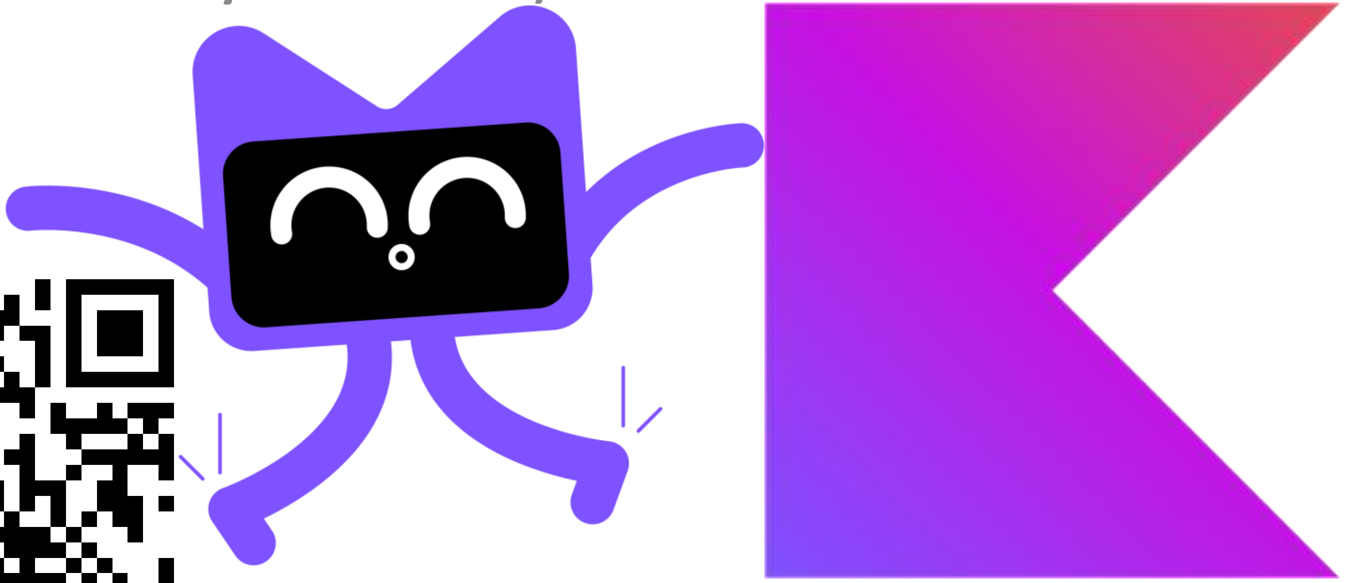
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ КОРАБЛЕБУДУВАННЯ ІМЕНІ АДМІРАЛА МАКАРОВА

Algorithms & Programming

(p.6 – Arrays & Lists)



Yevhen Berkunskyi, NUoS
eugeny.berkunsky@gmail.com
http://www.berkut.mk.ua

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ КОРАБЛЕБУДУВАННЯ ІМЕНІ АДМІРАЛА МАКАРОВА

Arrays

- Arrays in Kotlin are represented by the Array class, that has `get` and `set` functions (that turn into `[]` by operator overloading conventions), and `size` property, along with a few other useful member functions
- To create an array, we can use a library function `arrayOf()` and pass the item values to it, so that `arrayOf(1, 2, 3)` creates an array `[1, 2, 3]`

Arrays

- Another option is to use the Array constructor that takes the array size and the function that can return the initial value of each array element given its index:

```
// Creates an Array<String> with values
// ["0", "1", "4", "9", "16"]
val asc = Array(5) { i -> (i * i).toString() }
asc.forEach { println(it) }
```

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ КОРАБЛЕБУДУВАННЯ ІМЕНІ АДМІРАЛА МАКАРОВА

Primitive type arrays

- Kotlin also has specialized classes to represent arrays of primitive types without boxing overhead: `ByteArray`, `ShortArray`, `IntArray` and so on.
- These classes have no inheritance relation to the `Array` class, but they have the same set of methods and properties.
- Each of them also has a corresponding factory function

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ КОРАБЛЕБУДУВАННЯ ІМЕНІ АДМІРАЛА МАКАРОВА

Primitive type arrays

```kotlin
val x: IntArray = intArrayOf(1, 2, 3)
x[0] = x[1] + x[2]
```

```kotlin
// Array of int of size 5 with values [0,0,0,0,0]
val arr = IntArray(5)

// initialise the values in the array with a constant
// Array of int of size 5 with values [42,42,42,42,42]
val arr = IntArray(5) { 42 }

// initialise the values in the array using a lambda
// Array of int of size 5 with values [0, 1, 2, 3, 4]
// (values initialised to their index value)
var arr = IntArray(5) { it }
```

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ КОРАБЛЕБУДУВАННЯ ІМЕНІ АДМІРАЛА МАКАРОВА

List

- `List<T>` stores elements in a specified order and provides indexed access to them. Indices start from zero – the index of the first element – and go to `lastIndex` which is the `(list.size - 1)`.

```
val numbers = listOf("one", "two", "three", "four")
println("Number of elements: ${numbers.size}")
println("Third element: ${numbers.get(2)}")
println("Fourth element: ${numbers[3]}")
println("Index of element \"two\"
${numbers.indexOf("two")}")
```

List

- List elements (including nulls) can duplicate: a list can contain any number of equal objects or occurrences of a single object.
- Two lists are considered equal if they have the same sizes and structurally equal elements at the same positions.

```
val bob = Person("Bob", 31)
val people = listOf(Person("Adam", 20), bob, bob)
val people2 = listOf(
        Person("Adam", 20), Person("Bob", 31), bob)
println(people == people2)
bob.age = 32
println(people == people2)
```

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ КОРАБЛЕБУДУВАННЯ ІМЕНІ АДМІРАЛА МАКАРОВА

MutableList

- MutableList<T> is a List with list-specific write operations, for example, to add or remove an element at a specific position.

```kotlin
val numbers = mutableListOf(1, 2, 3, 4)
numbers.add(5)
numbers.removeAt(1)
numbers[0] = 0
numbers.shuffle()
println(numbers)
```

- *In some aspects lists are very similar to arrays.*
- *One important difference: an array's size is defined upon initialization and is never changed;*
- *List doesn't have a predefined size; a list's size can be changed as a result of write operations: adding, updating, or removing elements.*

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ КОРАБЛЕБУДУВАННЯ ІМЕНІ АДМІРАЛА МАКАРОВА

Task 1: Linear search in an array

- Traverse the **array** using a for loop.
- In every iteration, compare the target value with the current value of the **array**. If the values match, return the current index of the **array**. If the values do not match, move on to the next **array** element.
- If no match is found, return -1 .

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ КОРАБЛЕБУДУВАННЯ ІМЕНІ АДМІРАЛА МАКАРОВА

Task 1:
Linear search in an array

```
fun linearSearch(x:Int, a:IntArray): Int {
    for (i in 0..a.size-1) {
        if (a[i] == x) return i
    }
    return -1
}
```

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ КОРАБЛЕБУДУВАННЯ ІМЕНІ АДМІРАЛА МАКАРОВА

Task 1: Linear search in an array

```
fun linearSearch(x:Int, a:IntArray): Int {
    for (i in 0..a.size-1) {
        if (a[i] == x) return i
    }
    return -1
}
```

```
fun linearSearch1(x: Int, a: IntArray): Int {
    for (i in a.indices) {
        if (a[i] == x) return i
    }
    return -1
}
```

Task 1: Linear search in an array

```
fun linearSearch(x:Int, a:IntArray): Int {
    for (i in 0..a.size-1) {
        if (a[i] == x) return i
    }
    return -1
}
```

```
fun linearSearch1(x: Int, a: IntArray): Int {
    for (i in a.indices) {
        if (a[i] == x) return i
    }
    return -1
}
```

```
fun search(x: Int, a: IntArray): Int = a.indexOf(key)
```

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ КОРАБЛЕБУДУВАННЯ ІМЕНІ АДМІРАЛА МАКАРОВА

Task 2: Find Max / Min in an array

- Assume that it's present at the beginning of the **array** and stores that value in a variable.
- Then we compare it with the other **array** elements one by one,
  - if any element is greater than our assumed **maximum**,
  - then the **maximum** value and the index at which it occurs are updated.
- Same approach for **minimum**

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ КОРАБЛЕБУДУВАННЯ ІМЕНІ АДМІРАЛА МАКАРОВА

Task 2: Find Max / Min in an array

```
fun findMax(a: IntArray): Int {
    var result = a[0]
    for (i in 1..a.size-1) {
        if (result < a[i]) result = a[i]
    }
    return result
}
```

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ КОРАБЛЕБУДУВАННЯ ІМЕНІ АДМІРАЛА МАКАРОВА

Task 2:
Find Max / Min in an array

```
fun findMax(a: IntArray): Int {
    var result = a[0]
    for (i in 1..a.size-1) {
        if (result < a[i]) result = a[i]
    }
    return result
}
```

```
fun findMaxOrNull(a: IntArray): Int? {
    if (a.isEmpty()) return null
    return findMax(a)
}
```

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ КОРАБЛЕБУДУВАННЯ ІМЕНІ АДМІРАЛА МАКАРОВА

Task 2: Find Max / Min in an array

```
fun findMin(a: IntArray): Int {
    var result = a[0]
    for (i in 1..a.size-1) {
        if (result > a[i]) result = a[i]
    }
    return result
}
```

```
fun findMinOrNull(a: IntArray): Int? {
    if (a.isEmpty()) return null
    return findMin(a)
}
```

```
val min = a.min()
```

```
val min = a.minOrNull()
```

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ КОРАБЛЕБУДУВАННЯ ІМЕНІ АДМІРАЛА МАКАРОВА

Task 3: Calculate number of elements with condition

- Given an array of integers. The task is to calculate the count of a number of elements which are divisible by a given number k
- For example, calculate the count of even elements (divisible by 2)

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ КОРАБЛЕБУДУВАННЯ ІМЕНІ АДМІРАЛА МАКАРОВА

Task 3: Calculate number of elements with condition

```
fun calculateDivisible(a: IntArray, k: Int): Int {
    var count = 0
    for (x in a) {
        if (x % k == 0) count++
    }
    return count
}
```

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ КОРАБЛЕБУДУВАННЯ ІМЕНІ АДМІРАЛА МАКАРОВА

Task 3: Calculate number of elements with condition

```
fun calculateDivisible(a: IntArray, k: Int): Int {
    var count = 0
    for (x in a) {
        if (x % k == 0) count++
    }
    return count
}
```

```
fun calculateDivisible(a: IntArray, k: Int) = a.count {
    it % k == 0
}
```

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ КОРАБЛЕБУДУВАННЯ ІМЕНІ АДМІРАЛА МАКАРОВА

Task 4: filter elements with condition

- Given an array of integers. The task is to select elements which are divisible by a given number k and copy them into another array

Task 4: filter elements with condition

```
fun filterDivisible(a: IntArray, k: Int): IntArray {
    val list = mutableListOf<Int>()
    for (x in a) {
        if (x % k == 0) list.add(x)
    }
    return list.toIntArray()
}
```

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ КОРАБЛЕБУДУВАННЯ ІМЕНІ АДМІРАЛА МАКАРОВА

Task 4: filter elements with condition

```
fun filterDivisible(a: IntArray, k: Int): IntArray {
    val list = mutableListOf<Int>()
    for (x in a) {
        if (x % k == 0) list.add(x)
    }
    return list.toIntArray()
}
```

```
val b = a.filter { it % k == 0 }.toIntArray()
```

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ КОРАБЛЕБУДУВАННЯ ІМЕНІ АДМІРАЛА МАКАРОВА

Task 5: Binary search in an array

- Binary search works on sorted arrays.
- Binary search begins by comparing an element in the middle of the array with the target value.
- If the target value matches the element, its position in the array is returned.
- If the target value is less than the element, the search continues in the lower half of the array. If the target value is greater than the element, the search continues in the upper half of the array.
- By doing this, the algorithm eliminates the half in which the target value cannot lie in each iteration

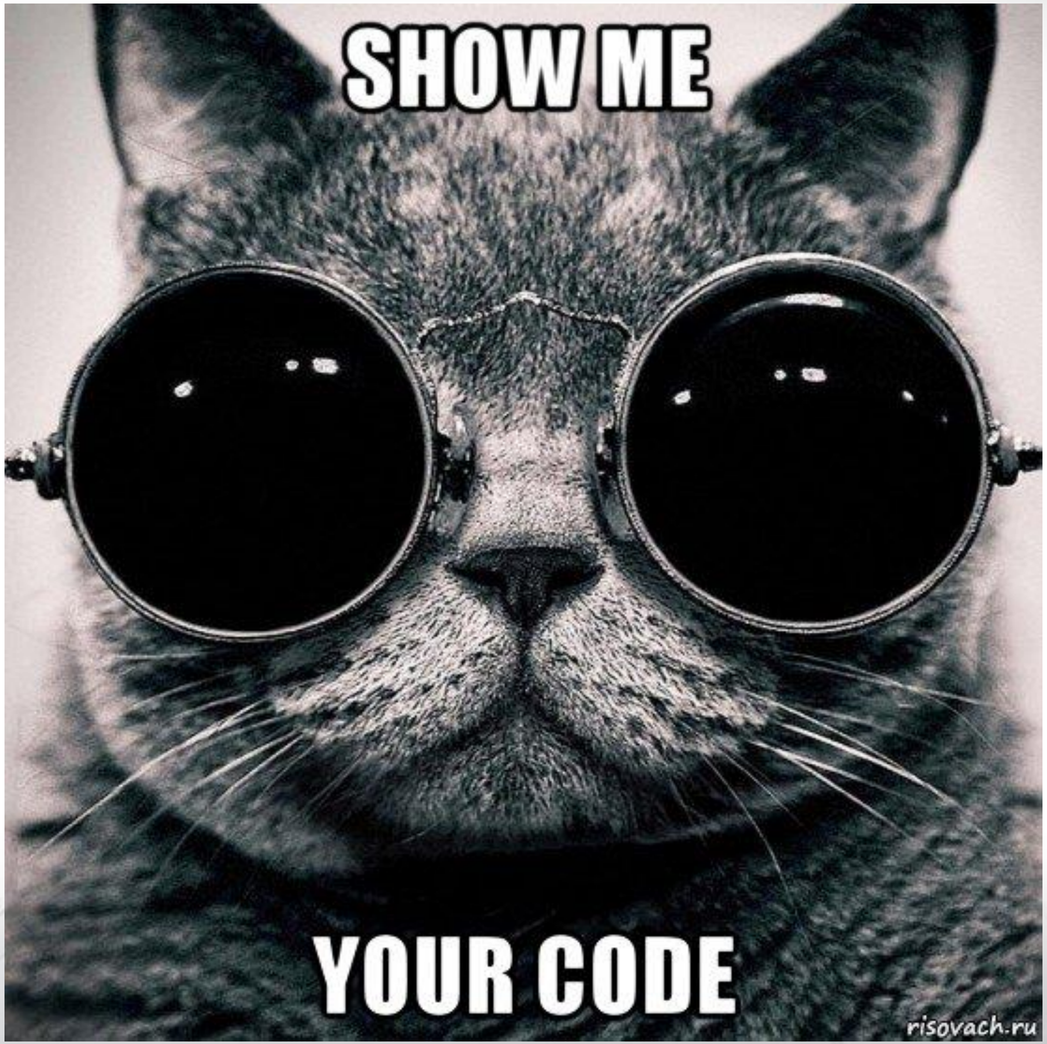
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ КОРАБЛЕБУДУВАННЯ ІМЕНІ АДМІРАЛА МАКАРОВА

Task 5: Binary search in an array

```
fun binarySearch(x: Int, a: IntArray): Int {
    var firstIndex = 0
    var lastIndex = a.size-1
    if (lastIndex == -1) return -1
    while (firstIndex <= lastIndex) {
        val avgIndex = (lastIndex + firstIndex) / 2
        if (x < a[avgIndex])
            lastIndex = avgIndex-1
        else if (x > a[avgIndex])
            firstIndex = avgIndex + 1
        else return avgIndex
    }
    return -(firstIndex+1)
}
```

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ КОРАБЛЕБУДУВАННЯ ІМЕНІ АДМІРАЛА МАКАРОВА

Let’s code!



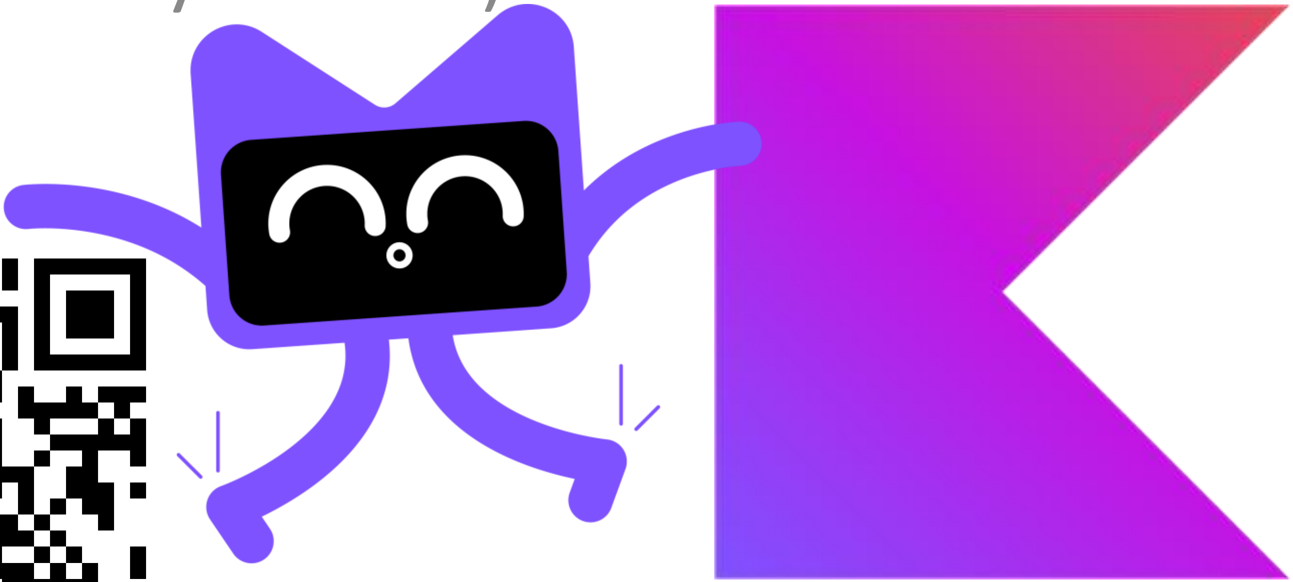
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ КОРАБЛЕБУДУВАННЯ ІМЕНІ АДМІРАЛА МАКАРОВА

Questions?

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ КОРАБЛЕБУДУВАННЯ ІМЕНІ АДМІРАЛА МАКАРОВА

Algorithms & Programming

(p.6 – Arrays & Lists)



Yevhen Berkunskyi, NUoS
eugeny.berkunsky@gmail.com
http://www.berkut.mk.ua