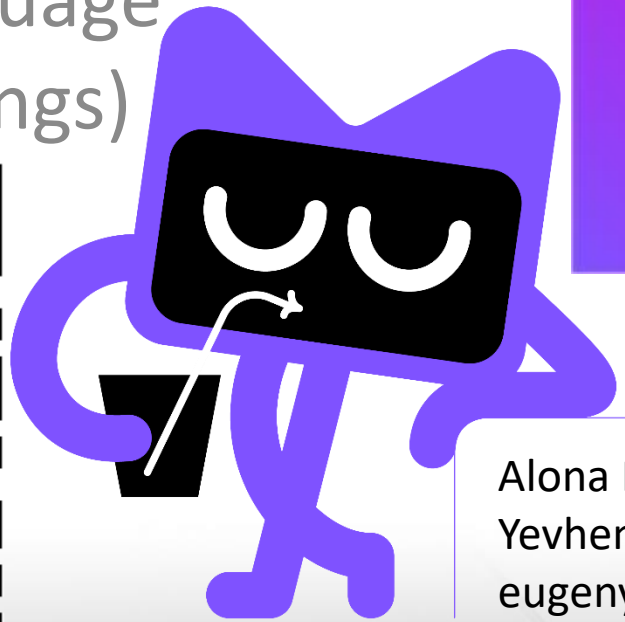
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ КОРАБЛЕБУДУВАННЯ ІМЕНІ АДМІРАЛА МАКАРОВА

# Algorithms & Programming

## Kotlin language

## (p.9 – strings)



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

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ КОРАБЛЕБУДУВАННЯ ІМЕНІ АДМІРАЛА МАКАРОВА

# Strings in Kotlin

- The String class represents character strings.
- All string literals in Kotlin programs, such as "abc", are implemented as instances of this class.
- In Kotlin, strings are represented by the String class, and they are immutable, which means their values cannot be changed after they are created.

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ КОРАБЛЕБУДУВАННЯ ІМЕНІ АДМІРАЛА МАКАРОВА

# Strings in Kotlin

Kotlin provides several ways to work with strings. Here are some common operations and examples:

- String Declaration:

```
val str: String = "Hello, Kotlin!"
```

- String Interpolation:

```
val name = "John"
val greeting = "Hello, $name!"
println(greeting) // Output: Hello, John!
```

# Strings in Kotlin

- String concatenation is the operation of joining character strings end-to-end:

```
val str1 = "Hello"
val str2 = "Kotlin"
val result = str1 + ", " + str2
println(result) // Output: Hello, Kotlin
```

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ КОРАБЛЕБУДУВАННЯ ІМЕНІ АДМІРАЛА МАКАРОВА

# Strings in Kotlin

- Multiline Strings:

```
val multilineString = """
    Line 1
    Line 2
    Line 3
""".trimIndent()
println(multilineString)
/*
   Output:
   Line 1
   Line 2
   Line 3
*/
```

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ КОРАБЛЕБУДУВАННЯ ІМЕНІ АДМІРАЛА МАКАРОВА

# Strings functions

- String Length and Indexing:

```
val str = "Kotlin"
println(str.length)
// Output: 6
println(str[0])
// Output: K
```

Strings have a "length" property - length

```
val s = "Admiral Makarov National University of Shipbuilding"
val len = s.length
println("String Length = $len")
```

String Length = 51

# Strings functions

- String Comparison:

```
fun compareTo(other: String): Int
```

Compares this object to a display of strings:

- Returns zero if this series is equal to another series,
- a negative number if it is "less than" another,
- or a positive number if it is "greater than" another.

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ КОРАБЛЕБУДУВАННЯ ІМЕНІ АДМІРАЛА МАКАРОВА

# Strings functions

- String Comparison:

```
val str1 = "apple"
val str2 = "banana"
println(str1 == str2) // Output: false
println(str1.compareTo(str2)) // Output: -1
(lexicographically smaller)
```

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ КОРАБЛЕБУДУВАННЯ ІМЕНІ АДМІРАЛА МАКАРОВА

# Strings functions

- Substring Extraction:

```
val str = "Hello, Kotlin!"
val substr = str.substring(7, 13)
println(substr) // Output: Kotlin
```

# Strings functions

- String Conversion:

```
val number = 42
val str = number.toString()
```

# Strings functions

- String Conversion:

```
val number = 42
val str = number.toString()
```

- These are just some of the common operations you can perform with strings in Kotlin.
- The **String** class provides many more functions for manipulating and working with strings.

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ КОРАБЛЕБУДУВАННЯ ІМЕНІ АДМІРАЛА МАКАРОВА

# Strings functions

## Equals function:

```
fun equals(other: Any?): Boolean
```

Indicates whether any other object is equal to this.
For strings (most other objects in Kotlin) you can use ==

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ КОРАБЛЕБУДУВАННЯ ІМЕНІ АДМІРАЛА МАКАРОВА

# Strings functions

## Get function:

```
fun get(index: Int): Char
```

Returns the character of this string at the specified index. Throws an IndexOutOfBoundsException if the index is outside of this string

Can be used in the form [ … ]

# String Templates

Let's say hello to the cat named "Peach"

```
val catName: String = "Peach"
println("Hi $catName! How are you?")
```

Of course, you can also do this:

```
val catName = "Peach"
```

# Extension functions

- The JetBrains developers have added many out-of-the-box extension functions for many classes, including strings. You can find them in the String.kt file (in IntelliJ IDEA, double-click the Shift key and type the name of this file in the search bar to view the source).

- Some examples of extension functions will be discussed below. In fact, there are many more, study them yourself.

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ КОРАБЛЕБУДУВАННЯ ІМЕНІ АДМІРАЛА МАКАРОВА

# String as an array

- A string can be thought of as an array of characters.

```
val cat = "Peach"
val character = cat[2]
println(character) // Output a
```

# Walk along the string

- Loop through the entire string without using an index

```
val cat = "Peach"

for(char in cat){
    println(char)
}
```

- Loop through entire row using index

```
val cat = "Peach"

for (char in cat.indices){
    print(cat[char] + "\n")
}
```

# Walk along the string

- The string itself can be explicitly converted to an array first

```
val cat = "Peach"
for(char in cat.toCharArray){
    println(char)
}
```

- You can call forEach

```
cat.forEach { char -> println(char) }
```

- If you need not only the character of the string, but also its index, then call forEachIndexed

```
cat.forEachIndexed {
   index, char -> println("Index $index Character $char")
}
```

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ КОРАБЛЕБУДУВАННЯ ІМЕНІ АДМІРАЛА МАКАРОВА

# Built-in Functions

```
    val blank = "   ".isBlank()
// true if empty string or empty space characters,
// tabs, etc.

// index of the last character
    val lastIndex = "cat Peach".lastIndex // 8

// uppercase the first character of the string
// decapitalize() do the inverse
    val capitalize = "cat Peach".capitalize()

// add a space before the line
    val withSpaces = "1".padStart(2)

// "100" add zeros to the end
    val endZeros = "1".padEnd(3, '0')
```

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ КОРАБЛЕБУДУВАННЯ ІМЕНІ АДМІРАЛА МАКАРОВА

# Built-in Functions

```
    val dropStart = "Kotlin".drop(2) // "tlin"
// remove the first characters in the specified number
    val dropEnd = "Kotlin".dropLast(3) // "Kot"
// remove the last characters in the specified amount

// return the string without the first character,
// which satisfies the condition
    val string = "Peach"
    val result = string.dropWhile{
        it == 'P'
    }
    println(result) // each

// return the string without the last character,
// which satisfies the condition
    val string = "Peachtt"
    val result = string.dropLastWhile{
        it == 't'
    }
    println(result) // Peach
```

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ КОРАБЛЕБУДУВАННЯ ІМЕНІ АДМІРАЛА МАКАРОВА

# Built-in Functions

```
// split into an array of strings
"A\nB\nC".lines() // [A, B, C]
"ABCD".zipWithNext() // [(A, B), (B, C), (C, D)]

// remove characters from the given range
val string = "The cat that walked by itself"
val result = string.removeRange(
    3..28 // range
)
```

# Built-in Functions

```
// remove the prefix from the string
    val string = "The cat that walked by itself"
    val result = string.removePrefix("The")

// remove the suffix from the string
    val string = "The cat that walked by itself"
    val result = string.removeSuffix("itself")

// remove the specified separator, which should surround
// the string from the beginning and from the end
    val string = "та, тра-та-та, мы везём с собой кота"

    val result = string.removeSurrounding(
        "та" // delimiter
    )

    println(result) // , тра-та-та, мы везём с собой ко
```

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ КОРАБЛЕБУДУВАННЯ ІМЕНІ АДМІРАЛА МАКАРОВА

# Built-in Functions

```
// You can also specify different start and end,
// which surround the string
val string = "Тра-та-та, тра-та-та, мы везём с собой кота"

val result = string.removeSurrounding(
    "Тра-", // prefix
    " кота" // suffix
)

println(result) // та-та, тра-та-та, мы везём с собой
```

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ КОРАБЛЕБУДУВАННЯ ІМЕНІ АДМІРАЛА МАКАРОВА

# Built-in Functions

```
// Add indents for explicit newline
val string =
    "Some long text, \n consisting of cat names: " +
            "\n Tom" +
            "\n Cat" +
            "\n Max"

val result = string.prependIndent(
    "      " // indent
)
```

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ КОРАБЛЕБУДУВАННЯ ІМЕНІ АДМІРАЛА МАКАРОВА

# Built-in Functions

```
// Divide characters into two groups.
// Uppercase characters will fall into the first group,
// second - lowercase characters
val string = "Cat Apricot and cat Peach are Friends!"

val result: Pair<String, String> = string.partition {
    it.isUpperCase()
}

println(result.first + " : " + result.second)
//CAPF : at pricot and cat each are riends!
```

# Built-in Functions

```
// Split the string into a list of strings.
// As a separator - newline
val s1 = "Cat Apricot\nCatPeach\n Cat Peach"

// Split string into lines (CRLF, LF or CR)
val lines: List<String> = s1.lines()

println("Number of lines: ${lines.size}")

// Space as delimiter
val s2 = " Cat Apricot Cat Peach Cat Peach"
val lines2: List<String> = s2.split(" ")

println("Number of lines: ${lines2.size}")
```

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ КОРАБЛЕБУДУВАННЯ ІМЕНІ АДМІРАЛА МАКАРОВА

# Using Predicates

```
// Whether the string contains only numbers
// (using a predicate)
val string = "09032020"

// Returns true if all characters match the given predicate
val result: Boolean = string.all{
    it.isDigit()
}
```

```
// Does the string contain at least one digit
// (use predicate)
val string = "3 cats"
// Returns true if at least one character matches the
// given predicate
val result: Boolean = string.any() {
    it.isDigit()
}
```

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ КОРАБЛЕБУДУВАННЯ ІМЕНІ АДМІРАЛА МАКАРОВА

# Built-in Functions

- To replace individual characters or strings, use the replace() function

```
val s1 = "Cot Cot"

val r1 = s1.replace(
    'o', // old char
    'a', // new char
    true // ignore case Boolean = false
)
```

```
val str = "Cat is man's best friend"
val res = string.replace(
    "Cat", // old value
    "Dog", // new value
    true // ignore case
)
```

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ КОРАБЛЕБУДУВАННЯ ІМЕНІ АДМІРАЛА МАКАРОВА

# Built-in Function

```
fun toDragonSpeak(phrase: String) =
    phrase.replace(Regex("[aeiou]")) {
        when (it.value) {
            "a" -> "4"
            "e" -> "3"
            "i" -> "1"
            "o" -> "0"
            "u" -> "|_|"
            else -> it.value
        }
    }
```

```
println(toDragonSpeak("Kitten")) // K1tt3n
```

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ КОРАБЛЕБУДУВАННЯ ІМЕНІ АДМІРАЛА МАКАРОВА

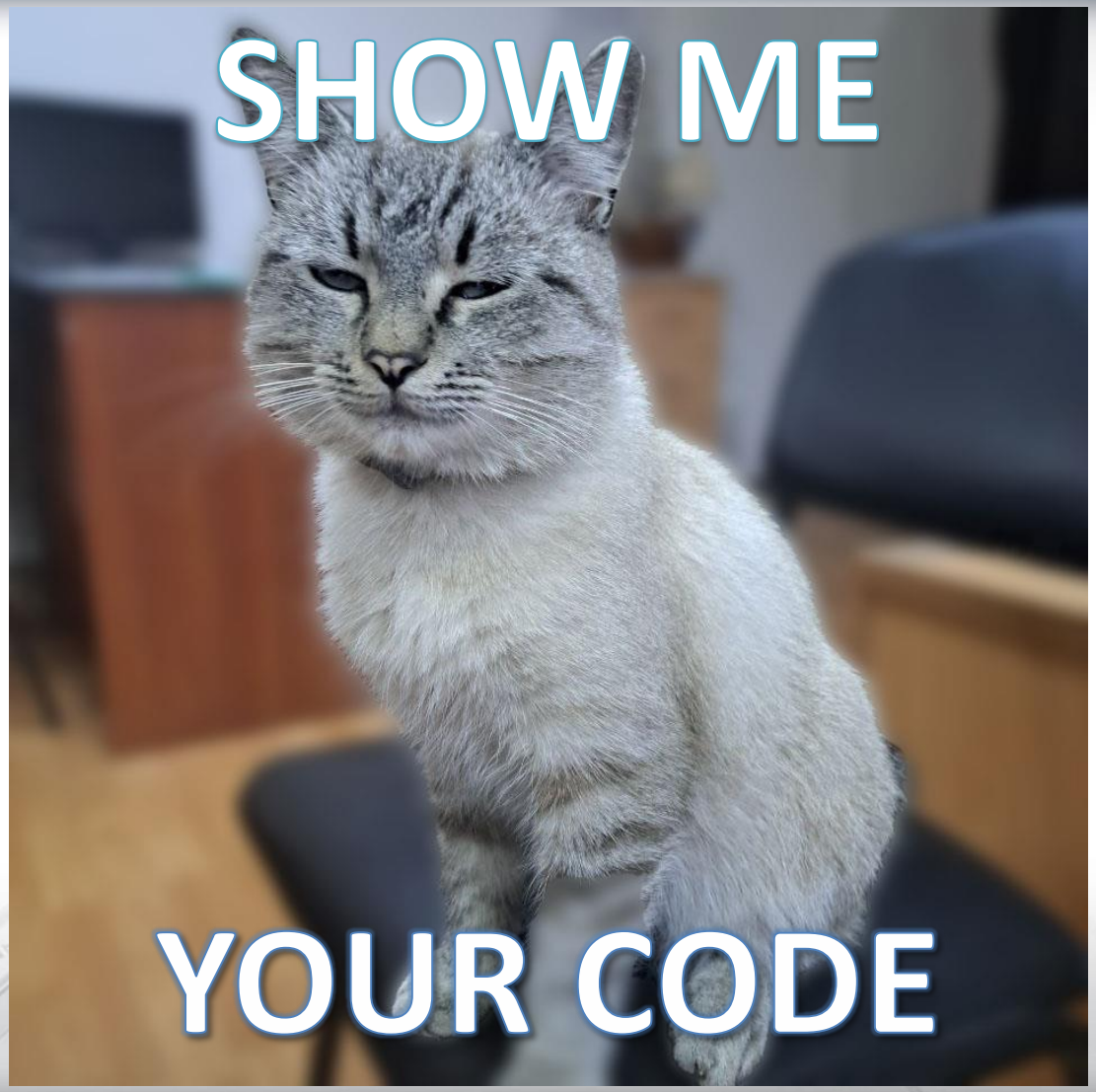
# Practical task

Practical task: Solve 16 problems from the competition “Strings”

https://www.eolymp.com/uk/contests/37015

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ КОРАБЛЕБУДУВАННЯ ІМЕНІ АДМІРАЛА МАКАРОВА

# Let’s code!



НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ КОРАБЛЕБУДУВАННЯ ІМЕНІ АДМІРАЛА МАКАРОВА

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ КОРАБЛЕБУДУВАННЯ ІМЕНІ АДМІРАЛА МАКАРОВА

# Algorithms & Programming

## Kotlin language

## (p.9 – strings)

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