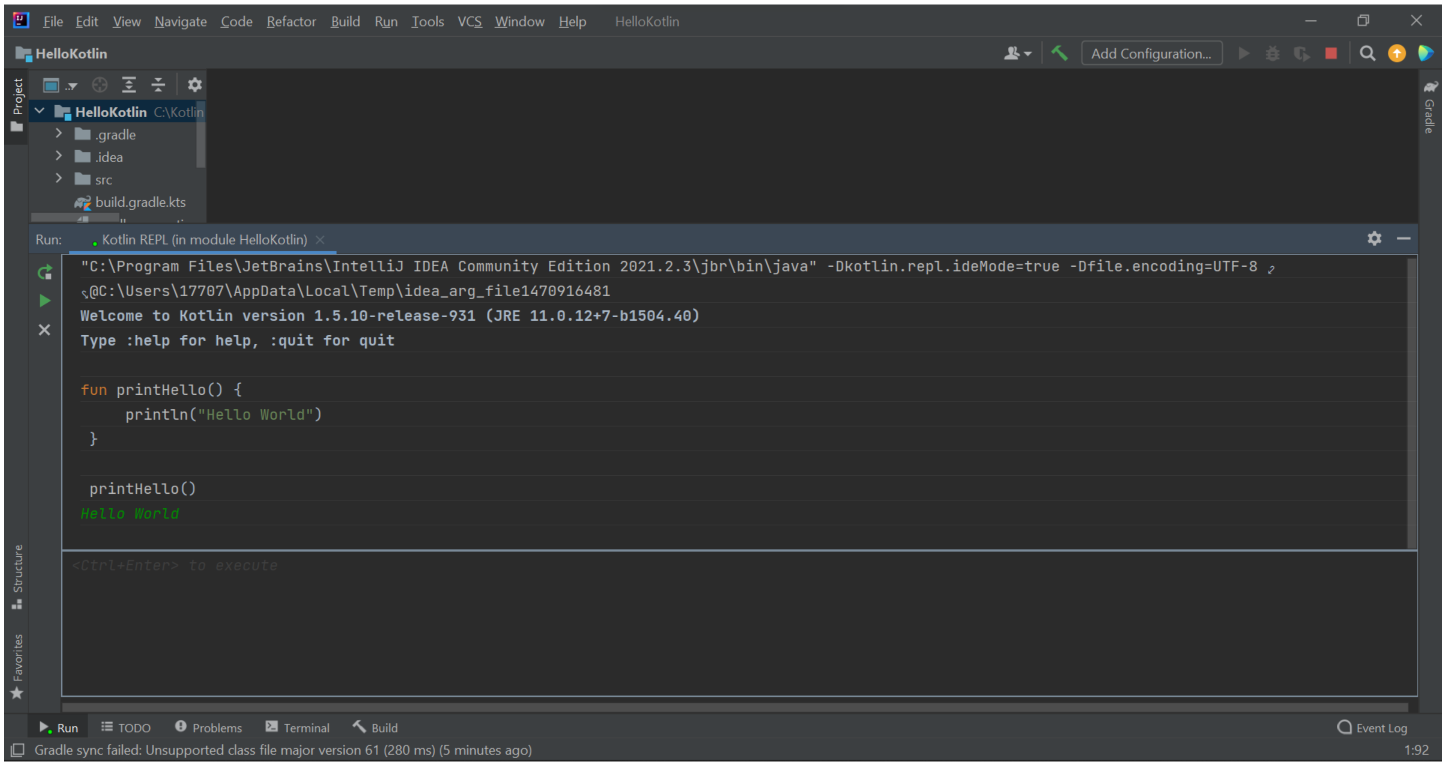
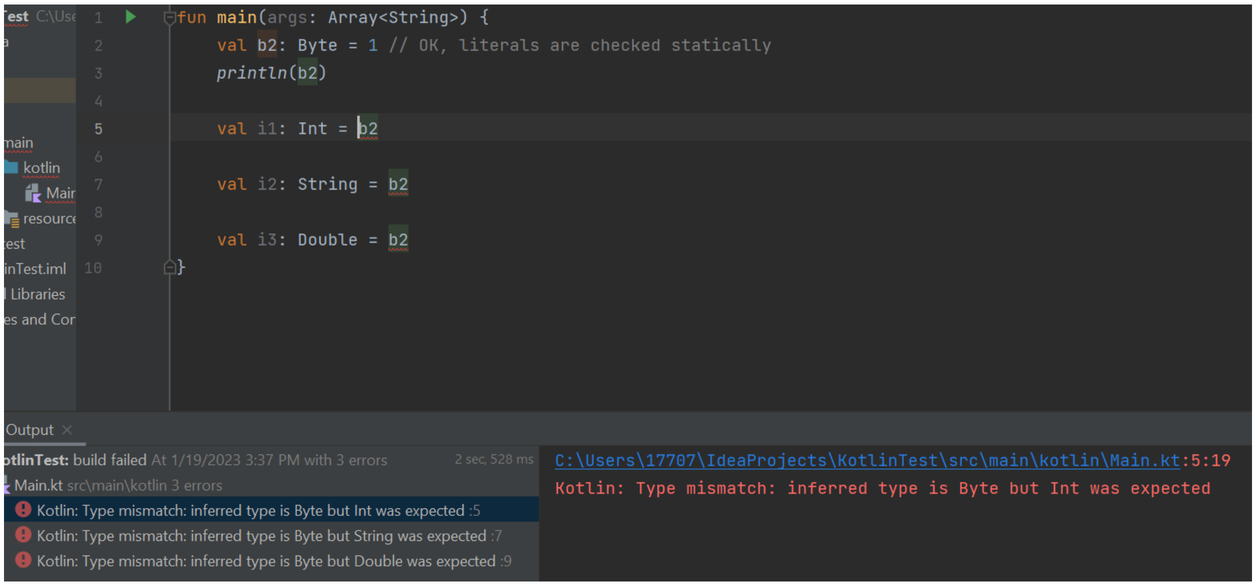

File Edit View Navigate Code Refactor Build Run Tools VCS Window Help HelloKotlin
HelloKotlin
Add Configuration...
HelloKotlin C:\Kotlin
.gradle
.idea
src
build.gradle.kts
Run: Kotlin REPL (in module HelloKotlin)
Run TODO Problems Terminal Build Event Log

```
"C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2021.2.3\jbr\bin\java" -Dkotlin.repl.ideMode=true -Dfile.encoding=UTF-8
@C:\Users\17707\AppData\Local\Temp\idea_arg_file1470916481
Welcome to Kotlin version 1.5.10-release-931 (JRE 11.0.12+7-b1504.40)
Type :help for help, :quit for quit

fun printHello() {
     println("Hello World")
 }

 printHello()
Hello World
```

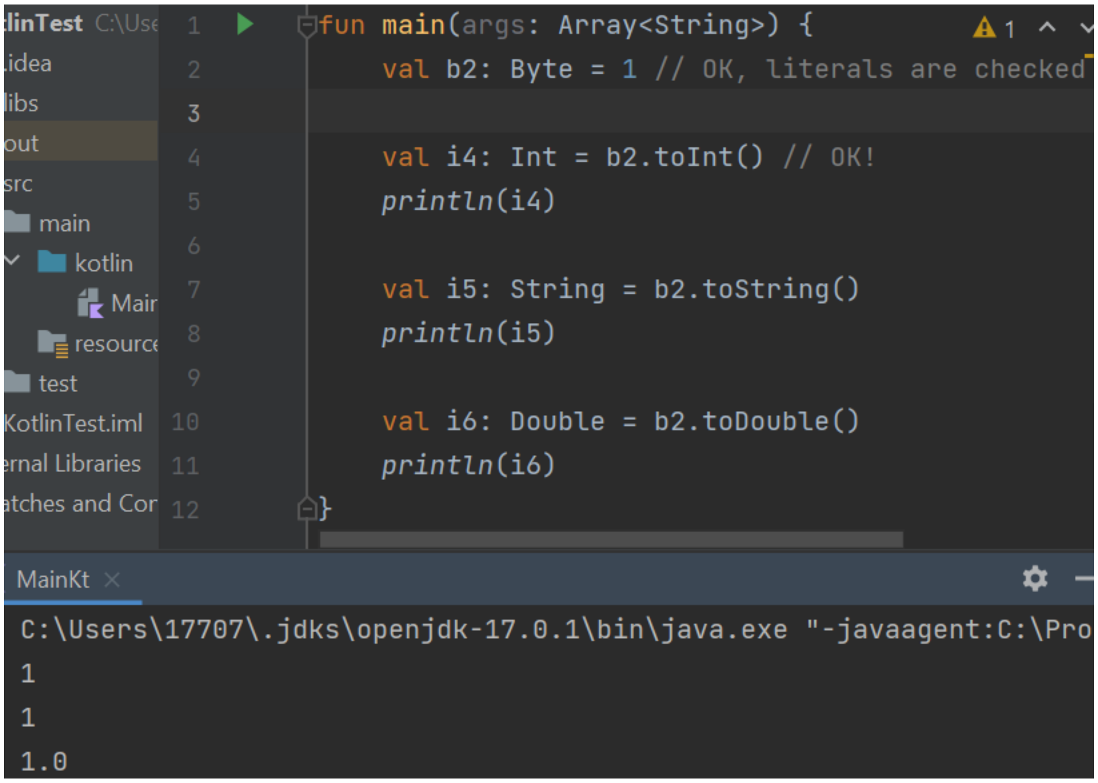
```
<Ctrl+Enter> to execute
```

```
Gradle sync failed: Unsupported class file major version 61 (280 ms) (5 minutes ago)
```



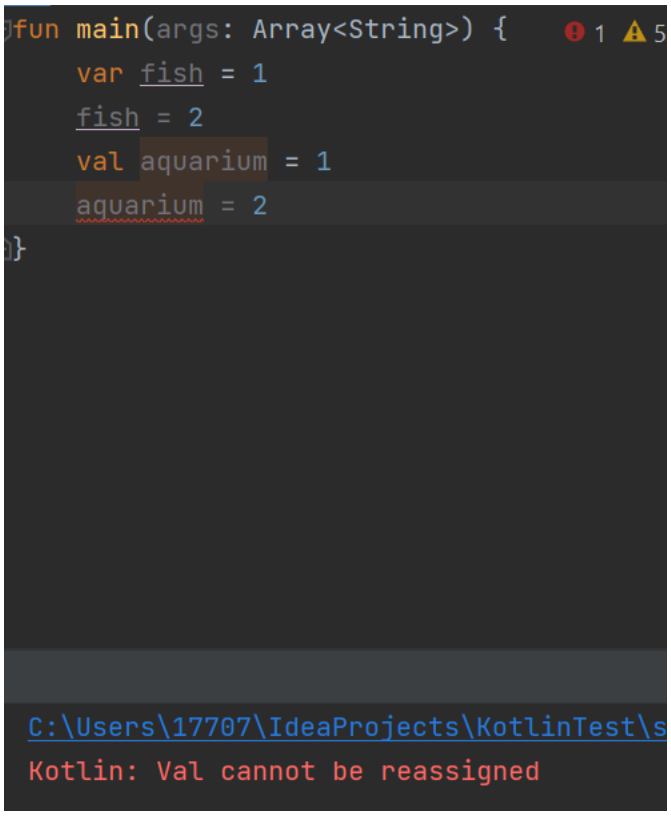
```
fun main(args: Array<String>) {
    val b2: Byte = 1 // OK, literals are checked statically
    println(b2)

    val i1: Int = b2

    val i2: String = b2

    val i3: Double = b2
}
```

```
otlinTest: build failed At 1/19/2023 3:37 PM with 3 errors        2 sec, 528 ms
 Main.kt src\main\kotlin 3 errors
  Kotlin: Type mismatch: inferred type is Byte but Int was expected :5
  Kotlin: Type mismatch: inferred type is Byte but String was expected :7
  Kotlin: Type mismatch: inferred type is Byte but Double was expected :9

C:\Users\17707\IdeaProjects\KotlinTest\src\main\kotlin\Main.kt:5:19
Kotlin: Type mismatch: inferred type is Byte but Int was expected
```



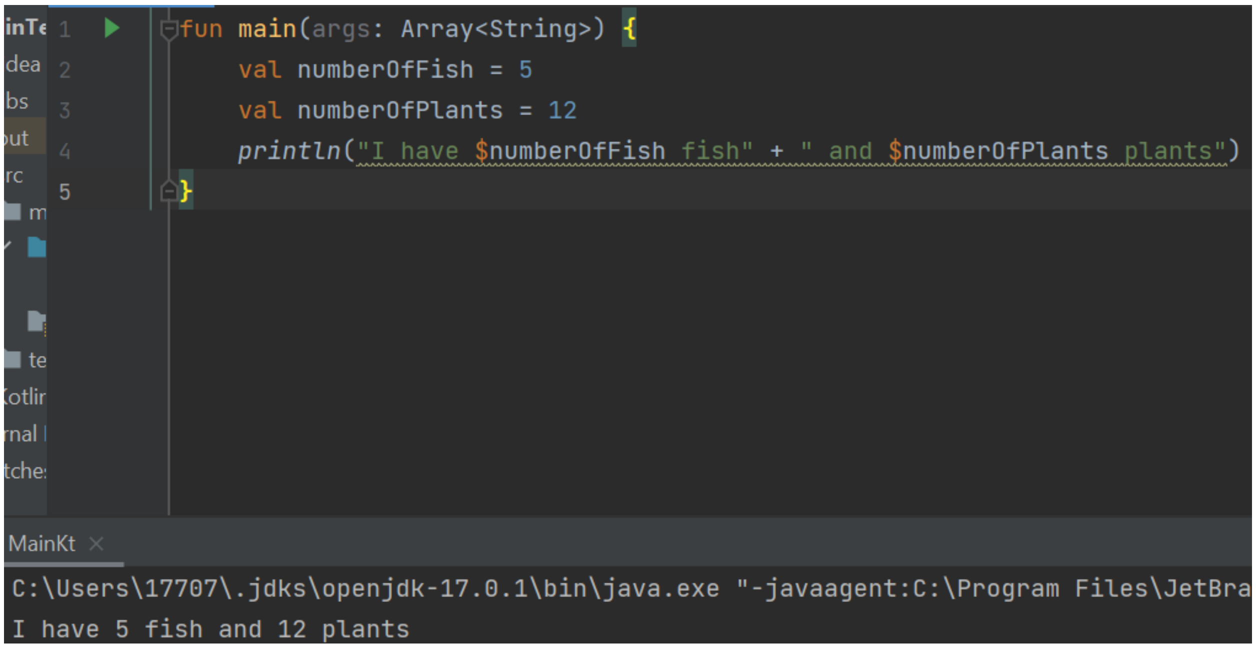
```kotlin
fun main(args: Array<String>) {
    val b2: Byte = 1 // OK, literals are checked

    val i4: Int = b2.toInt() // OK!
    println(i4)

    val i5: String = b2.toString()
    println(i5)

    val i6: Double = b2.toDouble()
    println(i6)
}
```

```
C:\Users\17707\.jdks\openjdk-17.0.1\bin\java.exe "-javaagent:C:\Pro
1
1
1.0
```



```kotlin
fun main(args: Array<String>) {
    var fish = 1
    fish = 2
    val aquarium = 1
    aquarium = 2
}
```

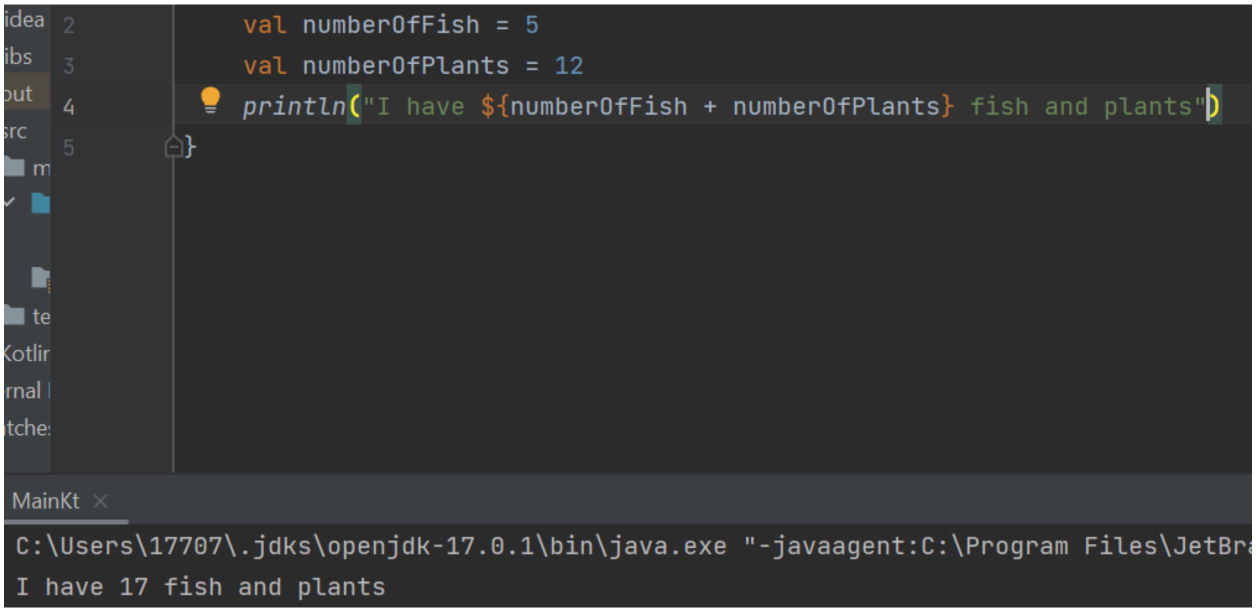
```
C:\Users\17707\IdeaProjects\KotlinTest\s
Kotlin: Val cannot be reassigned
```



```
fun main(args: Array<String>) {
    val numberOfFish = 5
    val numberOfPlants = 12
    println("I have $numberOfFish fish" + " and $numberOfPlants plants")
}
```

MainKt

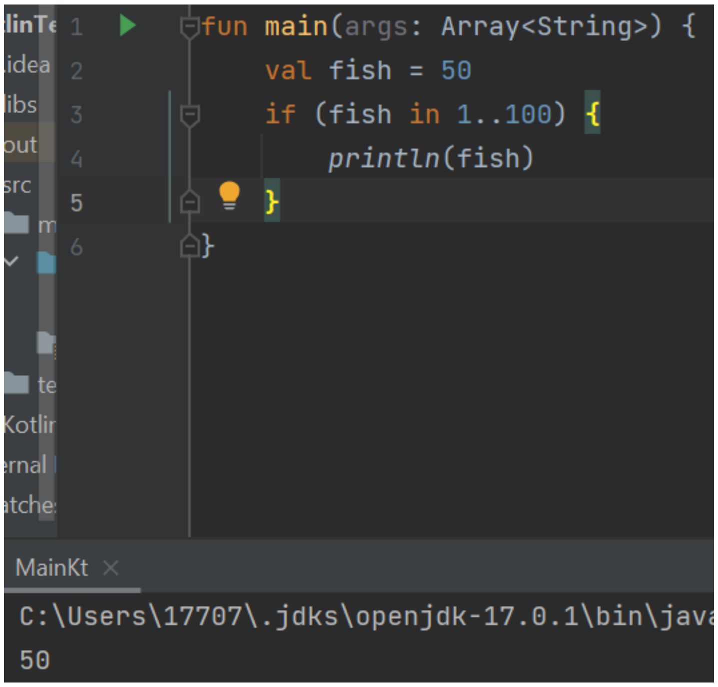
```
C:\Users\17707\.jdks\openjdk-17.0.1\bin\java.exe "-javaagent:C:\Program Files\JetBra
I have 5 fish and 12 plants
```



```
    val numberOfFish = 5
    val numberOfPlants = 12
    println("I have ${numberOfFish + numberOfPlants} fish and plants")
}
```

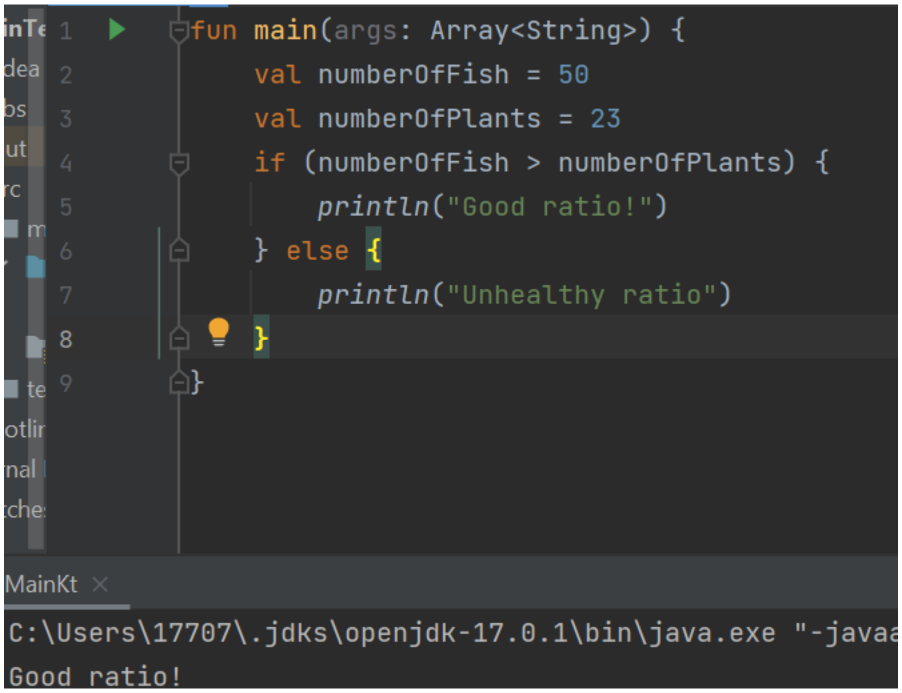
MainKt

```
C:\Users\17707\.jdks\openjdk-17.0.1\bin\java.exe "-javaagent:C:\Program Files\JetBra
I have 17 fish and plants
```



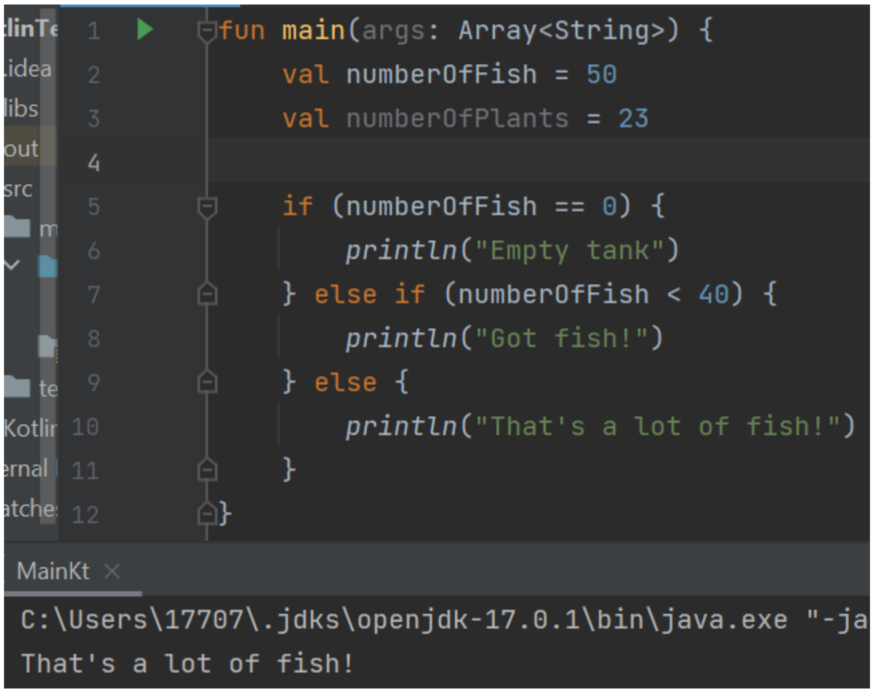
```kotlin
fun main(args: Array<String>) {
    val fish = 50
    if (fish in 1..100) {
        println(fish)
    }
}
```

```
C:\Users\17707\.jdks\openjdk-17.0.1\bin\java
50
```



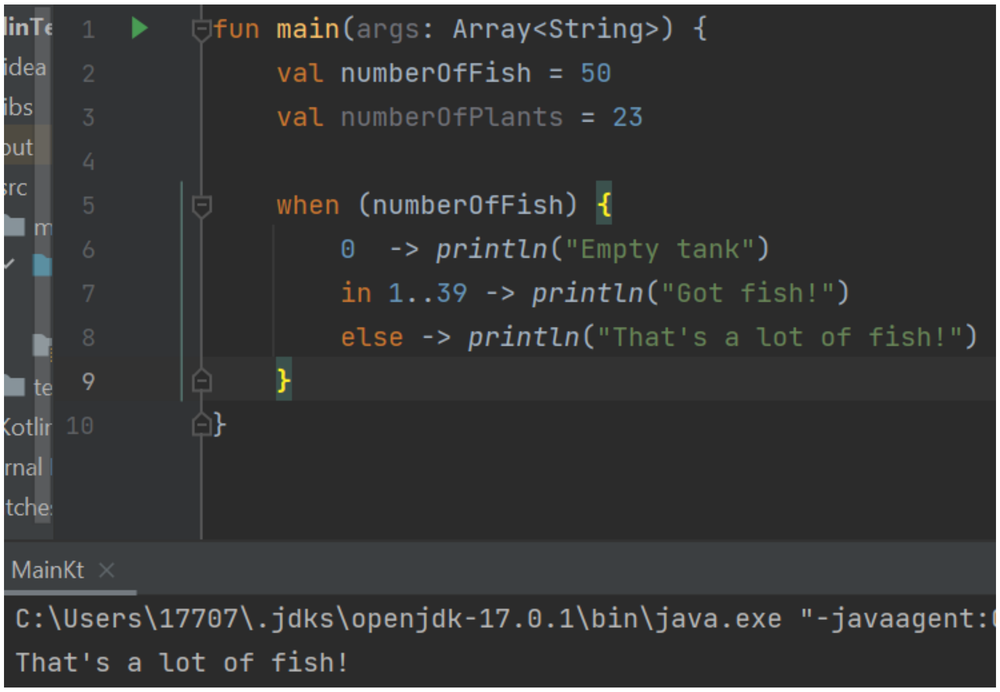
```kotlin
fun main(args: Array<String>) {
    val numberOfFish = 50
    val numberOfPlants = 23
    if (numberOfFish > numberOfPlants) {
        println("Good ratio!")
    } else {
        println("Unhealthy ratio")
    }
}
```

```
C:\Users\17707\.jdks\openjdk-17.0.1\bin\java.exe "-javaa
Good ratio!
```



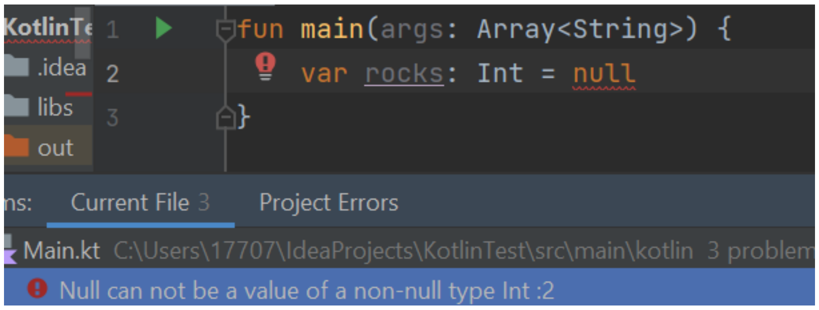
```kotlin
fun main(args: Array<String>) {
    val numberOfFish = 50
    val numberOfPlants = 23

    if (numberOfFish == 0) {
        println("Empty tank")
    } else if (numberOfFish < 40) {
        println("Got fish!")
    } else {
        println("That's a lot of fish!")
    }
}
```

```
C:\Users\17707\.jdks\openjdk-17.0.1\bin\java.exe "-ja
That's a lot of fish!
```



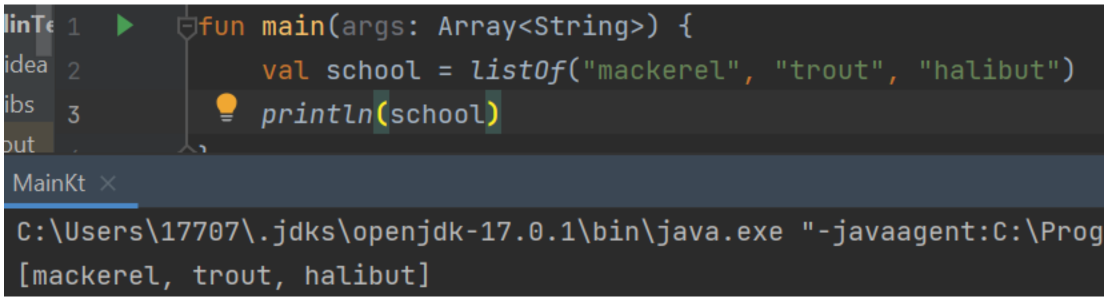
```kotlin
fun main(args: Array<String>) {
    val numberOfFish = 50
    val numberOfPlants = 23

    when (numberOfFish) {
        0  -> println("Empty tank")
        in 1..39 -> println("Got fish!")
        else -> println("That's a lot of fish!")
    }
}
```

```
C:\Users\17707\.jdks\openjdk-17.0.1\bin\java.exe "-javaagent:
That's a lot of fish!
```



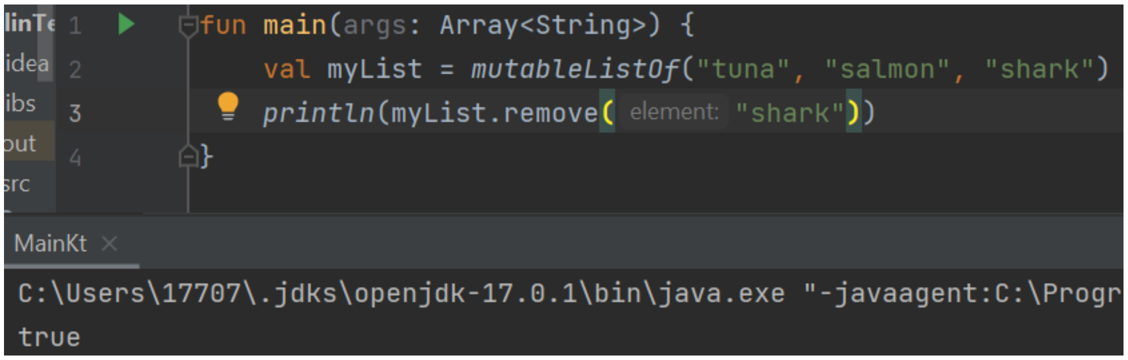
```kotlin
fun main(args: Array<String>) {
    var rocks: Int = null
}
```

```
Main.kt C:\Users\17707\IdeaProjects\KotlinTest\src\main\kotlin 3 problem
Null can not be a value of a non-null type Int :2
```



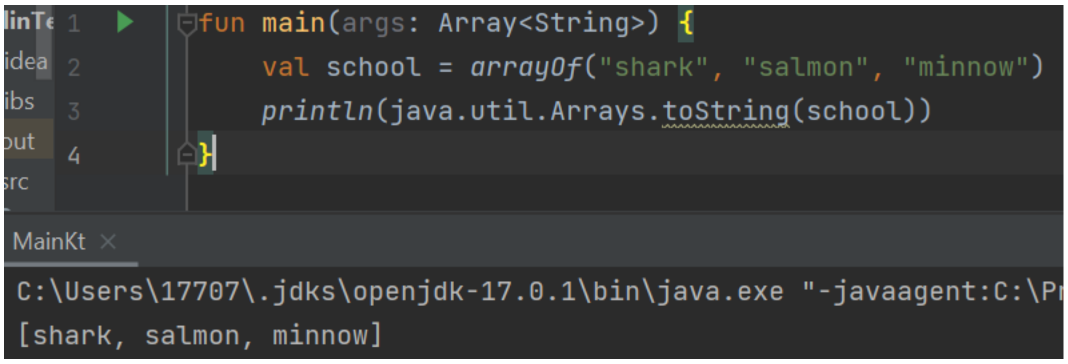
```
fun main(args: Array<String>) {
    val school = listOf("mackerel", "trout", "halibut")
    println(school)
```

```
C:\Users\17707\.jdks\openjdk-17.0.1\bin\java.exe "-javaagent:C:\Prog
[mackerel, trout, halibut]
```



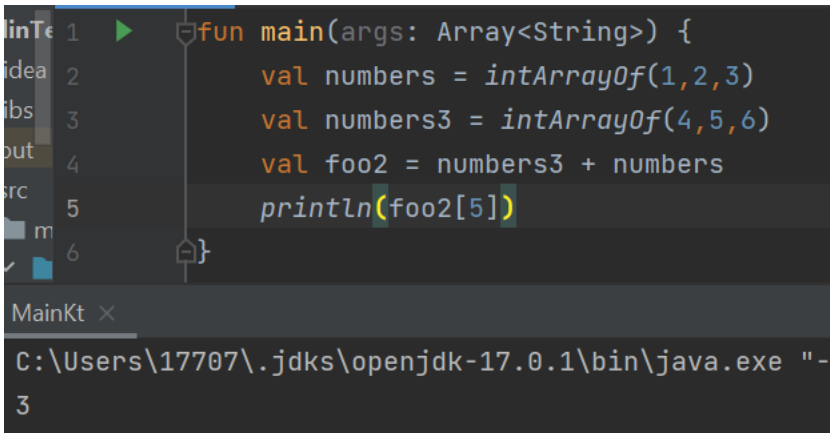
```
fun main(args: Array<String>) {
    val myList = mutableListOf("tuna", "salmon", "shark")
    println(myList.remove( element: "shark"))
}
```

```
C:\Users\17707\.jdks\openjdk-17.0.1\bin\java.exe "-javaagent:C:\Progr
true
```



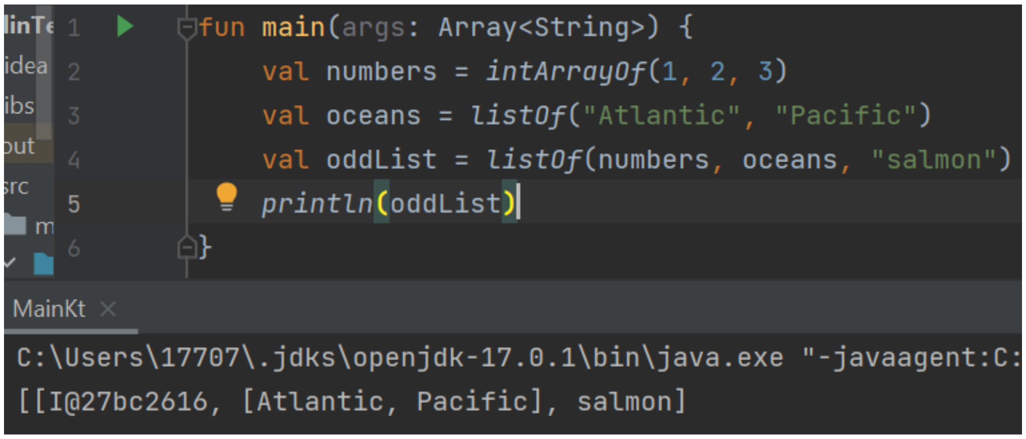
```
fun main(args: Array<String>) {
    val school = arrayOf("shark", "salmon", "minnow")
    println(java.util.Arrays.toString(school))
}
```

```
C:\Users\17707\.jdks\openjdk-17.0.1\bin\java.exe "-javaagent:C:\Pr
[shark, salmon, minnow]
```



```
fun main(args: Array<String>) {
    val numbers = intArrayOf(1,2,3)
    val numbers3 = intArrayOf(4,5,6)
    val foo2 = numbers3 + numbers
    println(foo2[5])
}
```

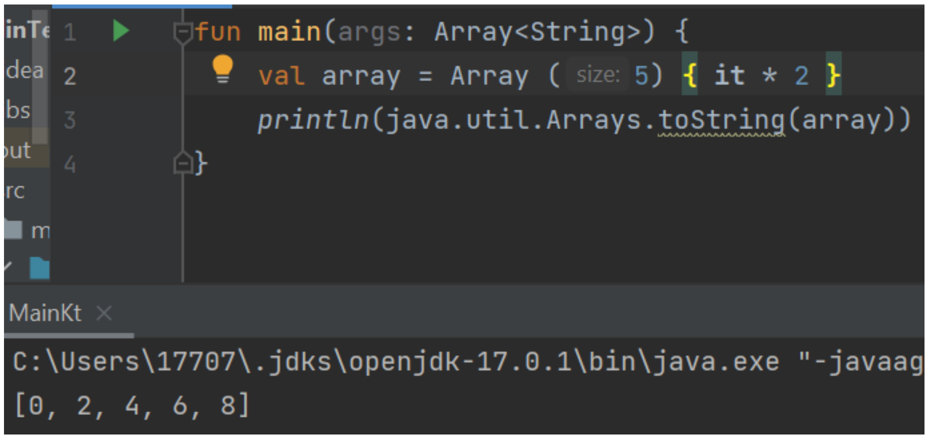
```
C:\Users\17707\.jdks\openjdk-17.0.1\bin\java.exe "-
3
```



```
fun main(args: Array<String>) {
    val numbers = intArrayOf(1, 2, 3)
    val oceans = listOf("Atlantic", "Pacific")
    val oddList = listOf(numbers, oceans, "salmon")
    println(oddList)
}
```

MainKt

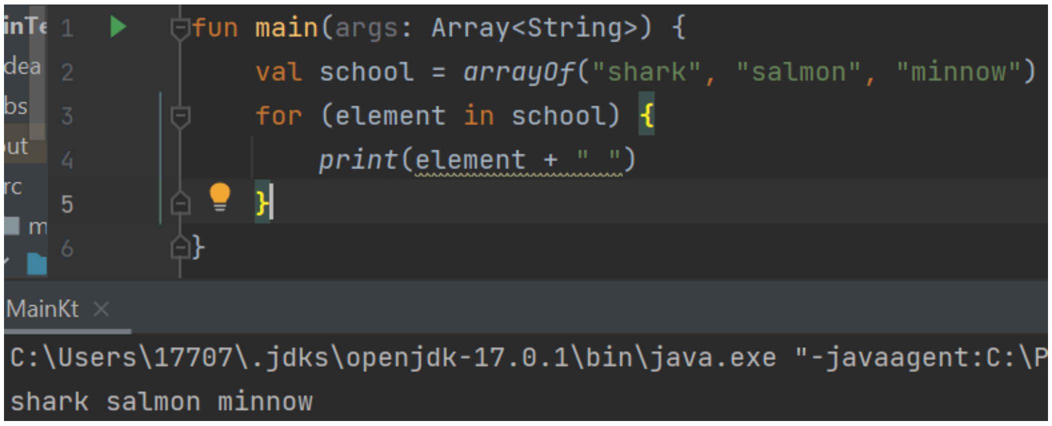
```
C:\Users\17707\.jdks\openjdk-17.0.1\bin\java.exe "-javaagent:C:
[[I@27bc2616, [Atlantic, Pacific], salmon]
```



```
fun main(args: Array<String>) {
    val array = Array ( size: 5) { it * 2 }
    println(java.util.Arrays.toString(array))
}
```

MainKt

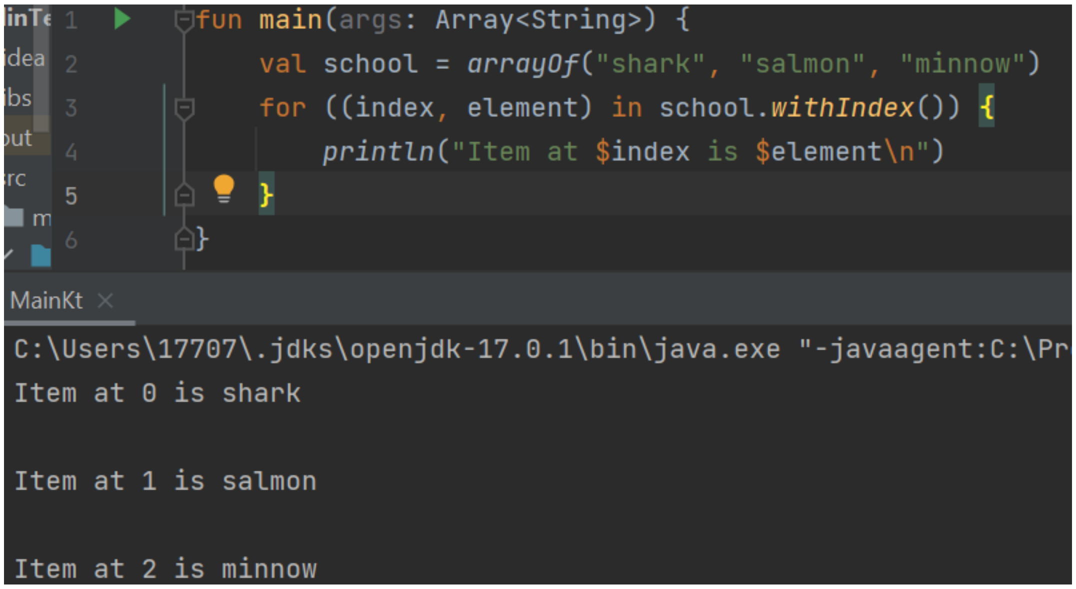
```
C:\Users\17707\.jdks\openjdk-17.0.1\bin\java.exe "-javaag
[0, 2, 4, 6, 8]
```



```
fun main(args: Array<String>) {
    val school = arrayOf("shark", "salmon", "minnow")
    for (element in school) {
        print(element + " ")
    }
}
```

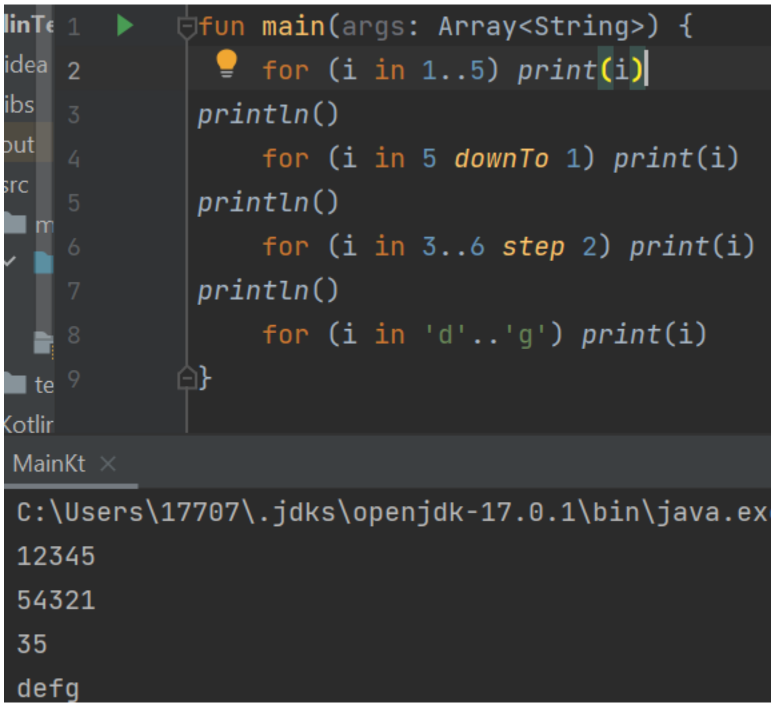
MainKt

```
C:\Users\17707\.jdks\openjdk-17.0.1\bin\java.exe "-javaagent:C:\P
shark salmon minnow
```



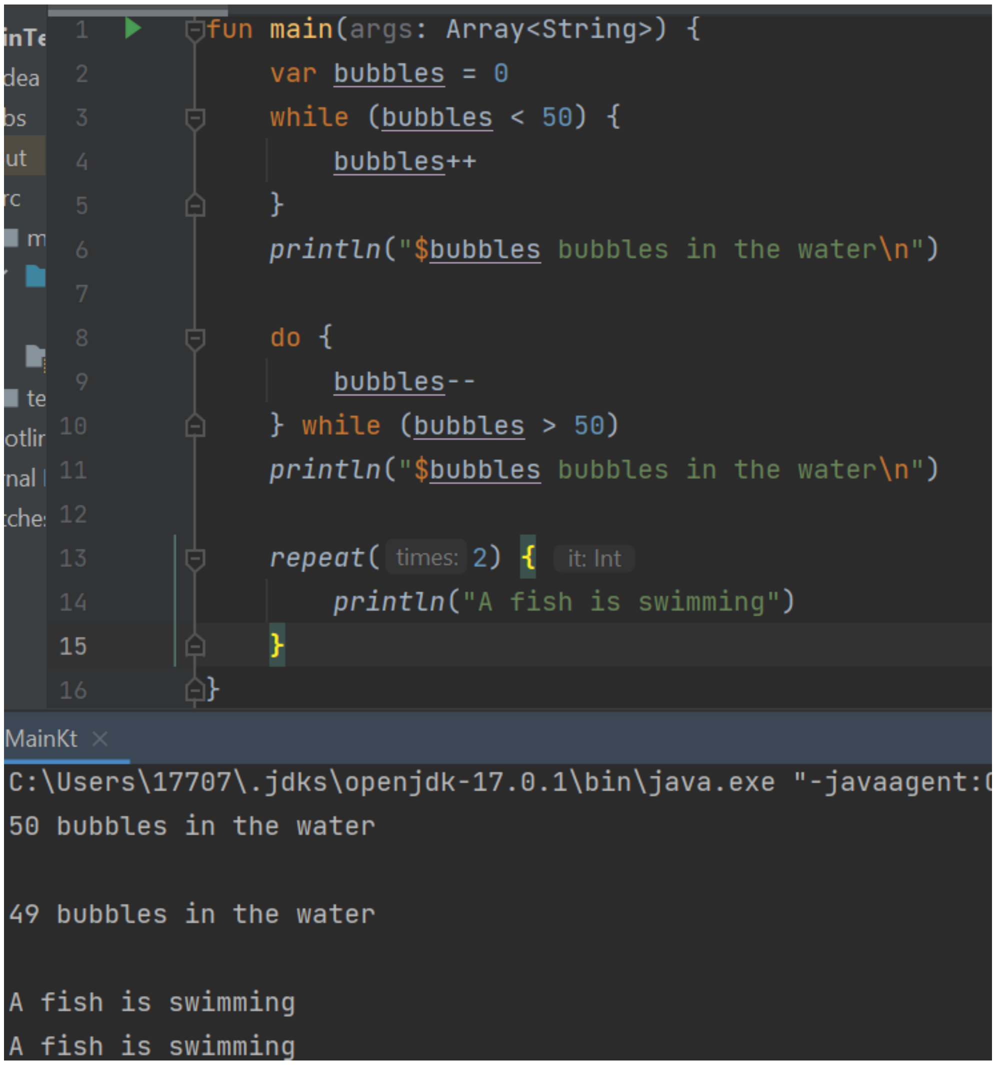
```kotlin
fun main(args: Array<String>) {
    val school = arrayOf("shark", "salmon", "minnow")
    for ((index, element) in school.withIndex()) {
        println("Item at $index is $element\n")
    }
}
```

```
C:\Users\17707\.jdks\openjdk-17.0.1\bin\java.exe "-javaagent:C:\Pr
Item at 0 is shark

Item at 1 is salmon

Item at 2 is minnow
```



```kotlin
fun main(args: Array<String>) {
    for (i in 1..5) print(i)
println()
    for (i in 5 downTo 1) print(i)
println()
    for (i in 3..6 step 2) print(i)
println()
    for (i in 'd'..'g') print(i)
}
```

```
C:\Users\17707\.jdks\openjdk-17.0.1\bin\java.ex
12345
54321
35
defg
```



```kotlin
fun main(args: Array<String>) {
    var bubbles = 0
    while (bubbles < 50) {
        bubbles++
    }
    println("$bubbles bubbles in the water\n")

    do {
        bubbles--
    } while (bubbles > 50)
    println("$bubbles bubbles in the water\n")

    repeat( times: 2) { it: Int
        println("A fish is swimming")
    }
}
```

```
C:\Users\17707\.jdks\openjdk-17.0.1\bin\java.exe "-javaagent:
50 bubbles in the water

49 bubbles in the water

A fish is swimming
A fish is swimming
```