Abstraction

[...]

Multiple Implementations

Java provides two mechanisms for defining a type that permits multiple implementations – abstract classes and interfaces.

Java 8 allows interfaces to have a default implementation.

Abstract Classes

An abstract class can be subclasses, but they can't be instantiated.

An abstract class may or may not include abstract methods, which are methods without an implementation.

When to Use Abstract Classes?

When you want to share code among related classes.

Sub-classes will have many common methods or fields.

[...]

Single Inheritance

Java only permits single inheritance, which limits the use of abstract classes as type definitions. (E.g. How do you label a type as a combination of types?)

Interfaces

Interfaces allow you to separate a class' interface from its implementation – interface specifies method signatures but not how they're implemented.

Any class that defines all the required methods, it is allowed to implement an interface, regardless of where the class resides in the class hierarchy.

Hierarchical Interfaces

Interfaces can extend each other and inherit all the method signatures.

Three ways of working with default methods:

- don't mention the default method, which lets the implementation be inherited
- redeclare the default method, making it abstract
- redefine the default method, overriding it

Multiple Inheritance (ish)

Classes can implement multiple interfaces, and interfaces can extend multiple other interfaces.

Retrofitting

Existing classes can easily be retrofitted to implement an existing interface.

Mixins

Mixins define extension functionality without predetermining what exactly is extended.

A class can choose to implement them to provide additional behaviour.

Non-hierarchical Frameworks

Can have very different classes that implement a particular interface.

Default Methods

Java 8 brings in default methods for interfaces. However, they still can only refer to other interface methods, and not state.

Additional Requirements

Writing method signatures requires you to see the methods from the user's point of view.

Usually you also need to include comments specifying how each method is used.