Министерсво образоваия Республики Беларусь

Учреждение образования

«Бресткий государтвенный технический университет»

Кафедра ИИТ

Лабораторная работа №2

за 5 семестр

По дисциплине «СПП»

Выполнил:

Студент 3 курса

Группы ПО-4(2)

Кречко К.А.

Проверил:

Монтик С.В.

Брест 2021



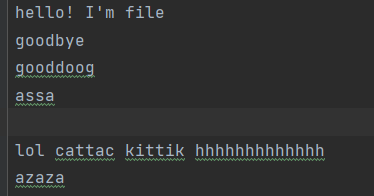




**Код:**

import java.io.File  
  
fun task1(){  
 val input = File("text.txt")  
 val array = *countAndCreatingArrayOfPalindromes*(input)  
  
 *sortingPalindromes*(array)  
 for(i in array)  
 *print*("$i ")  
}  
  
fun istPalindrome(word: String): Boolean {  
 word.*toCharArray*()  
 if (word.length == 0)  
 return false  
 var i1 = 0  
 var i2 = word.length - 1  
 while(i2 > i1){  
 if(word[i1] != word[i2]) return false  
 ++i1  
 --i2  
 }  
 return true  
}  
  
fun countAndCreatingArrayOfPalindromes (input: File): Array<String> {  
 var count = 0  
 var array = *emptyArray*<String>()  
  
 for(line in input.*readLines*()) {  
 for (word in line.*split*(" ")) {  
 if (*istPalindrome*(word)) {  
 count++  
 array += word  
 }  
 }  
 }  
 *print*("Количество палиндромов= $count\n")  
 return array  
}  
  
fun sortingPalindromes(array: Array<String>): Array<String>{  
 for (i in array.*indices*) {  
 for (j in i+1 *until* array.size) {  
 val tempI = array[i]  
 val tempJ = array[j]  
  
 if (tempI.length<tempJ.length){  
 array[i] = array[j]  
 array[j] = tempI  
 }  
 }  
 }  
 return array  
}

**Исходный файл:**

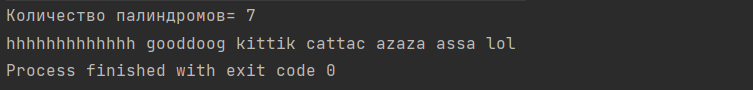
****

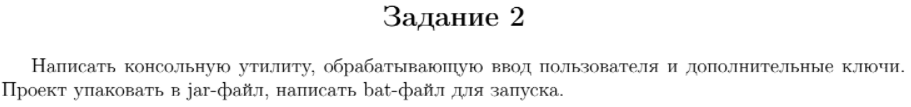
**Вызов в main:**

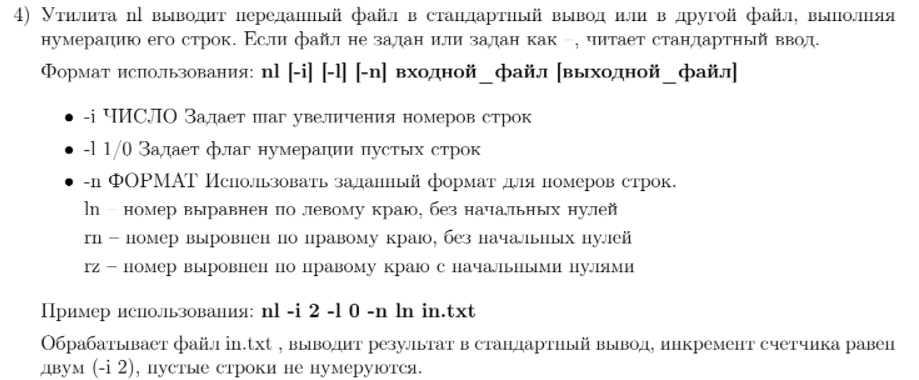
*task1*()

**Результат:**

****

****

****

****

**Код:**

import java.io.BufferedWriter  
import java.io.File  
  
fun getParameters(): Array<String>?  
{  
 val inputParametersFile = File("input.txt")  
  
 if(inputParametersFile.length() == 0L)  
 {  
 *println*("Чтения из файла не произошло. Введите утилиту")  
 return *readLine*()?.*split*(" ")?.*toTypedArray*<String>()  
 }  
  
 var parameters = *arrayOf*<String>()  
 for (line in inputParametersFile.*readLines*())  
 for (word in line.*split*(" "))  
 parameters += word  
 return parameters  
}  
  
fun foundMaxLength(numerationFile: File): Int{  
 var max = 0  
  
 for(line in numerationFile.*readLines*())  
 if (line.length > max)  
 max = line.length  
  
 return max  
}  
  
fun ln\_numeration(numerationFile: File, step: Int, flagNumeration: Int, output: String)  
{  
 var count = step  
 val writer: BufferedWriter = File(output).*bufferedWriter*()  
  
 for(line in numerationFile.*readLines*()) {  
  
 if (line.*isNotEmpty*() || flagNumeration == 1) {  
 if(output != "0")  
 writer.write("$count\t")  
 else  
 *print*("$count\t")  
 count += step  
 }  
  
 if(output != "0")  
 writer.write("$line\n")  
 else  
 *println*(line)  
 }  
 writer.close()  
}  
  
fun rn\_numeration(numerationFile: File, step: Int, flagNumeration: Int, output: String, format:String){  
  
 val rightBorder = *foundMaxLength*(numerationFile) + 5  
 var count = step  
  
 for(line in numerationFile.*readLines*()) {  
 val writer: BufferedWriter = File(output).*bufferedWriter*()  
  
 if(output != "0") //выводим строку  
 writer.write(line)  
 else  
 *print*(line)  
  
 var size = line.length  
 while(rightBorder > size){ //выводим отступы  
 if(output != "0")  
 writer.write(" ")  
 else  
 *print*(" ")  
 size++  
 }  
  
 if(line.*isNotEmpty*() || flagNumeration == 1) //выводим нумерацию  
 {  
 if(format == "rz") {  
 if(output != "0")  
 writer.write("0$count")  
 else  
 *println*("0$count")  
 }  
 else {  
 if(output != "0")  
 writer.write(count)  
 else  
 *println*(count)  
 }  
 count += step  
 }  
 else  
 *println*(" ")  
 }  
}  
  
fun task2()  
{  
 val parameters = *getParameters*()  
  
 if (parameters != null && parameters[0] == "nl")  
 {  
 val step = parameters[2].*toInt*()  
  
 val flagNumeration = parameters[4]  
 if(flagNumeration != "0" && flagNumeration != "1")  
 throw IllegalArgumentException()  
  
 val format = parameters[6]  
 if (format != "ln" && format != "rn" && format != "rz" )  
 throw IllegalArgumentException()  
  
 val numerationFile = File(parameters[7])  
  
 val output: String  
 output = if(parameters.size == 9)  
 parameters[8]  
 else  
 "0"  
  
 if (format == "ln")  
 *ln\_numeration*(numerationFile, step, flagNumeration.*toInt*(), output)  
 if (format == "rn" || format == "rz")  
 *rn\_numeration*(numerationFile, step, flagNumeration.*toInt*(), output, format)  
 }  
}

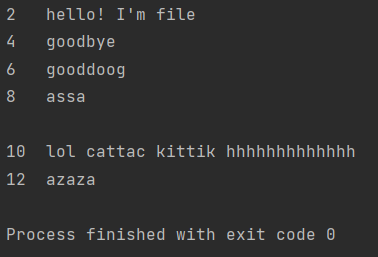
**input.txt (Входной файл)**

****

**Вызов в main:**

*task2*()

**Результат:**

****

**Вывод:** Я приобрел базовые навыки работы с файловой системой в Java