

Мироненко Артём Сергеевич ИС231
Практическая работа 5
Циклы

1. Вывод чисел от 1 до 10: Напишите программу, которая выводит числа от 1 до 10.

```
fun main() {  
    println((1..10).joinToString(", "))  
}
```

```
"C:\Users\User\AppData\Local\Program  
1, 2, 3, 4, 5, 6, 7, 8, 9, 10  
  
Process finished with exit code 0
```

2. Вывод четных чисел от 1 до 20: Напишите программу, которая выводит все четные числа от 1 до 20.

```
fun main() {  
    for (i in 2..20 step 2){  
        print(if (i<20) "$i, " else i)  
    }  
}
```

```
"C:\Users\User\AppData\Local\Programs\  
2, 4, 6, 8, 10, 12, 14, 16, 18, 20  
Process finished with exit code 0
```

3. Сумма чисел от 1 до N: Пользователь вводит число N, программа суммирует все числа от 1 до N и выводит результат.

```
fun main() {  
    print("Введите число N: ")  
    val n = readLine()!!.toInt()  
    print((1..n).joinToString(", "))  
    println("\nСумма: ${ (1..n).sum() }")  
}
```

```
"C:\Users\User\AppData\Local\Programs\  
Введите число N: 6  
1, 2, 3, 4, 5, 6  
Сумма: 21  
  
Process finished with exit code 0
```

4. Факториал числа: Напишите программу, которая вычисляет факториал введенного пользователем числа.

```
fun main() {  
    print("Введите число: ")  
    val n = readLine()!!.toInt()  
    println("Факториал числа $n = ${(1..n).reduce{acc, i -> acc *  
i}}}")  
}
```

```
"C:\Users\User\AppData\Local\Program  
Введите число: 5  
Факториал числа 5 = 120  
  
Process finished with exit code 0
```

5. Проверка числа на простоту: Пользователь вводит число, программа определяет, является ли оно простым.

```
fun main() {  
    print("Введите число: ")  
    val n = readLine()!!.toInt()  
    val isPrime = n > 1 &&  
(2..kotlin.math.sqrt(n.toDouble()).toInt()).all { n % it != 0 }  
    println(if (isPrime) "$n - простое число" else "$n - не простое  
число")  
}
```

```
"C:\Users\User\AppData\Local\Progra  
Введите число: 1  
1 - не простое число  
  
Process finished with exit code 0
```

```
"C:\Users\User\AppData\Local\Progra  
Введите число: 3  
3 - простое число  
  
Process finished with exit code 0
```

6. Вывод таблицы умножения: Напишите программу, которая выводит таблицу умножения от 1 до 10.

```
fun main() {  
    println("Таблица умножения:")  
    for (i in 1..10){  
        for (j in 1..10){  
            print("%4d".format(i*j))  
        }  
        println()  
    }  
}
```

```
"C:\Users\User\AppData\Local\Programs\Eclipse
Таблица умножения:
  1  2  3  4  5  6  7  8  9 10
  2  4  6  8 10 12 14 16 18 20
  3  6  9 12 15 18 21 24 27 30
  4  8 12 16 20 24 28 32 36 40
  5 10 15 20 25 30 35 40 45 50
  6 12 18 24 30 36 42 48 54 60
  7 14 21 28 35 42 49 56 63 70
  8 16 24 32 40 48 56 64 72 80
  9 18 27 36 45 54 63 72 81 90
 10 20 30 40 50 60 70 80 90 100

Process finished with exit code 0
```

7. Фибоначчи: Сгенерируйте первые N чисел Фибоначчи (где N вводит пользователь).

```
fun main() {
    print("Введите количество чисел: ")
    val n = readLine()!!.toInt()
    val nums = (1..n).fold(mutableListOf(0,1))
    {list, _ ->
        list.add(list[list.size - 1] + list[list.size - 2])
        list
    }
    println(nums.take(n).joinToString(", "))
}
```

```
"C:\Users\User\AppData\Local\Programs\Eclipse
Введите количество чисел: 10
0, 1, 1, 2, 3, 5, 8, 13, 21, 34

Process finished with exit code 0
```

8. Наибольший общий делитель (НОД): Напишите программу, которая находит НОД двух введенных чисел с использованием алгоритма Евклида.

```
fun main() {
    print("Введите два числа через пробел: ")
    var (a, b) = readLine()!!.split(" ").map { it.toInt() }
    while (b!=0) {
        a = b.also { b = a % b }
    }
    println("НОД = $a")
}
```

```
"C:\Users\User\AppData\Local\Programs\Ec
Введите два числа через пробел: 140 96
НОД = 4
```

```
Process finished with exit code 0
```

9. Обратный порядок: Пользователь вводит строку, и программа выводит ее в обратном порядке.

```
fun main() {
    print("Введите строку: ")
    println(readLine()!!.reversed())
}
```

```
"C:\Users\User\AppData\Local\Programs\
Введите строку: 123456 вода
адов 654321
```

```
Process finished with exit code 0
```

10. Сумма цифр числа: Напишите программу, которая находит сумму цифр введенного числа.

```
fun main() {
    print("Введите число: ")
    println("Сумма цифр:
${readLine()!!.map{it.toString().toInt()}.sum()}")
}
```

```
"C:\Users\User\AppData\Local\Program
Введите число: 1234
Сумма цифр: 10
```

```
Process finished with exit code 0
```

11. Анаграммы: Программа проверяет, являются ли две введенные строки анаграммами.

```
fun main() {
    print("Введите первую строку: ")
    val str1 = readLine()!!.lowercase().filter { it.isLetter()
}.toCharArray().sorted()
    print("Введите вторую строку: ")
    val str2 = readLine()!!.lowercase().filter { it.isLetter()
}.toCharArray().sorted()
    println(if (str1 == str2) "Строки являются анаграммами" else
"Строки не являются анаграммами")
}
```

```
"C:\Users\User\AppData\Local\Programs\E
Введите первую строку: автор
Введите вторую строку: товар
Строки являются анаграммами

Process finished with exit code 0
```

12. Числовая последовательность: Пользователь вводит начальное число и шаг, программа генерирует числовую последовательность.

```
fun main() {
    print("Введите начальное число: ")
    val start = readln().toInt()
    print("Введите шаг: ")
    val step = readln().toInt()
    print("Введите количество чисел: ")
    println((0 until readln().toInt()).map { start + it * step }
        .joinToString(", "))
}
```

```
"C:\Users\User\AppData\Local\Programs\Ecl
Введите начальное число: 0
Введите шаг: 2
Введите количество чисел: 10
0, 2, 4, 6, 8, 10, 12, 14, 16, 18

Process finished with exit code 0
```

13. Таблица квадратов: Выведите таблицу квадратов чисел от 1 до 20.

```
fun main() {
    println("Число | Квадрат")
    println("=====")
    (1..20).forEach { number ->
        val square = number * number
        println(String.format("%-5d | %d", number, square))
    }
}
```

```

"C:\Users\User\AppData\Local\Programs\Python\Python39\python.exe"
Число | Квадрат
=====
1      | 1
2      | 4
3      | 9
4      | 16
5      | 25
6      | 36
7      | 49
8      | 64
9      | 81
10     | 100
11     | 121
12     | 144
13     | 169
14     | 196
15     | 225
16     | 256
17     | 289
18     | 324
19     | 361
20     | 400

Process finished with exit code 0

```

14.Генерация случайных чисел: Сгенерируйте и выведите 10 случайных чисел от 1 до 100.

```

fun main() {
    println((1..10).map{ (1..100).random() }.joinToString(", "))
}

```

```

"C:\Users\User\AppData\Local\Programs\Python\Python39\python.exe"
95, 24, 73, 68, 53, 25, 10, 46, 95, 85

Process finished with exit code 0

```

15.Проверка палиндрома: Пользователь вводит строку, и программа проверяет, является ли она палиндромом.

```

fun main() {
    print("Введите строку: ")
    val txt = readln().lowercase().filter { it.isLetterOrDigit() }
    println(if (txt == txt.reversed()) "Это палиндром" else "Это не палиндром")
}

```

```
"C:\Users\User\AppData\Local\Program
Введите строку: радар
Это палиндром

Process finished with exit code 0
```

16. Сигма (сумма квадратов): Найдите сумму квадратов всех чисел от 1 до N.

```
fun main() {
    print("Введите N: ")
    println((1..readln().toInt()).sumOf { it * it })
}
```

```
"C:\Users\User\AppData\Local\Programs
Введите N: 3
14

Process finished with exit code 0
```

17. Вывод символов: Напишите программу, которая выводит символы строки по одному, используя циклы.

```
fun main() {
    print("Введите текст: ")
    val text = readLine() ?: ""
    var i = 0
    while (i < text.length) {
        println(text[i])
        i++
    }
}
```

```
"C:\Users\User\AppData\Local\Programs
Введите текст: строка
с
т
р
о
к
а

Process finished with exit code 0
```

18. Задача на лестницу: Напишите программу, которая выводит лестницу из символа "#" высотой N, где N задает пользователь.

```
fun main() {
    print("Введите высоту лестницы: ")
    (1..readln().toInt()).forEach{ println("#".repeat(it))}
}
```

```
"C:\Users\User\AppData\Local\Programs\
Введите высоту лестницы: 4
#
##
###
####

Process finished with exit code 0
```

19.Сортировка списка: Используя цикл, напишите простую сортировку для двухзначных чисел в массиве.

```
fun main() {
    val n = arrayOf(45, 32, 88, 12, 67, 94, 23, 11)
    println(n.sorted().joinToString(", "))
}
```

```
"C:\Users\User\AppData\Local\Programs
11, 12, 23, 32, 45, 67, 88, 94

Process finished with exit code 0
```

20.Простые числа в диапазоне: Выведите все простые числа в заданном пользователем диапазоне.

```
fun main() {
    println("Введите начало диапазона:")
    val start = readLine()!!.toInt()
    println("Введите конец диапазона:")
    val end = readLine()!!.toInt()

    (start..end)
        .filter { n -> (2..kotlin.math.sqrt(n.toDouble()).toInt())
            .none { n % it == 0 } && n > 1 }
        .forEach { print("$it ") }
}
```



```
"C:\Users\User\AppData\Local\Programs\Microsoft VS Code\bin\code.exe"
Введите начало диапазона:
0
Введите конец диапазона:
15
2 3 5 7 11 13
Process finished with exit code 0
```

21. Вывод даты: Пользователь вводит год и месяц, программа выводит все даты в этом месяце.

```
import java.time.YearMonth

fun main() {
    println("Введите год:")
    val year = readLine()!!.toInt()
    println("Введите месяц (1-12):")
    val month = readLine()!!.toInt()

    YearMonth.of(year, month).let { ym ->
        (1..ym.lengthOfMonth()).forEach { day ->
            println("%02d.%02d.%d".format(day, month, year))
        }
    }
}
```

```
"C:\Users\User\AppData\Local\Programs\Microsoft VS Code\bin\code.exe"
Введите год:
2025
Введите месяц (1-12):
2
01.02.2025
02.02.2025
03.02.2025
04.02.2025
05.02.2025
06.02.2025
07.02.2025
08.02.2025
09.02.2025
10.02.2025
11.02.2025
12.02.2025
13.02.2025
14.02.2025
15.02.2025
16.02.2025
17.02.2025
18.02.2025
19.02.2025
20.02.2025
21.02.2025
22.02.2025
```

```
22.02.2025
23.02.2025
24.02.2025
25.02.2025
26.02.2025
27.02.2025
28.02.2025
```

```
Process finished with exit code 0
```

22. Угадай число: Напишите игру, в которой пользователь должен угадать случайное число от 1 до 100.

```
fun main() {  
    val number = (1..100).random()  
    while (true) {  
        val guess = readLine()!!.toInt()  
        when {  
            guess > number -> println("Меньше")  
            guess < number -> println("Больше")  
            else -> break  
        }  
    }  
    println("Угадали!")  
}
```

```
"C:\Users\User\AppData\Local\Programs\Ecl  
5  
Больше  
10  
Больше  
45  
Больше  
67  
Меньше  
50  
Меньше  
47  
Меньше  
46  
Угадали!  
  
Process finished with exit code 0
```

23. Сложение и умножение: Напишите программу, которая запрашивает у пользователя две цифры и повторяет сложение или умножение, до тех пор, пока пользователь не введет "стоп".

```
fun main() {  
    println("Введите выражение в формате: число операция число")  
    println("Пример: 3+3 или 5 * 2")  
    println("Для выхода введите 'стоп'")  
  
    while (true) {  
        print("Введите выражение: ")  
        val input = readLine()?.trim() ?: break  
  
        if (input.lowercase() == "стоп") {
```

```

        println("Программа завершена")
        break
    }

    try {
        val parts = input.replace(" ", "").replace("+", " +")
            .replace("*", " * ").split(" ")
        val (a, op, b) = parts
        val result = when(op) {
            "+" -> a.toInt() + b.toInt()
            "*" -> a.toInt() * b.toInt()
            else -> throw Exception("Неподдерживаемая операция")
        }
        println("Результат: $result")
    } catch (e: Exception) {
        println("Ошибка! Проверьте формат ввода")
    }
}
}

```

```

"C:\Users\User\AppData\Local\Programs\Es
Введите выражение в формате: число опера
Пример: 3+3 или 5 * 2
Для выхода введите 'стоп'
Введите выражение: 3+3
Результат: 6
Введите выражение: 5+2
Результат: 7
Введите выражение: 6*2
Результат: 12
Введите выражение: стоп
Программа завершена

Process finished with exit code 0

```

24. Транспонирование матрицы: Напишите программу, которая транспонирует матрицу (двумерный массив).

```

fun main() {
    println("Введите количество строк:")
    val rows = readln().toInt()

    println("Введите количество столбцов:")
    val cols = readln().toInt()

    val matrix = Array(rows) { Array(cols) { 0 } }

    println("Введите элементы матрицы по одному:")
    for (i in 0 until rows) {

```

```

        for (j in 0 until cols) {
            print("Введите элемент [${i+1}][${j+1}]: ")
            matrix[i][j] = readln().toInt()
        }
    }

    println("\nИсходная матрица:")
    for (i in 0 until rows) {
        for (j in 0 until cols) {
            print("${matrix[i][j]} ")
        }
        println()
    }

    println("\nТранспонированная матрица:")
    for (i in 0 until cols) {
        for (j in 0 until rows) {
            print("${matrix[j][i]} ")
        }
        println()
    }
}

```

```

"C:\Users\User\AppData\Local\Programs\
Введите количество строк:
2
Введите количество столбцов:
3
Введите элементы матрицы по одному:
Введите элемент [1][1]: 1
Введите элемент [1][2]: 2
Введите элемент [1][3]: 3
Введите элемент [2][1]: 4
Введите элемент [2][2]: 5
Введите элемент [2][3]: 6

Исходная матрица:
1 2 3
4 5 6

Транспонированная матрица:
1 4
2 5
3 6

Process finished with exit code 0

```

25. Кубы чисел: Выведите кубы чисел от 1 до 10.

```
fun main() = (1..10).forEach { println("$it в кубе = ${it * it * it}") }
```

```
"C:\Users\User\AppData\Local\Programs\Microsoft VS Code\bin\code.exe"
1 в кубе = 1
2 в кубе = 8
3 в кубе = 27
4 в кубе = 64
5 в кубе = 125
6 в кубе = 216
7 в кубе = 343
8 в кубе = 512
9 в кубе = 729
10 в кубе = 1000

Process finished with exit code 0
```

26. Сумма четных и нечетных чисел: Пользователь вводит N, программа считает сумму четных и нечетных чисел от 1 до N.

```
fun main() {
    println("Введите N:")
    val n = readLine()!!.toInt()
    val (even, odd) = (1..n).partition { it % 2 == 0 }
    println("Сумма четных: ${even.sum()}")
    println("Сумма нечетных: ${odd.sum()}")
}
```

```
"C:\Users\User\AppData\Local\Programs\Microsoft VS Code\bin\code.exe"
Введите N:
5
Сумма четных: 6
Сумма нечетных: 9

Process finished with exit code 0
```

27. Печать числа "пирамида": Напишите программу, которая выводит "пирамиду" из чисел от 1 до N.

```
fun main() {
    println("Введите N:")
    val n = readLine()!!.toInt()
    for (i in 1..n) {
        println(" ".repeat(n - i) + (1..i).joinToString(""))
    }
}
```

```

"C:\Users\User\AppData\Local\Programs
Введите N:
6
1
12
123
1234
12345
123456

Process finished with exit code 0

```

28.Определение порядка: Программа принимает N чисел и выводит их в порядке возрастания.

```

fun main() {
    println("Введите числа через пробел:")
    val numbers = readLine()!!.split(" ").map { it.toInt() }
}.sorted()
    println("Отсортированные числа:")
    println(numbers.joinToString(" "))
}

```

```

"C:\Users\User\AppData\Local\Programs
Введите числа через пробел:
30 10 60 20 80 40 90 50 70
Отсортированные числа:
10 20 30 40 50 60 70 80 90

Process finished with exit code 0

```

29.Сумма ряда: Напишите программу, которая находит сумму ряда $1, 1/2, 1/3, \dots, 1/N$.

```

fun main() {
    println("Введите N:")
    val n = readLine()!!.toInt()
    val sum = (1..n).sumOf { 1.0/it }
    println("Сумма: %.4f".format(sum))
}

```

```

"C:\Users\User\AppData\Local\Programs
Введите N:
4
Сумма: 2,0833

Process finished with exit code 0

```

30. Конвертация в двоичную систему: Напишите программу, которая конвертирует целое число в двоичную систему.

```
fun main() = println("Введите число:").let {  
println(readLine()!!.toInt().toString(2)) }
```

```
"C:\Users\User\AppData\Local\Program  
Введите число:  
27  
11011  
  
Process finished with exit code 0
```