier than having separate files for each class.

Reference:

GeeksforGeeks. (2023, April 22). *Inner class in java*. GeeksforGeeks. https://www.geeksforgeeks.org/inner-class-java/

Christopher,

Inner classes are helpful for writing clean code. They form natural inheritance structures, without the need more dozens of files. This alone could justify their usage, but the other properties are great as well.

Anitha,

You touched on all the main advantages of inner classes. I personally value inner classes most for their tendency to result in clean code, but that is far from the only benefit. As you mentioned, anonymous inner classes in particular offer a unique solution to event handling and one-off implementations.