

# Алгоритмизация и программирование

Программирование на Kotlin  
(ч.8 – двумерные массивы)



Беркунский Е.Ю., кафедра ИУСТ, НУК  
eugeny.berkunsky@gmail.com  
<http://www.berkut.mk.ua>

# Двумерные массивы

Индексы строк --- ↓

|   | 0  | 1 | 2  | 3  | ←--- Индексы столбцов            |
|---|----|---|----|----|----------------------------------|
| 0 | 2  | 7 | 6  | 11 |                                  |
| 1 | -7 | 8 | 1  | 5  | <code>intMatrix[1][2] //1</code> |
| 2 | 1  | 4 | 18 | 3  | <code>intMatrix[2][3] //3</code> |

Всего в массиве **3 строки** и **4 столбца**, то есть  $3 \times 4 = 12$  элементов

```
val intMatrix = Array(3) { IntArray(4) }
```

# Матриці

**Матриця** – это прямоугольная таблица однотипных элементов.

**Матриця** – это массив, в котором каждый элемент имеет два индекса (номер строки и номер столбца).

A

|   | 0 | 1  | 2  | 3  | 4  |
|---|---|----|----|----|----|
| 0 | 1 | 4  | 6  | 3  | 7  |
| 1 | 2 | -5 | 0  | 15 | 10 |
| 2 | 8 | 12 | 11 | 12 | 11 |

столбец 4

строка 1

ячейка  $a[2][1]$

# Объявление двумерного массива

```
val intMatrix = Array(3) { IntArray(4) } // заполняем 0
```

- Мы объявили двумерный массив, состоящий из 3 строк, по 4 элемента в каждой строке
- Иногда говорят, что такой двумерный массив состоит из 3 строк и 4 столбцов

# Инициализация двумерного массива

- Для одномерного массива мы могли написать так:

```
val arr1 = IntArray(10) { 1 }  
val arr2 = IntArray(10) { it+1 }  
val arr3 = IntArray(10) { readLine()!!.toInt() }
```

А как быть с двумерным массивом? В чем сложность?

# Инициализация двумерного массива

- Для одномерного массива мы могли написать так:

```
val arr1 = IntArray(10) { 1 }  
val arr2 = IntArray(10) { it+1 }  
val arr3 = IntArray(10) { readLine()!!.toInt() }
```

- Для двумерного массива можно написать так:

```
val a = Array(3) { IntArray(4) { 1 } } //массив заполнен 1
```

# Инициализация двумерного массива

- Заполним массив «таблицей умножения»

```
val a = Array(10) {  
    row -> IntArray(10) {  
        column -> (row + 1) * (column + 1)  
    }  
}
```

Примечание: можно это было записать и в одну строку.


Пояснение: Создается двумерный массив из 10 строк. Каждая строка представляет собой одномерный массив из 10 целых чисел, в зависимости от места в массиве, значения элементов определяются по правилу:

$$(row + 1) * (column + 1)$$

## 1. Классический способ

```
val (rows, columns) = readInts()
val intMatrix = Array(rows) { IntArray(columns) }
for (i in intMatrix.indices) {
    for (j in intMatrix[i].indices) {
        intMatrix[i][j] = readInt()
    }
}
```

Сначала считываем размеры, потом все элементы массива по одному в строке





## 2. С помощью java.util.Scanner

```
val input = Scanner(System.`in`)
val rows = input.nextInt()
val columns = input.nextInt()
val intMatrix = Array(rows) { IntArray(columns) }
for (i in intMatrix.indices) {
    for (j in intMatrix[i].indices) {
        intMatrix[i][j] = input.nextInt()
    }
}
```

Сначала считываем размеры, потом все элементы массива по одному в строке



## 3. Построчно

```
val intMatrix = Array(3) {  
    readLine()!!.split(" ").map{ it.toInt() }.toIntArray()  
}
```

Сначала считываем размеры, потом все элементы массива построчно



## 4. С помощью `java.util.Scanner` (все и сразу)

```
val input = Scanner(System.`in`)  
val rows = input.nextInt()  
val columns = input.nextInt()  
val intMatrix = Array(rows) {  
    IntArray(columns) { input.nextInt() }  
}
```



## 5. Построчно (разное количество элементов)

```
val intMatrix = Array(3) {  
    readLine()!!.split(" ").map{ it.toInt() }.toIntArray()  
}
```

Что будет, если ввести разное количество элементов в строках?



## 1. Классический способ

```
for (i in intMatrix.indices) {  
    for (j in intMatrix[i].indices) {  
        print("%3d".format(intMatrix[i][j]))  
    }  
    println()  
}
```



# Выводим на экран

## 1. Классический способ

```
for (i in intMatrix.indices) {  
    for (j in intMatrix[i].indices) {  
        print("%3d".format(intMatrix[i][j]))  
    }  
    println()  
}
```

## 2. Улучшенный способ

```
for (row in intMatrix) {  
    for (x in row) {  
        print("%3d".format(x))  
    }  
    println()  
}
```

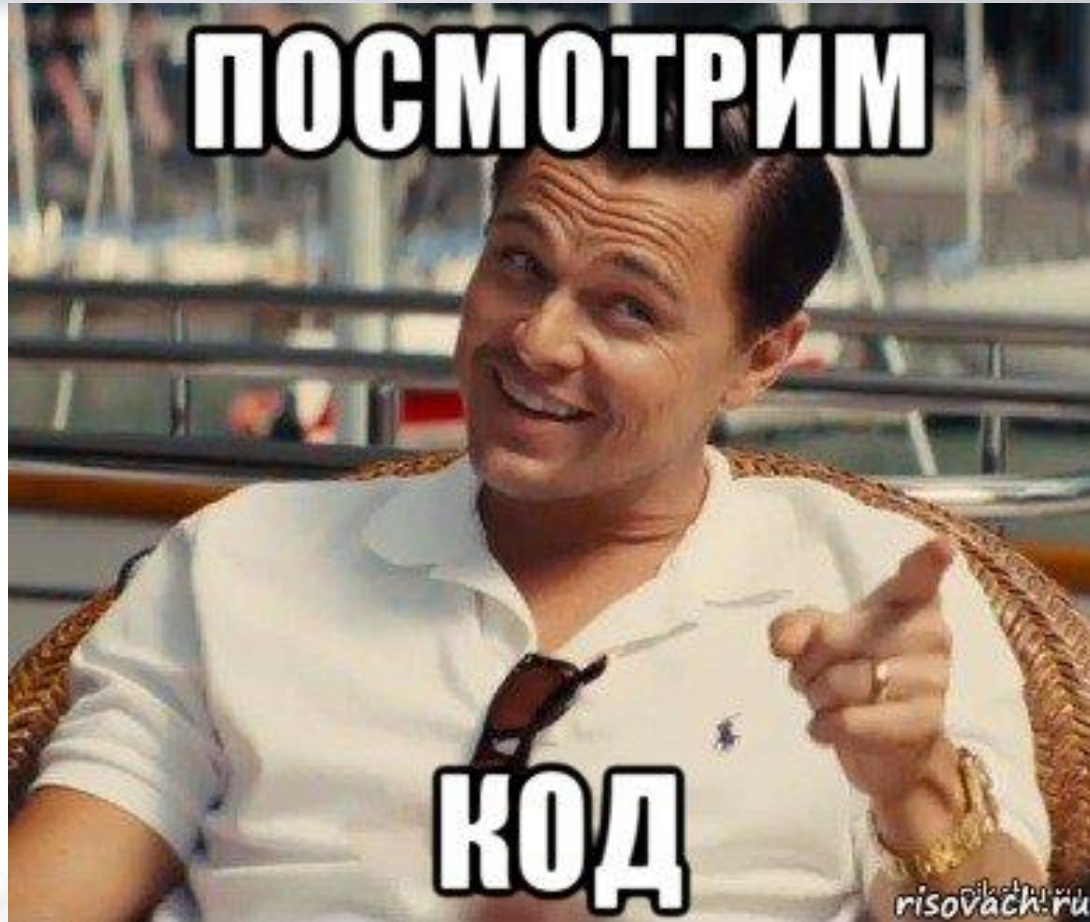
## 3. «Модный» способ

```
intMatrix.forEach {  
    it.forEach { print("%-3d".format(it)) }  
    println()  
}
```





# Демо





# Задачи с двумерными массивами

- Сумма элементов

```
fun sum(a: Array<IntArray>): Int {  
    var result = 0  
    for (row in a) {  
        for (x in row) {  
            result += x  
        }  
    }  
    return result  
}
```

# Задачі с двумерними массивами

- Наибольший элемент

```
fun max(a: Array<IntArray>): Int {  
    var result = a[0][0]  
    for (row in a) {  
        for (x in row) {  
            result = max(result, x)  
        }  
    }  
    return result  
}
```

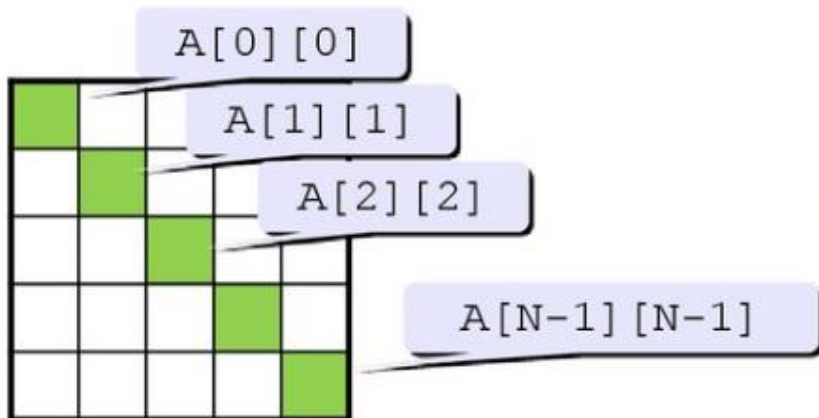
# Задачи с двумерными массивами

- Индексы наибольшего элемента

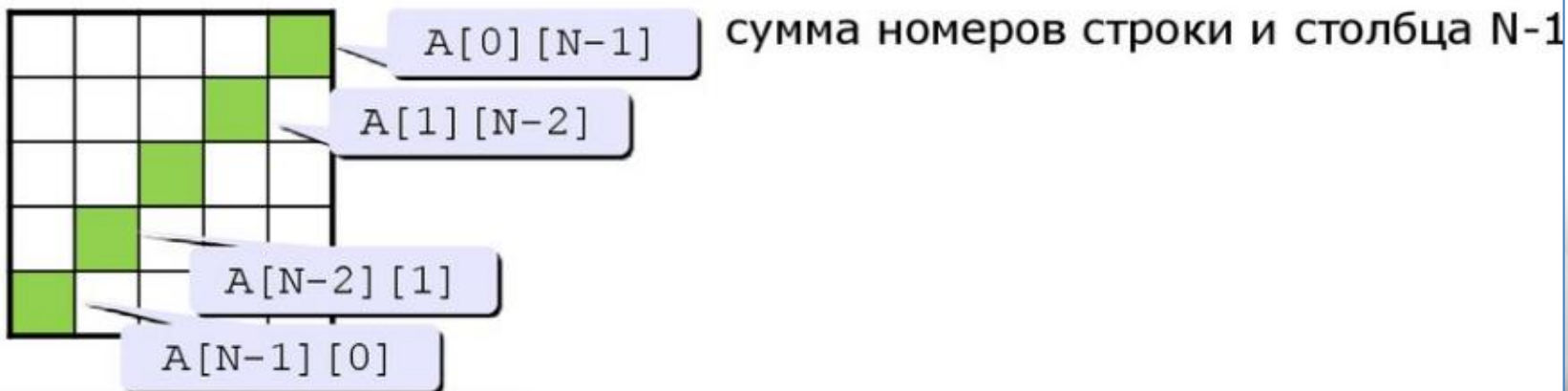
```
fun indicesOfMax(a: Array<IntArray>): Pair<Int, Int> {  
    var result = Pair(0,0)  
    var max = a[0][0]  
    for (i in a.indices) {  
        for (j in a[i].indices) {  
            if (a[i][j] > max) {  
                max = a[i][j]  
                result = Pair(i,j)  
            }  
        }  
    }  
    return result  
}
```

## без вложенных циклов

**Задача 1.** Вывести на экран главную диагональ квадратной матрицы из  $N$  строк и  $N$  столбцов.

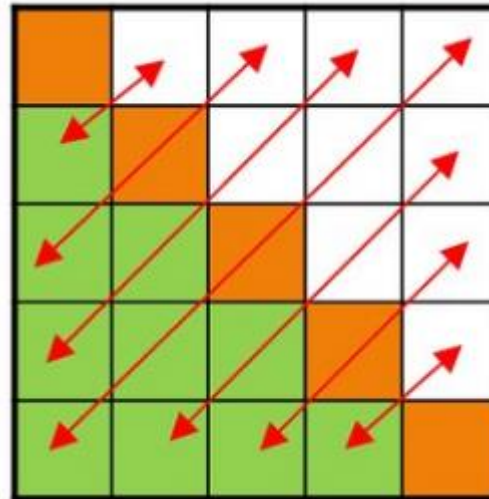


**Задача 2.** Вывести на экран вторую диагональ.



# Транспонирование

- Транспонирование – отображение элементов относительно главной диагонали



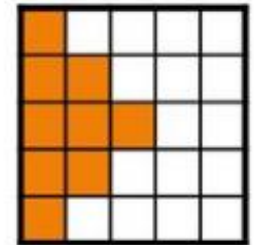
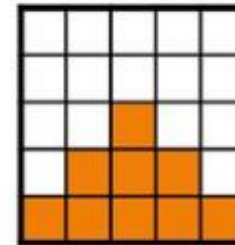
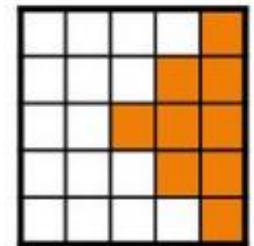
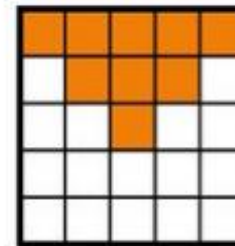
# Матричные операции

- Сложить две матрицы
- Умножить матрицу (все ее элементы) на одно число
- Перемножить две матрицы



# Еще задачи

Заполнить многомерный массив следующим образом: обнулить элементы, отмеченные оранжевым цветом фоном, остальные ячейки заполнить единицами, вывести полученный массив на экран.





НАЦІОНАЛЬНИЙ  
УНІВЕРСИТЕТ  
КОРАБЛЕБУДУВАННЯ  
ІМЕНІ АДМІРАЛА МАКАРОВА





# Алгоритмизация и программирование

Программирование на Kotlin  
(ч.8 – двумерные массивы)



Беркунский Е.Ю., кафедра ИУСТ, НУК  
eugeny.berkunsky@gmail.com  
<http://www.berkut.mk.ua>