Задание 1

Цель: Ввести n строк с консоли. Вывести на консоль те строки, длина которых меньше средней, также их длины.

Блок схема

Нахождение средней длины

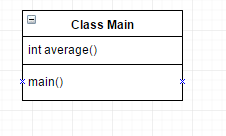


Основная функция Main()



Проект-схемы иерархии классов задачи. Описание классов и методов.

Схема иерархии классов



Класс Main - основной класс выполнении программы

Метод average – нахождение длины средней строки

Метод main() – выполнение основной задачи программы, а именно вывод в консоль те строки, длина которых меньше средней, также их длины.

Программный код реализации задачи.

**import** java.io.IOException;  
**import** java.util.ArrayList;  
**import** java.util.List;  
**import** java.util.Scanner;  
*/\*Первая лабораторная \*/  
/\*Ввести n строк с консоли. Вывести на консоль те строки, длина которых меньше сред-ней, также их длины.\*/***public class** Main { *// Класс Меин* **public static int** average(List <String> st) *// Функция для определение самой большой и самой маленькой строки и нахождение среднего* {  
 **int** minimun = st.get(0).length();  
 **int** maximum = st.get(0).length();  
 **int** av = 0;*// среднее значение* **for** (String s: st) *// прохождение списка через цикл* {  
  
 **if** (s.length() < minimun)  
 {  
 minimun = s.length();*// длина элемента списка* }  
 **if** (s.length() > maximum)  
 {  
 maximum = s.length();  
  
 }  
  
 }  
  
 av = (minimun + maximum)/2;*// нахождение среднего* **return** av;  
 }  
  
  
  
  
 **public static void** main(String[] args) **throws** IOException {  
 **int** av = 0;  
 **boolean** f = **false**;  
  
 System.***out***.print(**"Введите количество строк "**);  
 Scanner n = **new** Scanner(System.***in***);*// создаём объект класса Scanner* **int** number = n.nextInt(); *//считывает целое число с потока ввода и сохраняем в переменную* Scanner in = **new** Scanner(System.***in***);*// создаём объект класса Scanner* List<String> strings = **new** ArrayList<String>(); *//создает пустой список из строк* **for**(**int** i=1;i <= number; i++){  
 System.***out***.println(**"Строка номер "**+(i));  
 strings.add(in.next());*// добавление в список строк* }  
 av = *average*(strings);*// функция по нхождению среднеего числа строк* System.***out***.println(**"Вывод слов длина, которых меньше средней, средняя длина "** + av);  
  
 **for** (String s: strings) {  
 **if** (s.length() < av)  
 {  
 f = **true**;  
 System.***out***.println(**"Строка "**+s+**" Длина "**+ s.length()); *// Вывод* }  
  
  
 }  
 System.***out***.println();  
 **if** (f == **false**){  
 System.***out***.println(**"Таких строк нет"**);  
 }  
  
  
 }  
  
}

План тестирования по задачи.

|  |  |  |  |
| --- | --- | --- | --- |
| N – количество строк | Ввод строк | Вывод средней длины | Вывод строк |
| 5 | Привет, няня, как,дела, вявя | 4 | Строка как Длина 3 |
| 8 | Привет, няня, как, дела, вявя, ны, ла, ряр, мяу | 4 | Строка как Длина 3  Строка ны Длина 2  Строка ла Длина 2  Строка ряр Длина 3  Строка мяу Длина 3 |
| 6 | Ты, Мой, Маленький ,пирожок, Даа, Нееет | 5 | Строка Ты Длина 2  Строка Мой Длина 3  Строка Даа Длина 3 |
| 3 | Я, Я, Я | 1 | Таких строк нет |
| 4 | 1234, 21, 12, 32 | 3 | Строка 21 Длина 2  Строка 12 Длина 2  Строка 32 Длина 2 |

Скриншоты работы программы.

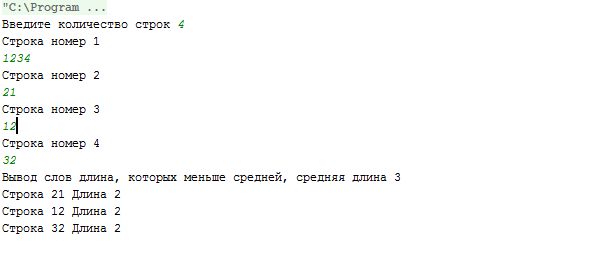


Рис. 1

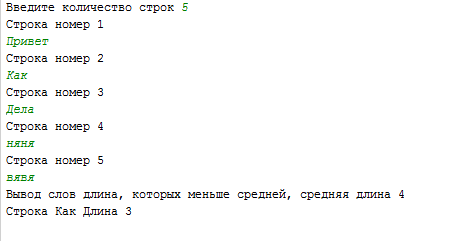


Рис. 2

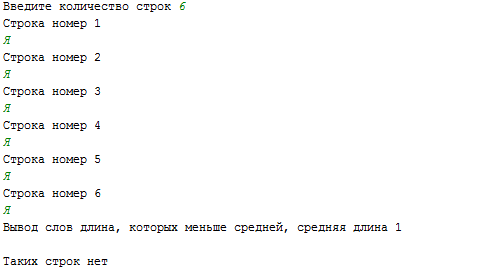


Рис. 3

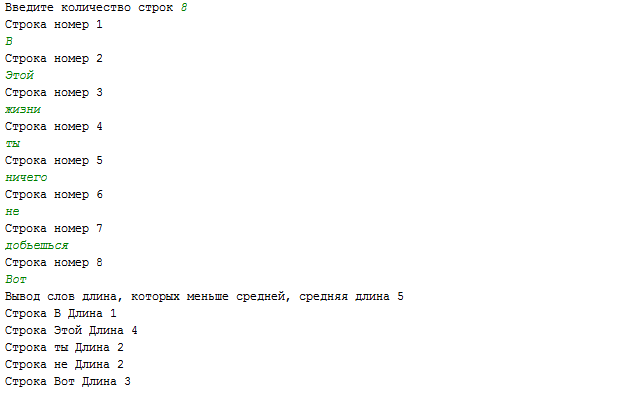


Рис 4.

Вывод: были получены навыки в разработке консольных java-приложений

Задание 2

Цель: Написать регулярное выражение, определяющее является ли заданная строка правильным MAC-адресом.:

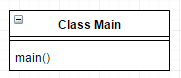
1. пример правильных выражений: aE:dC:cA:56:76:54.
2. пример неправильных выражений: 01:23:45:67:89:Az.

Блок-схема задачи

Функция main()



Проект-схемы иерархии классов задачи. Описание классов и методов.



Класс Main - основной класс выполнении программы

Метод main() – выполняет основную задачу, а именно выполняет регулярное выражение, определяющее является ли заданная строка правильным MAC-адресом.

Программный код реализации задачи.

**import** java.util.Scanner;  
**import** java.util.regex.Matcher;  
**import** java.util.regex.Pattern;  
*/\*3. Написать регулярное выражение, определяющее является ли заданная строка  
правиль-ным MAC-адресом. – пример правильных выражений: aE:dC:cA:56:76:54.  
– пример неправильных выражений: 01:23:45:67:89:Az. \*/***public class** Main {  
  
 **public static void** main(String[] args) {  
*// Ввод, мас адрес не введен* Pattern p1 = Pattern.*compile*(**"([a-f]+[A-F]+:){3}+([0-9]{2}+:){2}+[0-9]{2}"**); */\*регулярное выражение\*/* System.***out***.print(**"Введите Мас адрес "**);  
 Scanner n = **new** Scanner(System.***in***);*// создаём объект класса Scanner* String str = n.next(); *//считывает целое число с потока ввода и сохраняем в переменную* Matcher m1 = p1.matcher(str); *// возвращает true, если шаблон соответствует всей строке, иначе false.* **if** (str.length() == 0)  
 {  
 System.***out***.println(**"Введена пустая строка"**);  
 }  
 **else** {  
 **boolean** b = m1.matches();  
 String m = b ? **"Введен правильный MAC адрес: "** + str : **"Введен не правильный MAC адрес: "** + str;  
 System.***out***.println(m);  
 }  
 */\*re+ Одно или более совпадений предыдущего выражения.  
 re{n} Соответствует ровно N Количество вхождений предшествующего выражения.  
 (?: re) Группы регулярных выражений, не помнящая найденный текст.  
 [...] Соответствует любой одиночный символ в скобках  
 \* \*/* }  
}

План тестирования по задачи.

|  |  |
| --- | --- |
| Ввод MAС адреса | Вывод проверки МАС адреса |
| aE:dC:cA:56:76:54 | Введен правильный MAC адрес: aE:dC:cA:56:76:54 |
| 01:23:45:67:89:Az | Введен не правильный MAC адрес: 01:23:45:67:89:Az |
| 01:23:45:67:89:Az234sf234 | Введен не правильный MAC адрес: 01:23:45:67:89:Az234sf234 |
| aF:dB:eD:56:76:54 | Введен правильный MAC адрес: aF:dB:eD:56:76:54 |
| “” | Введена пустая строка |

Снимок экрана, отражающего работу программы по задаче.

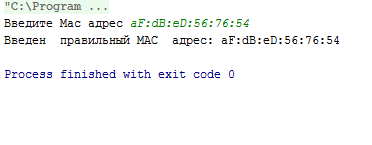


Рис. 1

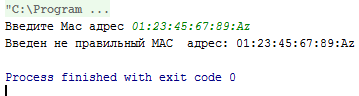


Рис. 2



Рис. 3

Вывод: Был получен опыт по написанию регулярных выражений на языке java.