

Final Classes

Using the final keyword on a class means it can't be extended.

You declare a class `final` if its definition is complete, and no subclasses are desired or required.

Enums and Records are final classes.

Subclasses can take advantage of mutable fields on parent classes, if the parent classes aren't implementing defensive code.

One of the easiest ways to prevent this, is to make your class final.

Final Classes

Operations	final class	abstract class	private constructors only	protected constructors only
Instantiate a new instance	yes	no	no	yes, but only subclasses, and classes in same package
A subclass can be declared successfully	no	yes	no	yes

Private constructors will prevent both a new instance and a new subclass from being created.

Protected constructors will prevent an instance from being created outside of a subclass or the package.

Final Classes

Operations	final class	abstract class	private constructors only	protected constructors only
Instantiate a new instance	yes	no	no	yes, but only subclasses, and classes in same package
A subclass can be declared successfully	no	yes	no	yes

The final and abstract modifiers are incompatible and wouldn't be used in the same declaration.

You can see that if you don't want your class to be instantiated, you can either make it abstract or use a more restrictive access modifier on the class.