Comparison of Enum, Class, Interface, Inner Class, and Nested Class in Java

Feature	Enum	Class	Interface	Inner Class	Nested Class
Definition	A special data type that represents a fixed set of constants.	A blueprint for creating objects (instances).	A reference type that defines a contract for classes to implement.	A class defined within another class, can access its enclosing class's members.	A class defined within another class, can be static or non-static.
Instantiation	Cannot be instantiated directly; uses predefined constants.	Can be instantiated using the new keyword.	Cannot be instantiated directly; must be implemented by a class.	Can be instantiated like a normal class or via its enclosing class.	Can be instantiated like a normal class if non-static; static nested classes are instantiated independently.
Access Modifiers	Implicitly public, static, and final.	Can have various access modifiers (public, private, etc.).	Cannot have access modifiers; all members are implicitly public.	Can have access modifiers; depends on its enclosing class.	Can have access modifiers; can be static or non-static.
Members	Can have fields, methods, and constructors.	Can have fields, methods, constructors, and blocks.	Can have abstract methods, default methods, static methods, and constants.	Can have fields, methods, and constructors.	Can have fields, methods, and constructors.
Purpose	To define a set of related constants.	To model real-world entities and behaviors.	To define a contract for implementing classes.	To logically group classes that are closely related to their enclosing class.	To logically group classes, can be independent of the enclosing class.
Use Cases	Days of the week, states of a process, etc.	Represents objects, encapsulates data and behavior.	API design, callbacks, event handling, etc.	GUI components, utility classes related to another class.	Utility classes, helper classes, or configurations related to the enclosing class.