Sealed Classes

45

```kotlin
sealed interface Account {
    val iban: String
    val balance: BigDecimal
}
```

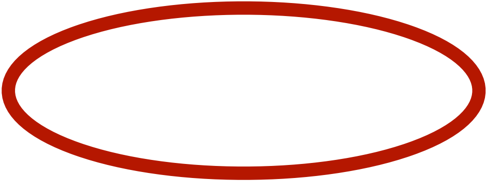
```kotlin
data class SavingsAccount(
    override val iban: String,
    override val balance: BigDecimal,
    val interest: BigDecimal
) : Account
```

```kotlin
data class CreditAccount(
    override val iban: String,
    override val balance: BigDecimal,
    val creditLimit: BigDecimal,
    val interest: BigDecimal
) : Account
```

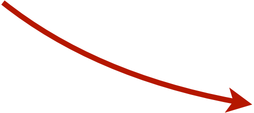
```kotlin
data class CurrentAccount(
    override val iban: String,
    override val balance: BigDecimal
) : Account
```

```kotlin
object TechnicalAccount : Account {
    override val iban: String =
        "RO99TECH1234567812345678"
    override var balance: BigDecimal = ZERO
}
```

starting with Kotlin 1.5 there are also **sealed interfaces** the sealed class/interface and all implementations have to be in the **same package and module**



Objects as implementation



Sealed Classes

starting with Kotlin 1.5 there are also **sealed interfaces**

```kotlin
sealed interface Account {
    val iban: String
    val balance: BigDecimal
}
```

the sealed class/interface and all implementations have to be in the **same package and module**

```kotlin
data class CurrentAccount(
    override val iban: String,
    override val balance: BigDecimal
) : Account

data class CreditAccount(
    override val iban: String,
    override val balance: BigDecimal,
    val creditLimit: BigDecimal,
    val interest: BigDecimal
) : Account
```

```kotlin
data class SavingsAccount(
    override val iban: String,
    override val balance: BigDecimal,
    val interest: BigDecimal
) : Account

object TechnicalAccount : Account {
    override val iban: String =
        "RO99TECH1234567812345678"
    override var balance: BigDecimal = ZERO
}
```

Objects as implementation

Sealed Classes

when is exhaustive without else