Міністерство освіти і науки України

Харківський національний університет ім. В. Н. Каразіна

Факультет комп’ютерних наук

Кафедра безпеки інформаційних систем і технологій

Лабораторна робота №5

*з навчальної дисципліни*

«Спеціалізовані мови програмування та проектування електронних елементів і систем»

Виконав:

Студент групи КБ-41

Кравченко Є.М.

Перевірив:

Олійников Р.В.

Харків – 2020 р.

1.С клавиатуры ввести несколько слов (фраз), объединить их через StringBuilder (каждый раз выводя на печать текущую емкость), распечатать итоговое значение.

**public static void** task1() {  
 StringBuilder stringBuilder = **new** StringBuilder();  
 Scanner scanner = **new** Scanner(System.***in***);  
 **for** (**int** i = 0; i < 3; i++) {  
 stringBuilder.append(scanner.nextLine());  
 System.***out***.println(**"String Builder capacity: "** + stringBuilder.capacity());  
 }  
}

2.Перебирая символы строки, распечатать информацию о каждом: является ли цифрой, символом (верхнего или нижнего регистра) или пробелом.

**public static void** task2(String str) {  
 **for** (**char** c : str.toCharArray()) {  
 System.***out***.println(c);  
 System.***out***.println(**"Number: "** + (!Character.*isLetter*(c) && !Character.*isSpaceChar*(c)));  
 System.***out***.println(**"Symbol: "** + Character.*isLetter*(c));  
 System.***out***.println(**"Upper case: "** + Character.*isUpperCase*(c));  
 System.***out***.println(**"Is space: "** + Character.*isSpaceChar*(c));  
 System.***out***.println();  
 }  
}

3.Реализовать программу, которая получает два имени текстовых файлов через параметр командной строки, затем выполняет копирование одного файла в другой, без использования системных функций копирования (открывать, читать и записывать данные - самостоятельно, через классы FileInputStream, FileOutputStream, PrintWriter, FileWriter и т.п.). Использовать автозакрытие файлов через try(), обрабатывать ошибки ввода-вывода (печатать про них информацию).

**private static void** task3(String fileName1, String fileName2) {  
 **try** (FileInputStream fileInputStream = **new** FileInputStream(fileName1);  
 FileOutputStream fileOutputStream = **new** FileOutputStream(fileName2)) {  
 **int** a;  
 **while** ((a = fileInputStream.read()) != -1) {  
 fileOutputStream.write(a);  
 }  
 } **catch** (IOException e) {  
 System.***out***.println(e.getMessage());  
 }  
}

4.Разработать программу, запрашивающую имя файла у пользователя, затем печатающего содержимое файла как шестнадцатеричный дамп на экране. В случае возникновения ошибки (файл не найден, запрет доступа) программа также печатает сообщение об ошибке. В конце работы программа в любом случае выводит сообщение (любое) о завершении собственной работы.

**private static void** task4() {  
 String filename;  
 **while** (**true**) {  
 System.***out***.println(**"Input file name: "**);  
 filename = **new** Scanner(System.***in***).nextLine();  
 **if** (**""**.equals(filename)) {  
 System.***out***.println(**"Work with program ended"**);  
 **break**;  
 }  
 **try** (FileInputStream fis = **new** FileInputStream(filename)) {  
 **int** a;  
 **while** ((a = fis.read()) != -1) {  
 System.***out***.printf(**"%X "**, a);  
 }  
 System.***out***.println();  
 } **catch** (IOException e) {  
 System.***out***.println(e.getMessage());  
 }  
 }  
}

5.Разработать класс, реализующий интерфейс Closeable. В конструкторе вывести сообщение о создании, в методе close() - о закрытии. Проверить функционирование блока try() с ресурсами.

**private static void** task5() {  
 **try** (MyCloseable myCloseable = **new** MyCloseable()) {  
 System.***out***.println(**"Work"**);  
 } **catch** (IOException e) {  
 e.printStackTrace();  
 }  
}

6.Каскадирование: разработать три собственных класса, реализующих исключения. В блоке try {}, в зависимости от пользовательского ввода, выбросить одно из двух. Перехватывать оба возможных варианта, в обработчике выбросить третье исключение и позже перехватить его. На финальном этапе (перехвате третьего) распечатать стек вызовов каждого из перехваченных исключений.

**private static void** task6() {  
 System.***out***.println(**"1: Exception 1"** + System.*lineSeparator*() + **"2: Exception 2"**);  
 **try** {  
 **switch** (**new** Scanner(System.***in***).nextInt()) {  
 **case** 1 -> **throw new** MyException1();  
 **case** 2 -> **throw new** MyException2();  
 }  
 } **catch** (MyException1 | MyException2 myException1) {  
 System.***out***.println(Arrays.*toString*(myException1.getStackTrace()));  
 **try** {  
 **throw new** MyException3();  
 } **catch** (MyException3 myException3) {  
 System.***out***.println(Arrays.*toString*(myException3.getStackTrace()));  
 }  
 }  
}