

Classes Implementing Interfaces

A Slice class can also be used as a servant in a server, that is, an instance of a class can be used to provide the behavior for an interface, for example:

Slice

```
interface Time {
    idempotent TimeOfDay getTime();
    idempotent void setTime(TimeOfDay time);
};

class Clock implements Time {
    TimeOfDay time;
};
```

The `implements` keyword indicates that the class `Clock` provides an *implementation* of the `Time` interface. The class can provide data members and operations of its own; in the preceding example, the `Clock` class stores the current time that is accessed via the `Time` interface. A class can implement several interfaces, for example:

Slice

```
interface Time {
    idempotent TimeOfDay getTime();
    idempotent void setTime(TimeOfDay time);
};

interface Radio {
    idempotent void setFrequency(long hertz);
    idempotent void setVolume(long dB);
};

class RadioClock implements Time, Radio {
    TimeOfDay time;
    long hertz;
};
```

The class `RadioClock` implements both `Time` and `Radio` interfaces.

A class, in addition to implementing an interface, can also extend another class:

Slice

```

interface Time {
    idempotent TimeOfDay getTime();
    idempotent void setTime(TimeOfDay time);
};

class Clock implements Time {
    TimeOfDay time;
};

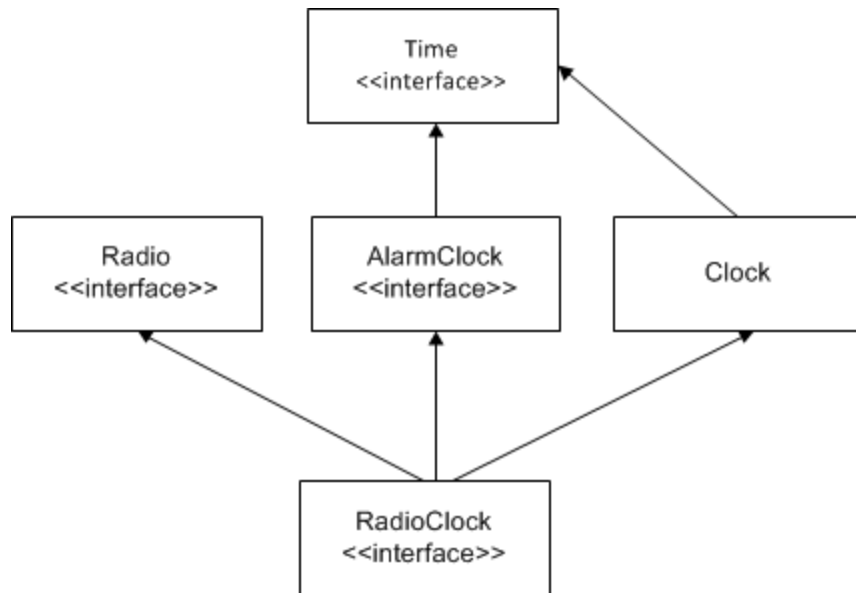
interface AlarmClock extends Time {
    idempotent TimeOfDay getAlarmTime();
    idempotent void setAlarmTime(TimeOfDay alarmTime);
};

interface Radio {
    idempotent void setFrequency(long hertz);
    idempotent void setVolume(long dB);
};

class RadioAlarmClock extends Clock
    implements AlarmClock, Radio {
    TimeOfDay alarmTime;
    long hertz;
};

```

These definitions result in the following inheritance graph:



A Class using implementation and interface inheritance.

For this definition, **Radio** and **AlarmClock** are abstract interfaces, and **Clock** and **RadioAlarmClock** are concrete classes. As for Java, a class can implement multiple interfaces, but can extend at most one class.

See Also

- [Architectural Implications of Classes](#)
- [Class Inheritance Limitations](#)