Classes

In Kotlin class is defined using the same word we use in Java: *class*.

Let's go through *ClassExamples.kt*:

```
// Simple class definition
class Dummy {}

// If there is no body like in previous example
class NoBody
```

Constructors

In Kotlin, class can have a primary constructor and one or more secondary constructors. The primary constructor is a part of class header. **If the primary constructor does not have any annotations or visibility modifiers, the constructor keyword can be omitted.**

```
class Car constructor(val brand: String)
class Plane(val brand: String)

val car = Car("BMW")
val plane = Plane("Boeing")
println(car.brand)
println(plane.brand)
```

Console output:

```
BMW
Boeing
```

The primary constructor can't contain any code. Initialization code can be placed in initializer blocks, which are prefixed with the init keyword. Take a look at example:

```
class Calculator(val parameter: Int){
   val calcualtedValue: Int
   init {
      calcualtedValue = parameter * 2
   }
}

val calculator = Calculator(2)
println("Calculated value: ${calculator.calcualtedValue}")
```

Console output:

```
Calculated value: 4
```

Let's say that we don't want class to be constructed publicly, we can do it the following way:

```
class NotPublicConstructed private constructor(val name: String){
   // ...
}
```

Secondary constructors

Let's take a look at the following example:

```
class Person(val name: String) {
   constructor(name: String, year: Int) : this(name) {}
   constructor(name: String, year: Int, height: Int) : this(name) {}
}

val person1 = Person("John Smith")
val person2 = Person("John Doe", 1985)
val person3 = Person("John SomeOther", 1987, 190)
```

The code demonstrates usage of secondary constructor in Kotlin. If we didn't define primary constructor, class will have default – empty constructor, with no arguments.

We instantiated three instances of *Person class*. Each with different constructor. You can also notice that Kotlin does not have word *new* for instantiation.

Members

In Kotlin classes can have the following members:

- Constructors
- Initialization blocks
- Functions
- Properties
- Nested classes
- Inner classes
- Misc object declarations – for example: *companion object*.

Inheritance

In Java, all classes (who do not extend some other class) are extending *Object* super class. Kotlin has its super class called ***Any***. From *Any* class we inherit the following methods:

- *equals()*
- *hashCode()*
- *toString()*