Interfaces

In Kotlin, the way we work with *interfaces* is the same as in Java. Let's take a look at example *Interfaces.kt*:

```
interface Audio {
    fun volumeUp();
    fun volumeDown();
    fun setVolume(volume: Int)
}

class Tv : Audio {
    override fun volumeUp() {
        println("Volume up")
    }

    override fun volumeDown() {
        println("Volume down")
    }

    override fun setVolume(volume: Int) {
        println("Set volume to $volume")
    }
}
```

Interfaces can have *properties*. *Property* can be abstract or with value assigned.

```
interface Video {
    val brightness: Int

    val contrast: Int
        get() = 100

    fun play() {
        println("Play")
    }

    fun pause()
}

open class MultimediaDevice : Audio {
    override fun volumeUp() {
        println("Volume up")
    }

    override fun volumeDown() {
        println("Volume down")
    }
    override fun setVolume(volume: Int) {
        println("Set volume to $volume")
    }
}

class DvdPlayer : Tv(), Video {
    override val brightness: Int = 100

    override fun pause() {
        println("Pause")
    }
}
```

Pay attention that *interfaces* in Kotlin can have methods with implementation.