

## Практическая работа №9

1.

```
import kotlin.random.Random
```

```
fun main() {
```

```
    val random = Random
```

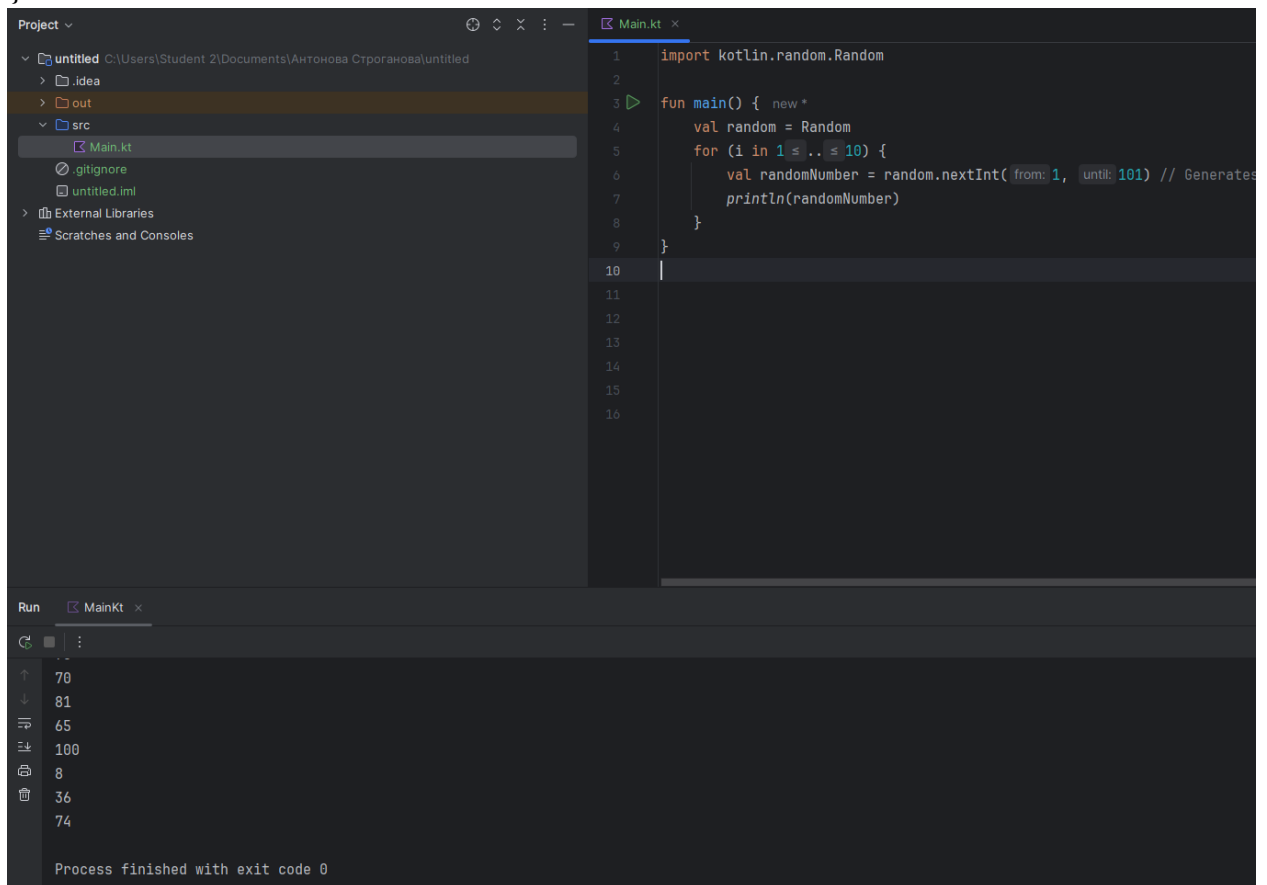
```
    for (i in 1..10) {
```

```
        val randomNumber = random.nextInt(1, 101) // Generates a random number between 1  
(inclusive) and 101 (exclusive), so 1-100
```

```
        println(randomNumber)
```

```
    }
```

```
}
```



2.

```
fun main() {
```

```
    println("Введите строку:")
```

```
    val inputString = readlnOrNull() ?: "" // Read input from the user, or default to an empty string
```

```

val vowels = "aeiouAEIOUаеёиоуыэюяАЕЁИОУЫЭЮЯ" // List of vowels (English and Russian)

var vowelCount = 0

var consonantCount = 0

for (char in inputString) {
    if (char.isLetter()) { // Check if the character is a letter
        if (vowels.contains(char)) {
            vowelCount++
        } else {
            consonantCount++
        }
    }
}

println("Количество гласных: $vowelCount")
println("Количество согласных: $consonantCount")
}

```

The screenshot shows an IDE with a project named 'untitled'. The source file 'Main.kt' contains the following Kotlin code:

```

1 fun main() { new *
2     println("Введите строку:")
3     val inputString = readlnOrNull() ?: "" // Read input from the user, or default to an empty string
4
5     val vowels = "aeiouAEIOUаеёиоуыэюяАЕЁИОУЫЭЮЯ" // List of vowels (English and Russian)
6     var vowelCount = 0
7     var consonantCount = 0
8
9     for (char in inputString) {
10         if (char.isLetter()) { // Check if the character is a letter
11             if (vowels.contains(char)) {
12                 vowelCount++
13             } else {
14                 consonantCount++
15             }
16         }
17     }
18
19     println("Количество гласных: $vowelCount")
20     println("Количество согласных: $consonantCount")
21 }
22
23

```

The Run console shows the execution of the program:

```

Run MainKt x
C:\Program Files\Eclipse Adoptium\jdk-17.0.11.9-hotspot\bin\java.exe" "-javaagent:C:\Users\Student 2\AppData\Local\JetBrains\IntelliJ IDEA Community Edition 2024.2.1
Введите строку:
8
Количество гласных: 0
Количество согласных: 0
Process finished with exit code 0

```

3.

```
fun main() {
```

```
val usdToEurRate = 0.92 // Example exchange rate (as of Oct 26, 2023). This WILL change.  
val eurToUsdRate = 1.09 //Reciprocal to above.
```

```
println("Выберите конвертацию:")
```

```
println("1. USD в EUR")
```

```
println("2. EUR в USD")
```

```
val choice = readlnOrNull()?.toIntOrNull()
```

```
when (choice) {
```

```
    1 -> {
```

```
        println("Введите сумму в USD:")
```

```
        val usdAmount = readlnOrNull()?.toDoubleOrNull()
```

```
        if (usdAmount != null) {
```

```
            val eurAmount = usdAmount * usdToEurRate
```

```
            println("$usdAmount USD = $eurAmount EUR")
```

```
        } else {
```

```
            println("Некорректный ввод суммы в USD.")
```

```
        }
```

```
    }
```

```
    2 -> {
```

```
        println("Введите сумму в EUR:")
```

```
        val eurAmount = readlnOrNull()?.toDoubleOrNull()
```

```
        if (eurAmount != null) {
```

```
            val usdAmount = eurAmount * eurToUsdRate
```

```
            println("$eurAmount EUR = $usdAmount USD")
```

```
        } else {
```

```
            println("Некорректный ввод суммы в EUR.")
```

```
        }
```

```
    }
```

```

    else -> {

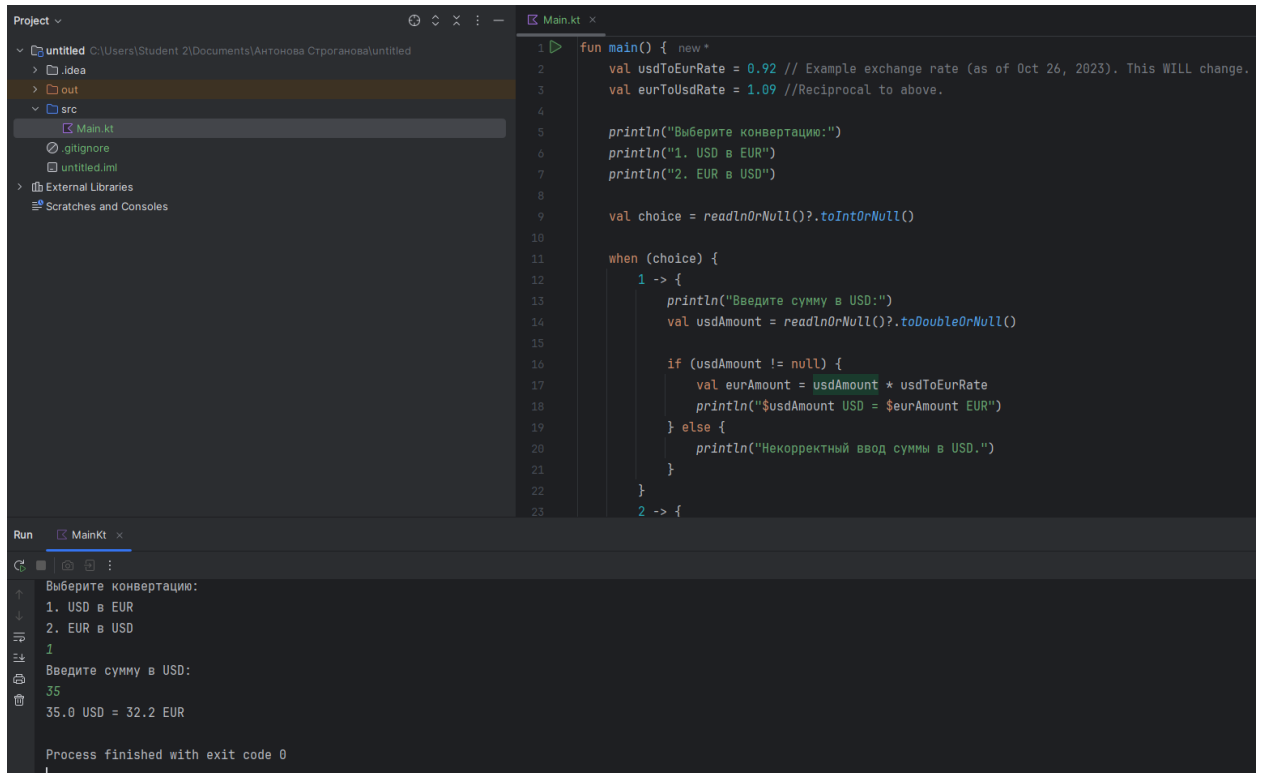
        println("Некорректный выбор операции.")

    }

}

}

```



4.

```
fun areAnagrams(str1: String, str2: String): Boolean {
```

```
    // Remove spaces and convert to lowercase for case-insensitive comparison
```

```
    val cleanStr1 = str1.replace("\\s".toRegex(), "").lowercase()
```

```
    val cleanStr2 = str2.replace("\\s".toRegex(), "").lowercase()
```

```
    // Check if the lengths are different
```

```
    if (cleanStr1.length != cleanStr2.length) {
```

```
        return false
```

```
    }
```

```
    // Convert the strings to character arrays and sort them
```

```
    val charArray1 = cleanStr1.toCharArray().apply { sort() }
```

```
    val charArray2 = cleanStr2.toCharArray().apply { sort() }
```

```

// Compare the sorted character arrays
return charArray1.contentEquals(charArray2)
}

fun main() {

    println("Являются ли \"listen\" и \"silent\" анаграммами: ${areAnagrams(\"listen\", \"silent\")}") // Output: true

    println("Являются ли \"triangle\" и \"integral\" анаграммами: ${areAnagrams(\"triangle\", \"integral\")}") // Output: true

    println("Являются ли \"hello\" и \"world\" анаграммами: ${areAnagrams(\"hello\", \"world\")}") // Output: false

    println("Являются ли \"Debit Card\" и \"Bad Credit\" анаграммами: ${areAnagrams(\"Debit Card\", \"Bad Credit\")}") // Output: true
}

```

The screenshot shows an IDE with a project named 'untitled'. The source file 'Main.kt' contains the following code:

```

1 fun areAnagrams(str1: String, str2: String): Boolean {
2     //
3     // Check if the lengths are different
4     if (cleanStr1.length != cleanStr2.length) {
5         return false
6     }
7
8     // Convert the strings to character arrays and sort them
9     val charArray1 = cleanStr1.toCharArray().apply { sort() }
10    val charArray2 = cleanStr2.toCharArray().apply { sort() }
11
12    // Compare the sorted character arrays
13    return charArray1.contentEquals(charArray2)
14 }
15
16 fun main() {
17     println("Являются ли \"listen\" и \"silent\" анаграммами: ${areAnagrams(str1: \"listen\", str2: \"silent\")}") // Output: true
18     println("Являются ли \"triangle\" и \"integral\" анаграммами: ${areAnagrams(str1: \"triangle\", str2: \"integral\")}") // Output: true
19     println("Являются ли \"hello\" и \"world\" анаграммами: ${areAnagrams(str1: \"hello\", str2: \"world\")}") // Output: false
20     println("Являются ли \"Debit Card\" и \"Bad Credit\" анаграммами: ${areAnagrams(str1: \"Debit Card\", str2: \"Bad Credit\")}") // Output: true
21 }

```

The Run console shows the following output:

```

Являются ли "listen" и "silent" анаграммами: true
Являются ли "triangle" и "integral" анаграммами: true
Являются ли "hello" и "world" анаграммами: false
Являются ли "Debit Card" и "Bad Credit" анаграммами: true
Process finished with exit code 0

```

5.

```

fun isPrime(num: Int): Boolean {

    if (num <= 1) return false

    for (i in 2..Math.sqrt(num.toDouble()).toInt()) {

        if (num % i == 0) return false

    }

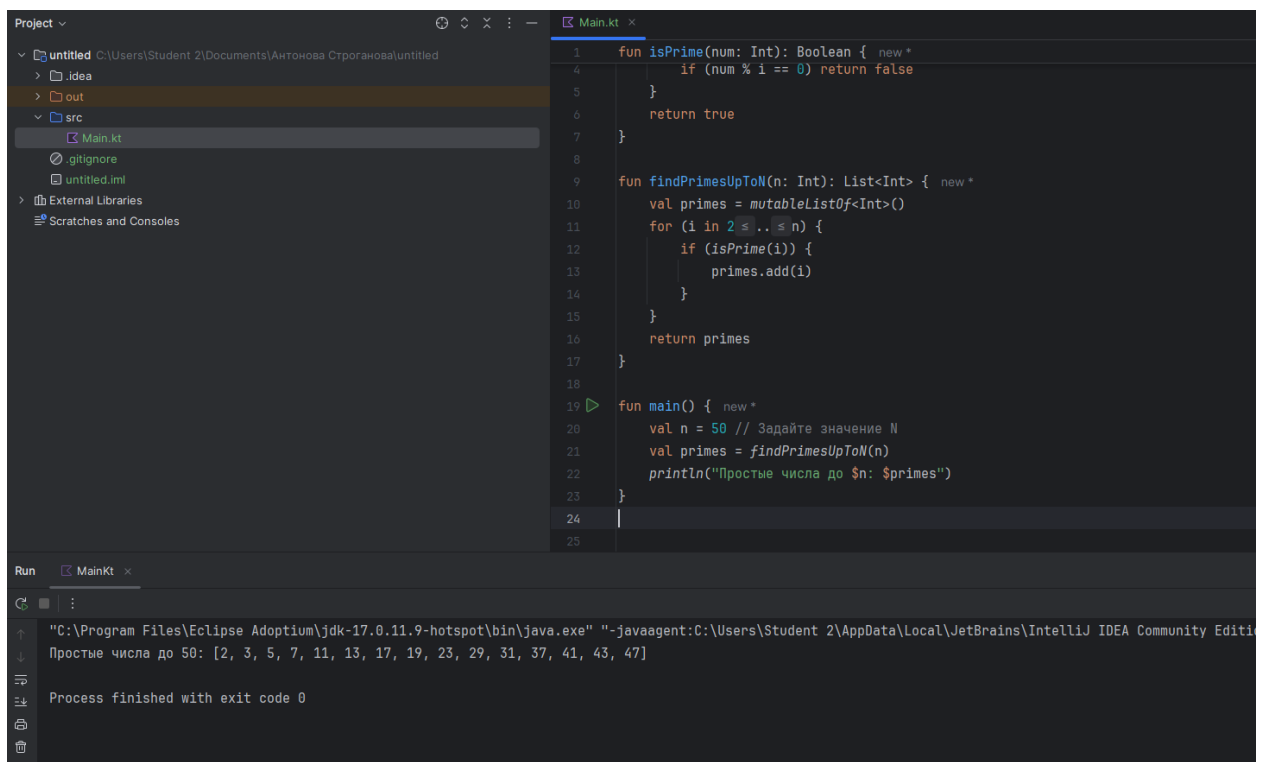
    return true
}

```

```
}
```

```
fun findPrimesUpToN(n: Int): List<Int> {  
    val primes = mutableListOf<Int>()  
    for (i in 2..n) {  
        if (isPrime(i)) {  
            primes.add(i)  
        }  
    }  
    return primes  
}
```

```
fun main() {  
    val n = 50 // Задайте значение N  
    val primes = findPrimesUpToN(n)  
    println("Простые числа до $n: $primes")  
}
```



6.

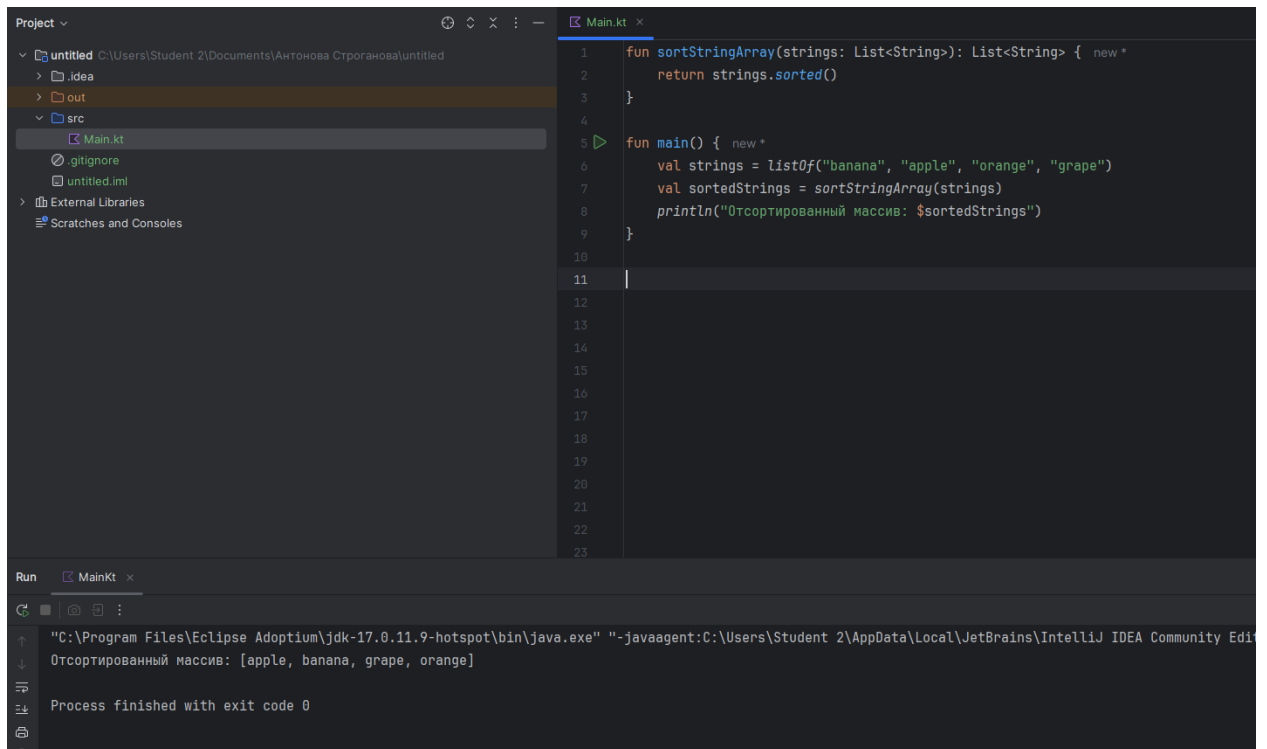
```
fun sortStringArray(strings: List<String>): List<String> {
```

```

        return strings.sorted()
    }

fun main() {
    val strings = listOf("banana", "apple", "orange", "grape")
    val sortedStrings = sortStringArray(strings)
    println("Отсортированный массив: $sortedStrings")
}

```



7.

```

fun toggleCase(input: String): String {
    return input.map {
        if (it.isUpperCase()) it.toLowerCase()
        else it.toUpperCase()
    }.joinToString("")
}

fun main() {
    val inputString = "Hello World" // Задайте строку
    val toggledString = toggleCase(inputString)
}

```

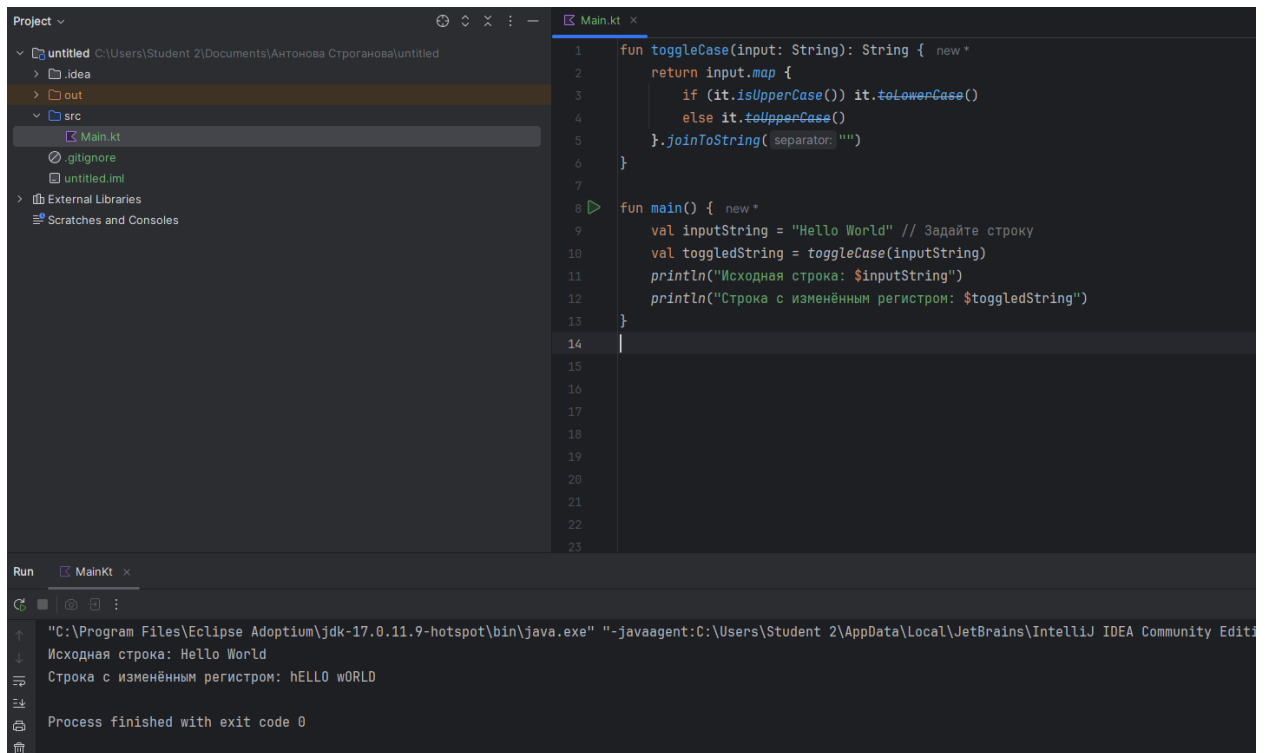
```

println("Исходная строка: $inputString")

println("Строка с изменённым регистром: $toggledString")

}

```



8.

```
import kotlin.random.Random
```

```

fun main() {

    val secretNumber = Random.nextInt(1, 101)

    var guessed = false

    println("Добро пожаловать в игру 'Угадай число!'")

    println("Я загадал число от 1 до 100. Попробуйте угадать его!")

    while (!guessed) {

        println("Введите ваше число:")

        val userInput = readLine()

        if (userInput != null && userInput.toIntOrNull() != null) {

            val guessedNumber = userInput.toInt()

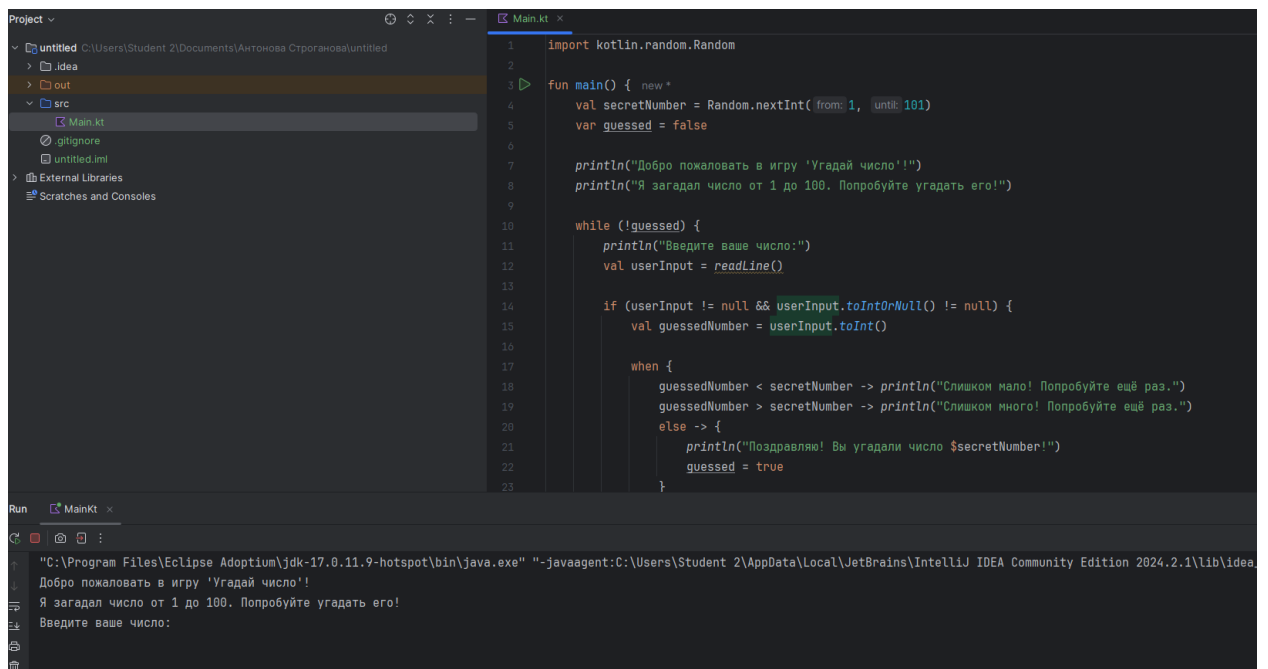
```



```

when {
    guessedNumber < secretNumber -> println("Слишком мало! Попробуйте ещё раз.")
    guessedNumber > secretNumber -> println("Слишком много! Попробуйте ещё
раз.")
    else -> {
        println("Поздравляю! Вы угадали число $secretNumber!")
        guessed = true
    }
}
} else {
    println("Пожалуйста, введите корректное число.")
}
}
}

```



9.

```
import kotlin.random.Random
```

```

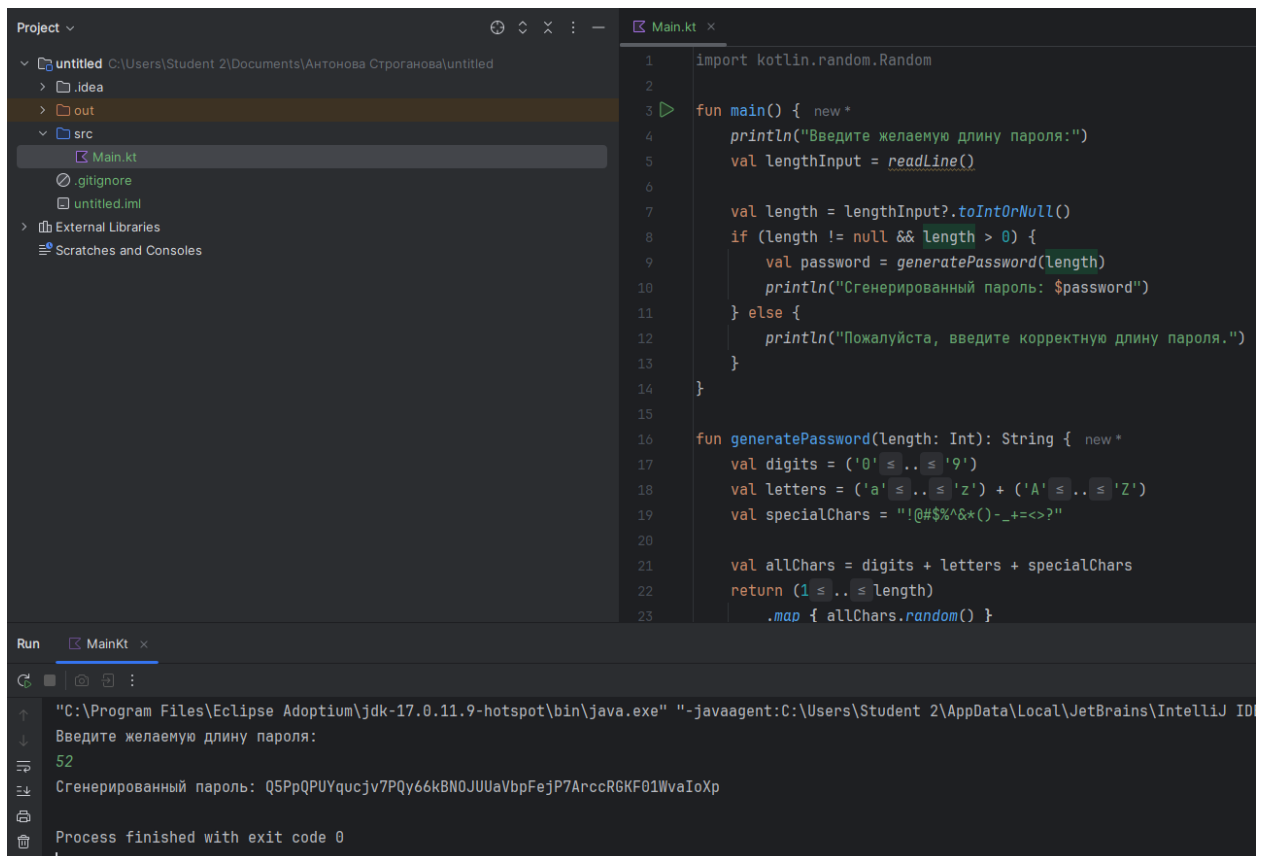
fun main() {
    println("Введите желаемую длину пароля:")
    val lengthInput = readLine()

```

```
val length = lengthInput?.toIntOrNull()
if (length != null && length > 0) {
    val password = generatePassword(length)
    println("Сгенерированный пароль: $password")
} else {
    println("Пожалуйста, введите корректную длину пароля.")
}
}
```

```
fun generatePassword(length: Int): String {
    val digits = ('0'..'9')
    val letters = ('a'..'z') + ('A'..'Z')
    val specialChars = "!@#$$%^&*()-_+=<>?"

    val allChars = digits + letters + specialChars
    return (1..length)
        .map { allChars.random() }
        .joinToString("")
}
```



10.

```

fun main() {

    println("Введите строку:")

    val input = readLine() ?: ""

    if (input.isNotBlank()) {

        val longestWord = findLongestWord(input)

        println("Самое длинное слово: $longestWord")

    } else {

        println("Строка не должна быть пустой.")

    }

}

```

```

fun findLongestWord(input: String): String {

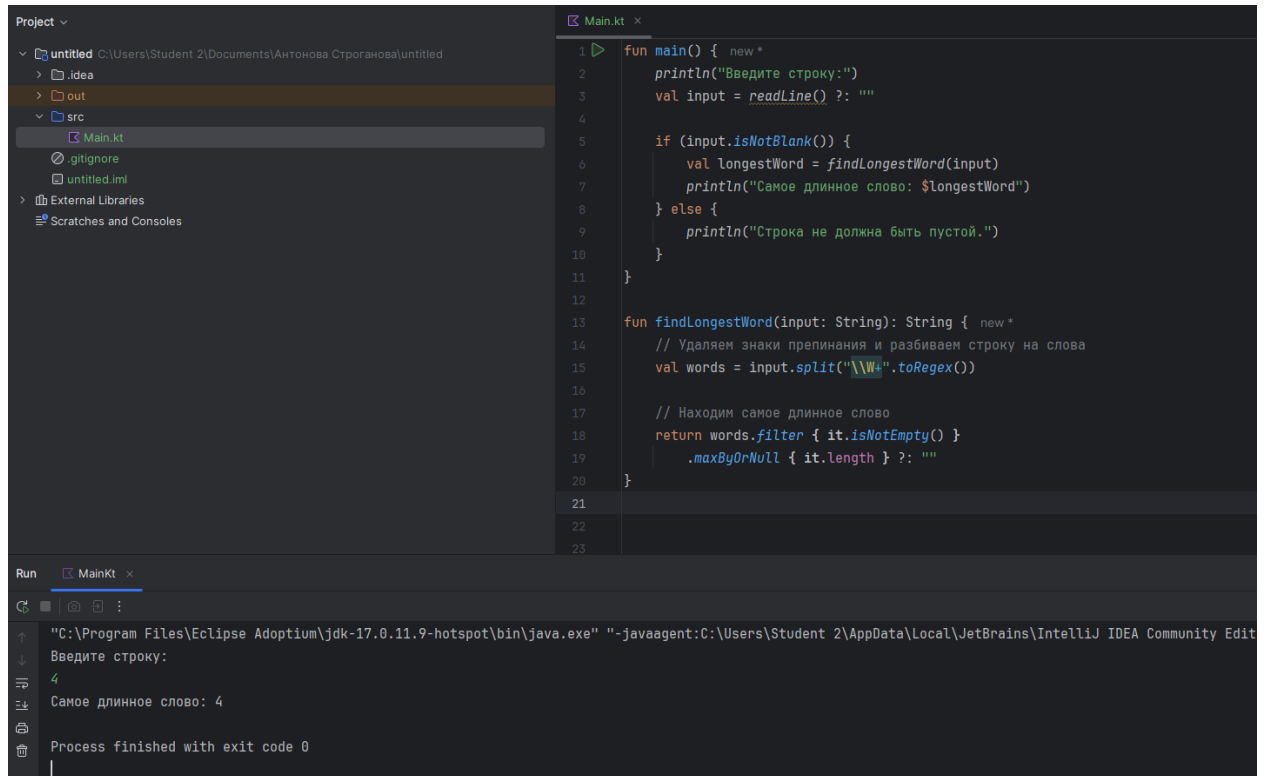
    // Удаляем знаки препинания и разбиваем строку на слова

    val words = input.split("\\W+").toRegex()

```

```
// Находим самое длинное слово
```

```
return words.filter { it.isNotEmpty() }  
    .maxByOrNull { it.length } ?: ""  
}
```



The screenshot displays the IntelliJ IDEA IDE interface. The left sidebar shows the project structure with 'Main.kt' selected. The main editor window shows the following Kotlin code:

```
1 fun main() { new *  
2     println("Введите строку:")  
3     val input = readLine() ?: ""  
4  
5     if (input.isNotBlank()) {  
6         val longestWord = findLongestWord(input)  
7         println("Самое длинное слово: $longestWord")  
8     } else {  
9         println("Строка не должна быть пустой.")  
10    }  
11 }  
12  
13 fun findLongestWord(input: String): String { new *  
14     // Удаляем знаки препинания и разбиваем строку на слова  
15     val words = input.split("\\W+").toRegex()  
16  
17     // Находим самое длинное слово  
18     return words.filter { it.isNotEmpty() }  
19         .maxByOrNull { it.length } ?: ""  
20 }  
21  
22  
23
```

The bottom panel shows the Run output for 'MainKt':

```
"C:\Program Files\Eclipse Adoptium\jdk-17.0.11.9-hotspot\bin\java.exe" "-javaagent:C:\Users\Student 2\AppData\Local\JetBrains\IntelliJ IDEA Community Edit  
Введите строку:  
4  
Самое длинное слово: 4  
Process finished with exit code 0
```