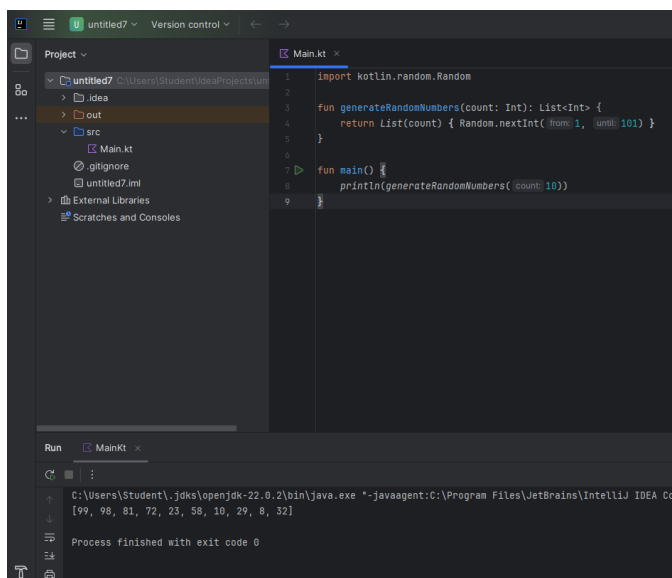


1. import kotlin.random.Random

```
fun generateRandomNumbers(count: Int): List<Int> {  
    return List(count) { Random.nextInt(1, 101) }  
}
```

```
fun main() {  
    println(generateRandomNumbers(10))  
}
```



2. fun analyzeString(text: String): Pair<Int, Int> {

```
    val vowels = "аеёиоуыэюяАЕЁИОУЫЭЮЯ"
```

```
    var vowelCount = 0
```

```
    var consonantCount = 0
```

```
    for (char in text) {  
        if (char.isLetter()) {  
            if (vowels.contains(char)) {  
                vowelCount++  
            } else {
```

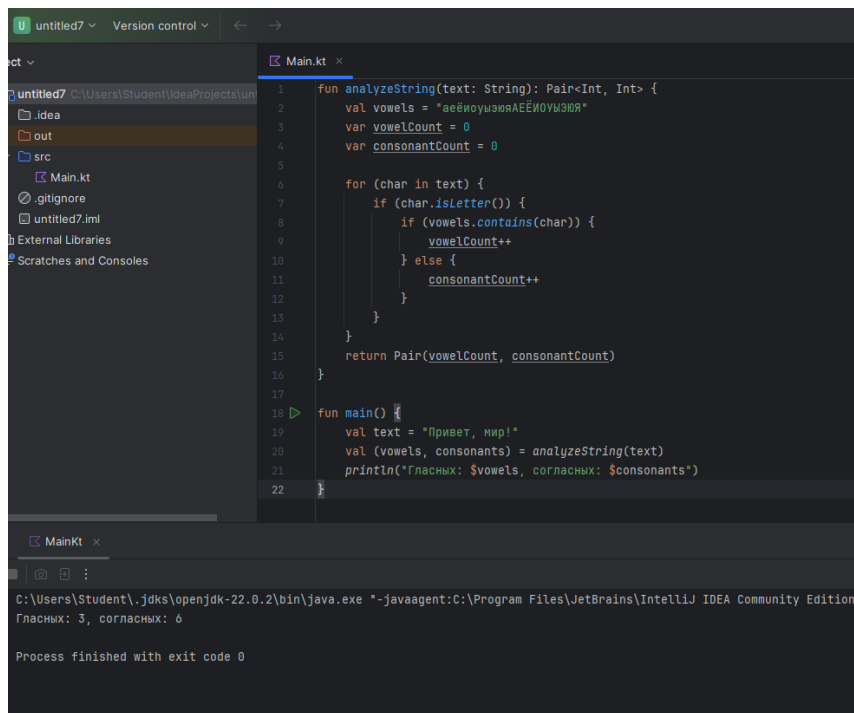
```

        consonantCount++
    }
}

return Pair(vowelCount, consonantCount)
}

fun main() {
    val text = "Привет, мир!"
    val (vowels, consonants) = analyzeString(text)
    println("Гласных: $vowels, согласных: $consonants")
}

```



```

3. fun convertCurrency(amount: Double, rate: Double): Double {
    return amount * rate
}

```

```

fun main() {
    val dollars = 100.0
    val euroRate = 0.92
}

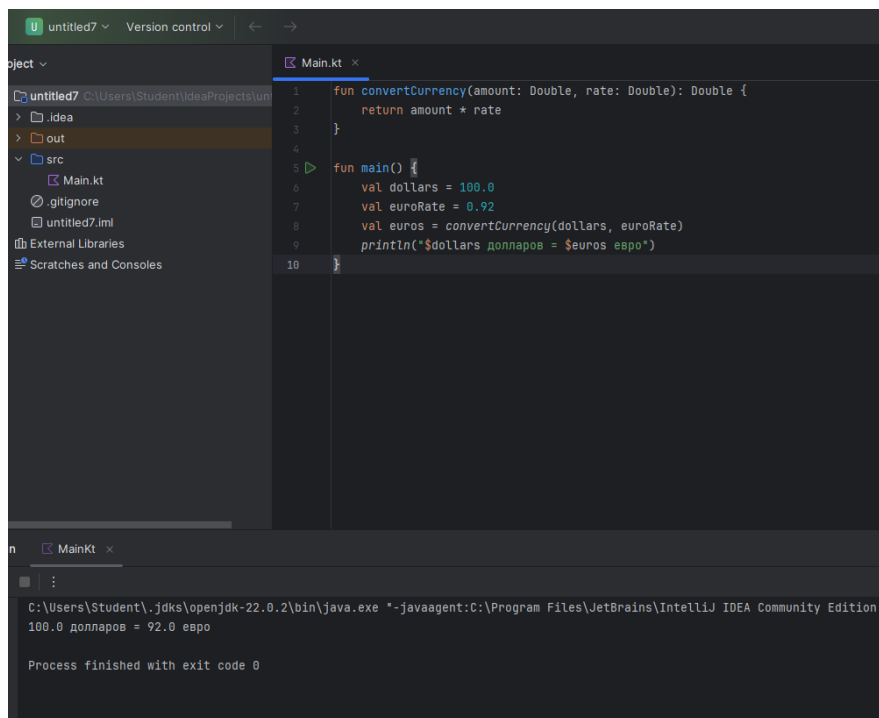
```

```

    val euros = convertCurrency(dollars, euroRate)

    println("$dollars долларов = $euros евро")
}

```



```

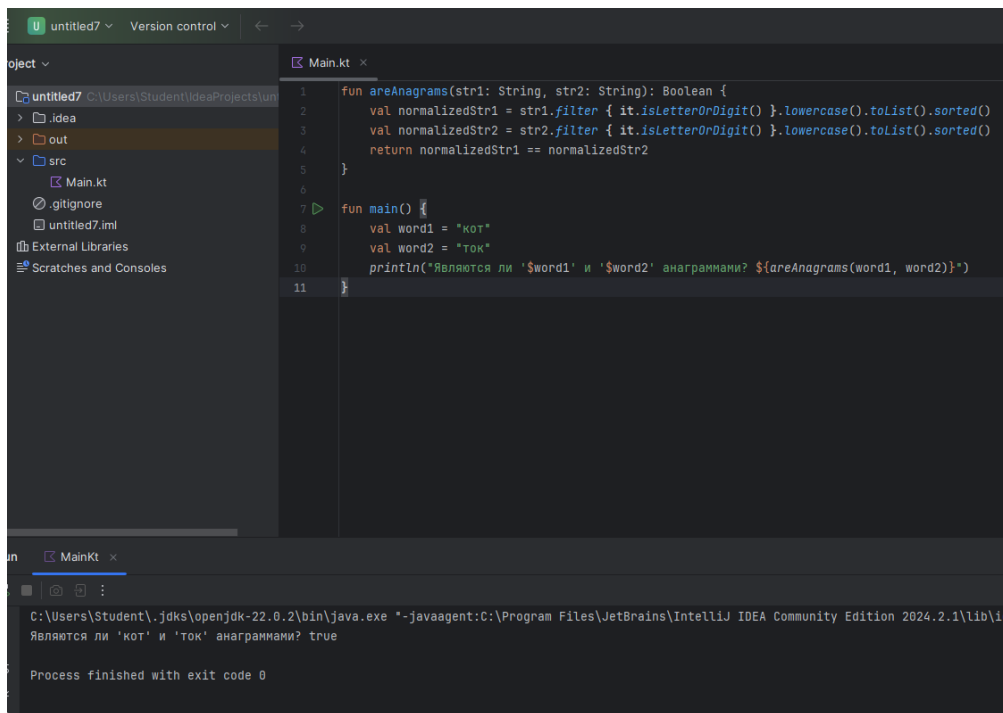
4. fun areAnagrams(str1: String, str2: String): Boolean {
    val normalizedStr1 = str1.filter { it.isLetterOrDigit() }.lowercase().toList().sorted()
    val normalizedStr2 = str2.filter { it.isLetterOrDigit() }.lowercase().toList().sorted()
    return normalizedStr1 == normalizedStr2
}

```

```

fun main() {
    val word1 = "кот"
    val word2 = "ток"
    println("Являются ли '$word1' и '$word2' анаграммами? ${areAnagrams(word1, word2)}")
}

```



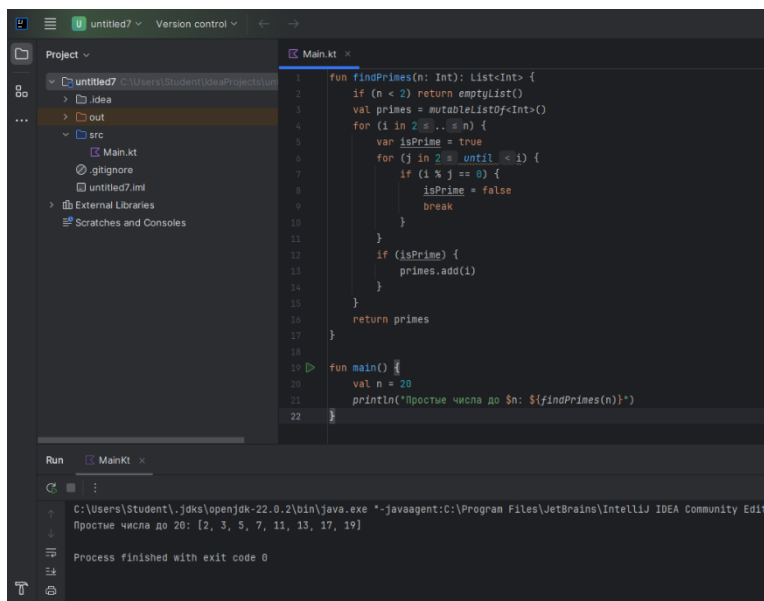
```
5. fun findPrimes(n: Int): List<Int> {  
    if (n < 2) return emptyList()  
    val primes = mutableListOf<Int>()  
    for (i in 2..n) {  
        var isPrime = true  
        for (j in 2 until i) {  
            if (i % j == 0) {  
                isPrime = false  
                break  
            }  
        }  
        if (isPrime) {  
            primes.add(i)  
        }  
    }  
    return primes  
}
```

```
fun main() {
```

```
val n = 20
```

```
println("Простые числа до $n: ${findPrimes(n)}")
```

```
}
```



```
6. fun sortString(text: String): String {
```

```
    return text.toList().sorted().joinToString("")
```

```
}
```

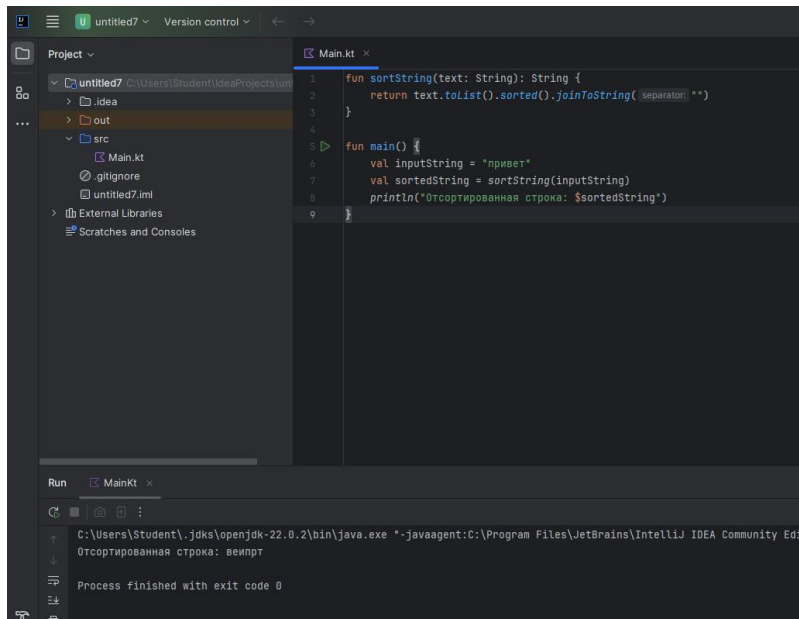
```
fun main() {
```

```
    val inputString = "привет"
```

```
    val sortedString = sortString(inputString)
```

```
    println("Отсортированная строка: $sortedString")
```

```
}
```



```
7. fun invertCase(text: String): String {
```

```
    val result = StringBuilder()
```

```
    for (char in text) {
```

```
        if (char.isLetter()) {
```

```
            if (char.isUpperCase()) {
```

```
                result.append(char.lowercaseChar())
```

```
            } else {
```

```
                result.append(char.uppercaseChar())
```

```
            }
```

```
        } else {
```

```
            result.append(char)
```

```
        }
```

```
    }
```

```
    return result.toString()
```

```
}
```

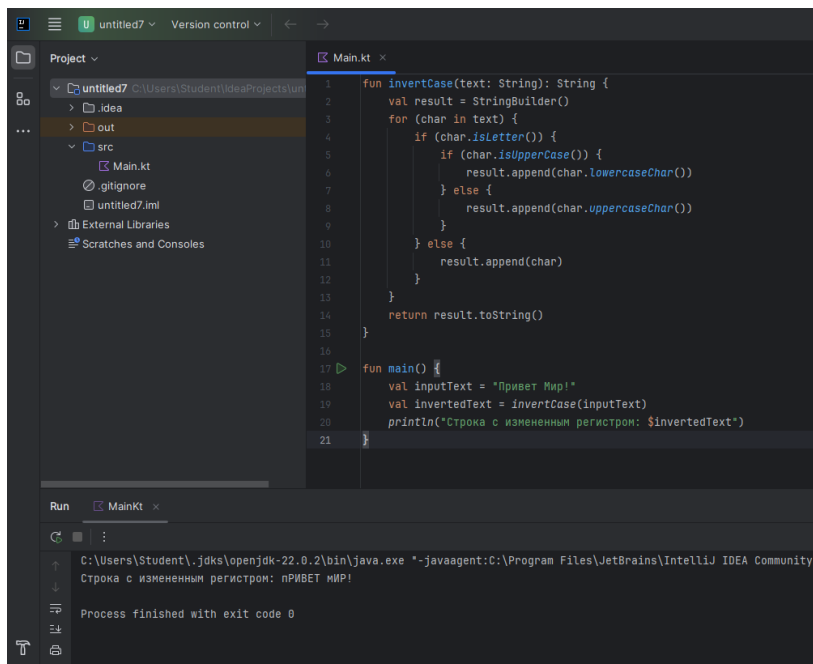
```
fun main() {
```

```
    val inputText = "Привет Мир!"
```

```
    val invertedText = invertCase(inputText)
```

```
    println("Строка с измененным регистром: $invertedText")
```

```
}
```



## 8. import kotlin.random.Random

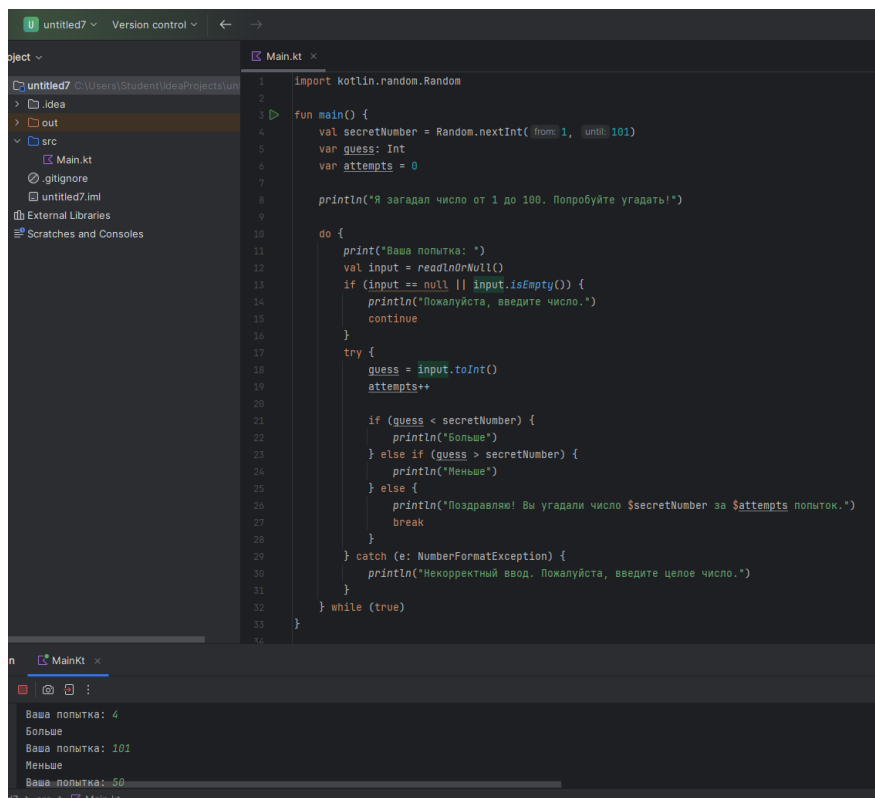
```
fun main() {  
  
    val secretNumber = Random.nextInt(1, 101)  
  
    var guess: Int  
  
    var attempts = 0  
  
    println("Я загадал число от 1 до 100. Попробуйте угадать!")  
  
    do {  
  
        print("Ваша попытка: ")  
  
        val input = readlnOrNull()  
  
        if (input == null || input.isEmpty()) {  
  
            println("Пожалуйста, введите число.")  
  
            continue  
  
        }  
  
        try {  
  
            guess = input.toInt()  
  
            attempts++  
  

```

```

        if (guess < secretNumber) {
            println("Больше")
        } else if (guess > secretNumber) {
            println("Меньше")
        } else {
            println("Поздравляю! Вы угадали число $secretNumber за $attempts попыток.")
            break
        }
    } catch (e: NumberFormatException) {
        println("Некорректный ввод. Пожалуйста, введите целое число.")
    }
} while (true)
}

```



9. `import kotlin.random.Random`

```

fun generatePassword(length: Int): String {
    val allowedChars = ('A'..'Z') + ('a'..'z') + ('0'..'9') + listOf('!', '@', '#', '$', '%', '^', '&', '*')
    return (1..length)
}

```



```

        .map { allowedChars[Random.nextInt(allowedChars.size)] }

        .joinToString("")

    }

```

```

fun main() {

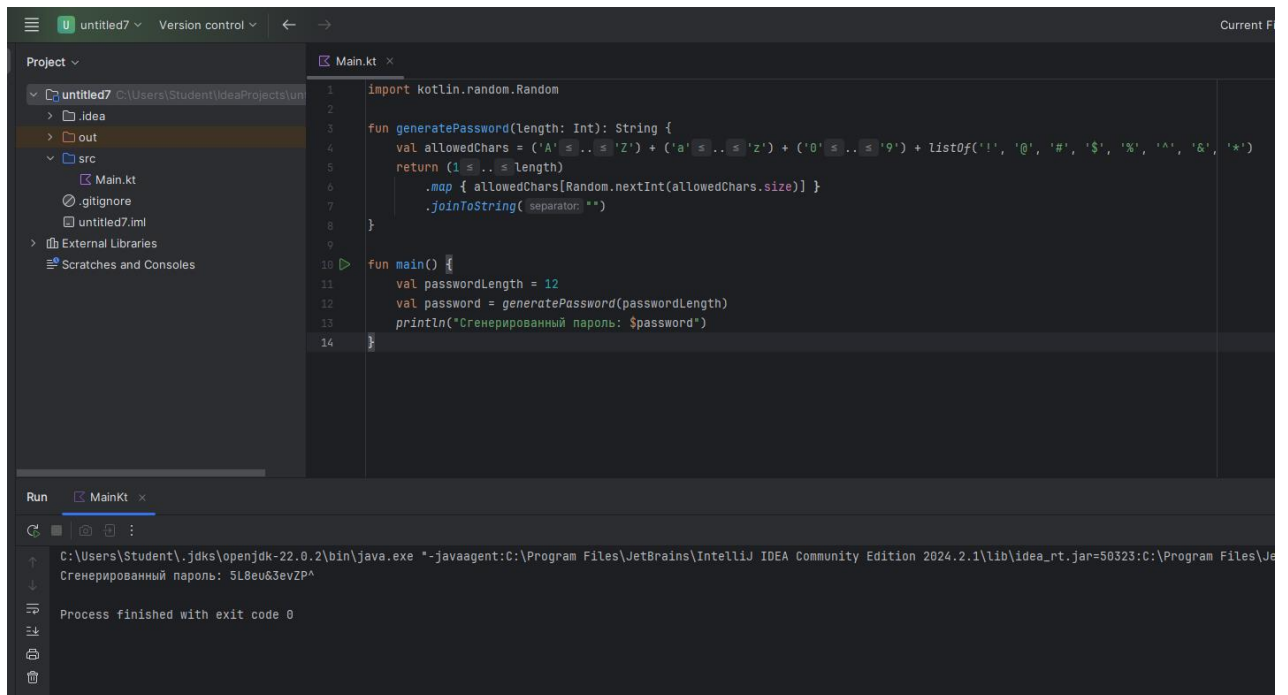
    val passwordLength = 12

    val password = generatePassword(passwordLength)

    println("Сгенерированный пароль: $password")

}

```



```

10. fun findLongestWord(text: String): String {

    val words = text.split("\\s+").toRegex()

    var longestWord = ""

    for (word in words) {

        val cleanWord = word.filter { it.isLetterOrDigit() }

        if (cleanWord.length > longestWord.length) {

            longestWord = cleanWord

        }

    }

    return longestWord
}

```

```
}
```

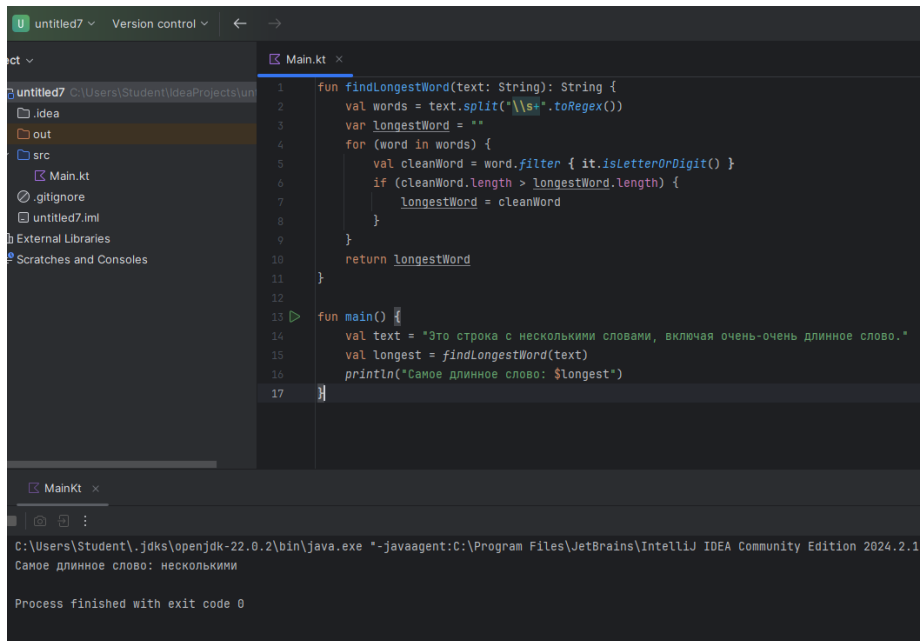
```
fun main() {
```

```
    val text = "Это строка с несколькими словами, включая очень-очень длинное слово."
```

```
    val longest = findLongestWord(text)
```

```
    println("Самое длинное слово: $longest")
```

```
}
```



The screenshot shows the IntelliJ IDEA IDE with a Kotlin file named Main.kt. The code defines a function `findLongestWord` that takes a string and returns the longest word. It uses `split("\\s+\\.toRegex())` to split the string into words, then iterates through them, filtering out non-alphanumeric characters and comparing lengths. The `main` function tests this with the string "Это строка с несколькими словами, включая очень-очень длинное слово." and prints the result.

```
1 fun findLongestWord(text: String): String {
2     val words = text.split("\\s+\\.toRegex())
3     var longestWord = ""
4     for (word in words) {
5         val cleanWord = word.filter { it.isLetterOrDigit() }
6         if (cleanWord.length > longestWord.length) {
7             longestWord = cleanWord
8         }
9     }
10    return longestWord
11 }
12
13 fun main() {
14     val text = "Это строка с несколькими словами, включая очень-очень длинное слово."
15     val longest = findLongestWord(text)
16     println("Самое длинное слово: $longest")
17 }
```

The console output at the bottom shows the command executed and the result: "Самое длинное слово: несколькими".

```
C:\Users\Student\.jdk\openjdk-22.0.2\bin\java.exe "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2024.2.1
Самое длинное слово: несколькими

Process finished with exit code 0
```