

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ  
«БРЕСТСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

ФАКУЛЬТЕТ ЭЛЕКТРОННО-ИНФОРМАЦИОННЫХ СИСТЕМ

Кафедра интеллектуальных информационных технологий

Отчет по лабораторной работе №2

Выполнил  
Васильюк П. О.,  
студент группы ПО-5

Проверил  
Крощенко А.А.,  
ст. преп. Кафедры ИИТ,  
«\_\_» \_\_\_\_\_ 2021 г.

Брест, 2021

**Цель работы: приобрести базовые навыки работы с файловой системой в Kotlin.**

## **Вариант 2**

**Задание 1. Напишите программу, которая читает текст построчно, а затем разбивает каждую строку на лексемы и выводит их в обратном порядке.**

**Задание 2. Утилита tail выводит несколько (по умолчанию 10) последних строк из файла. Формат использования: tail [-n] file. Ключ -n <количество строк> (или просто <количество строк>) позволяет изменить количество выводимых строк.**

### **Выполнение:**

#### **Код программы:**

```
import java.io.File
import java.io.InputStream

fun main() {
    task_1()
    task_2()
}

fun task_1() {
    val inputStream: InputStream = File("file.txt").inputStream()
    val lineList = mutableListOf<String>() //buff for file content
    var splitList = arrayListOf<List<String>>()

    inputStream.bufferedReader().useLines { lines -> lines.forEach {
lineList.add(it) } }
    lineList.forEach{splitList.add(it.split(" "))}

    println("Regular line")
    for(i in 0 until splitList.size){
        println(splitList[i])
    }

    println("Reversed line")
    for(i in 0 until splitList.size){
        println(splitList[i].reversed())
    }
}

fun task_2() {
    print("\nEnter utility: ")
    val Line:String = readLine().toString()
    val splitLine:List<String> = Line.split(" ")
    val lineList = mutableListOf<String>() //buff for file content

    var numCheck:Boolean = false
    for(it in splitLine)
    {
        if(it.matches(Regex("(\\d+)")) numCheck = true
    }
```

```

        } //check for numbers of lines

        when{
            splitLine.contains("tail") || Line.contains(Regex("(\\w+)\\. (\\w+)"))
-> {
                when{
                    splitLine.contains("-n") || numCheck -> {
                        val number = Regex("(\\d+)").find(Line)

//numbers of lines
                        val filename =
Regex("(\\w+)\\. (\\w+)").find(Line)

                        if (filename != null && number != null) {
                            val inputStream: InputStream =
File(filename.value).inputStream()
                            inputStream.bufferedReader().useLines
{ lines -> lines.forEach { lineList.add(it) } }
                            val lineNumber = lineList.count() -
number.value.toInt() //line number we will start with

                            if(lineNumber < 0){
                                for(it in lineList)
                                    println(it)
                            }
                            else{
                                for(i in lineNumber until
lineList.count())
                                    println(lineList[i])
                            }
                        }
                    }
                }
            }
        }
    }
}
else -> {
    val number = 10
    val filename =
Regex("(\\w+)\\. (\\w+)").find(Line)

    if (filename != null) {
        val inputStream: InputStream =
File(filename.value).inputStream()
        inputStream.bufferedReader().useLines
{ lines -> lines.forEach { lineList.add(it) } }
        val lineNumber = lineList.count() -
number //line number we will start with

        if(lineNumber < 0){
            for(it in lineList)
                println(it)
        }
        else{
            for(i in lineNumber until
lineList.count())
                println(lineList[i])
        }
    }
}
}
}
else -> println("Error, wrong utility!")

```

```
    }
}
```

## Результаты работы программы:

```
Regular line
[Loren, ipsum, dolor, sit, amet, consectetur, adipiscing, elit, parturient, eros, pharetra, habitant., ]
[Lacinia, sollicitudin, mus, congue, tellus, eu, ultricies, sit, pharetra, consectetur., ]
[Sodales, rhoncus, dictumst, nec, convallis, etiam, vestibulum, aliquam, sagittis, dictum., nibh, fames, ut, ipsum, inceptos., ]
[Turpis, fames, velit, placerat, vitae, nascetur, ornare, sen, netus.,, scelerisque, primis., ]
[Purus, sit, natoque, interdum, luctus, velit, senectus, eu, dignissim, morbi.,, tincidunt.]

Reversed line
[, habitant., pharetra, eros, parturient, elit, adipiscing, consectetur, amet, sit, dolor, ipsum, Loren]
[, consectetur., pharetra, sit, ultricies, eu, tellus, congue, mus, sollicitudin, Lacinia]
[, inceptos., ipsum, ut, fames, nibh, dictum.,, sagittis, aliquam, vestibulum, etiam, convallis, nec, dictumst, rhoncus, Sodales]
[, primis., scelerisque, netus.,, sen, ornare, nascetur, vitae, placerat, velit, fames, Turpis]
[tincidunt., morbi.,, dignissim, eu, senectus, velit, luctus, interdum, natoque, sit, Purus]
```

file2.txt - Блокнот

Файл Правка Формат Вид Справка

Лишь многие известные личности, вне зависимости от их уровня, должны быть описаны максимально подробно.

Но акционеры крупнейших компаний набирают популярность среди определенных слоев населения, а значит, должны быть ограничены исключительно также как консультация с широким активом влечет за собой процесс внедрения и модернизации распределения внутренних резервов и ресурсов.

Равным образом, сплоченность команды профессионалов является качественно новой ступенью модели развития.

Каждый из нас понимает очевидную вещь: высококачественный прототип будущего проекта требует определения и уточнения направлений прогресса.

Но многие известные личности могут быть указаны как претенденты на роль ключевых факторов.

В частности, высокое качество позиционных исследований однозначно определяет каждого участника как способного принимать собственные решения.

Банальные, но неопровержимые выводы, а также элементы политического процесса являются только методом политического участия и рассмотрены.

Имеется спорная точка зрения, гласящая примерно следующее: элементы политического процесса преданы социально-демократической анафеме.

Базовые сценарии поведения пользователей объединены в целые кластеры себе подобных.

Но интерактивные прототипы представляют собой не что иное, как квинтэссенцию победы маркетинга над разумом и должны быть указаны как прет.

Но интерактивные прототипы объявлены нарушающими общечеловеческие нормы этики и морали!

Явные признаки победы институционализации описаны максимально подробно.

Enter utility: `tail 12 file2.txt`

Но акционеры крупнейших компаний набирают популярность среди определенных слоев населения,

Также как консультация с широким активом влечет за собой процесс внедрения и модернизации

Равным образом, сплоченность команды профессионалов является качественно новой ступенью мод

Каждый из нас понимает очевидную вещь: высококачественный прототип будущего проекта требует

Но многие известные личности могут быть указаны как претенденты на роль ключевых факторов.

В частности, высокое качество позиционных исследований однозначно определяет каждого участ

Банальные, но неопровержимые выводы, а также элементы политического процесса являются толь

Имеется спорная точка зрения, гласящая примерно следующее: элементы политического процесса

Базовые сценарии поведения пользователей объединены в целые кластеры себе подобных.

Но интерактивные прототипы представляют собой не что иное, как квинтэссенцию победы маркет

Но интерактивные прототипы объявлены нарушающими общечеловеческие нормы этики и морали!

Явные признаки победы институционализации описаны максимально подробно.

**Вывод: приобрел базовые навыки работы с файловой системой в Kotlin.**