|  |
| --- |
|  |
| МИНОБРНАУКИ РОССИИ |
| *Федеральное государственное бюджетное образовательное учреждение высшего образования* ***«МИРЭА – Российский технологический университет»***  **РТУ МИРЭА** |

**Институт информационных технологий (ИТ)**

**Кафедра инструментального и прикладного программного обеспечения (ИиППО)**

**Дисциплина «Программирование на языке Джава»**

**ОТЧЕТ**

**ПО ПРАКТИЧЕСКОМУ ЗАНЯТИЮ №12**

Выполнил студент группы ИНБО-02-20 Судариков Д.А.

Принял Степанов П.В.

Практические работы выполнены «\_\_\_»\_\_\_\_\_\_\_2021г.

«\_\_\_\_\_\_\_\_\_\_\_\_\_\_» «\_\_\_»\_\_\_\_\_\_\_2021г.

Отметка о выполнении

**Москва – 2021 г.**

## **Задание**

Закрепить знания в области обработки строк, научиться применять методы класса String и других классов для обработки строк.

## **Ход Работы**

В ходе выполнения работы были получены следующие исходные коды:

package ru.mirea.lab.lab12\_13\_14\_15.ex1;  
  
public class Person {  
 private String name, surname, patronymic;  
  
 public Person(String surname) {  
 this.surname = surname;  
 }  
  
 public Person(String name, String surname, String patronymic) {  
 this.name = name;  
 this.surname = surname;  
 this.patronymic = patronymic;  
 }  
 public String getFio(){  
 StringBuilder sb = new StringBuilder(surname);  
 if(name != null && ! name.equals(""))  
 sb.append(" ").append(name);  
 if(patronymic != null && ! patronymic.equals(""))  
 sb.append(" ").append(patronymic);  
 return sb.toString();  
 }  
  
 public static void main(String[] args) {  
 Person p1 = new Person("Простаков");  
 Person p2 = new Person("", "Суриков", "");  
 Person p3 = new Person("Николай", "Олегович", "Александрович");  
  
 System.*out*.println(p1.getFio());  
 System.*out*.println(p2.getFio());  
 System.*out*.println(p3.getFio());  
 }  
}

----

package ru.mirea.lab.lab12\_13\_14\_15.ex2;  
  
import java.util.StringTokenizer;  
  
 public class Add {  
 private String country, region, city, street, house, housing, flat;  
  
 public Add(String stringToParse, boolean Deluteli){  
 if(stringToParse == null) throw new NullPointerException();  
 String[] s;  
 if(Deluteli){  
 StringTokenizer st = new StringTokenizer(stringToParse, ",.;");  
 s = new String[st.countTokens()];  
 int i = 0;  
 while(st.hasMoreTokens()) s[i++] = st.nextToken();  
 }else{  
 s = stringToParse.split(",");  
 }  
  
 if(s.length < 7) throw new IllegalArgumentException("Строка содержит недостаточно данных!");  
 country = s[0].trim();  
 region = s[1].trim();  
 city = s[2].trim();  
 street = s[3].trim();  
 house = s[4].trim();  
 housing = s[5].trim();  
 flat = s[6].trim();  
 }  
  
 @Override  
 public String toString() {  
 return "Address{" +  
 "country='" + country + '\'' + ", region='" + region + '\'' + ", city='" + city + '\'' + ", street='" + street + '\'' +  
 ", house='" + house + '\'' + ", housing='" + housing + '\'' + ", flat='" + flat + '\'' + '}';  
 }  
  
 public static void main(String[] args) {  
 Add a1 = new Add("Россия, МО, Щёлково, Неделина, 25, 1, 76", false);  
 Add a2 = new Add("Россия, МО, Ногинск, Самодеятельная, 14; 1. 46", true);  
 Add a3 = new Add("UK, London, London, Trafalgar square, 1, 2, 3", false);  
  
 System.*out*.println(a1);  
 System.*out*.println(a2);  
 System.*out*.println(a3);  
  
 }  
 }

----

package ru.mirea.lab.lab12\_13\_14\_15.ex3;  
  
public class Shirts {  
 private String article, Name, color, size;  
  
 public Shirts(String stringToParse){  
 String[] s = stringToParse.split(",");  
 article = s[0];  
 Name = s[1];  
 color = s[2];  
 size = s[3];  
 }  
  
 @Override  
 public String toString() {  
 return "Shirt:\n" + "\tarticle='" + article + '\'' + "\n\tName='" + Name + '\'' + "\n\tcolor='" + color + '\'' + "\n\tsize='" + size + '\'';  
 }  
  
 public static void main(String[] args) {  
 String[] strs = new String[]{  
 "S001,Black Polo Shirt,Black,XL",  
 "S002,Black Polo Shirt,Black,L",  
 "S003,Blue Polo Shirt,Blue,XL",  
 "S004,Blue Polo Shirt,Blue,M",  
 "S005,Tan Polo Shirt,Tan,XL",  
 "S006,Black T-Shirt,Black,XL",  
 "S007,White T-Shirt,White,XL",  
 "S008,White T-Shirt,White,L",  
 "S009,Green T-Shirt,Green,S",  
 "S010,Orange T-Shirt,Orange,S",  
 "S011,Maroon Polo Shirt,Maroon,S"  
 };  
 Shirts[] shirts = new Shirts[strs.length];  
 for (int i = 0; i < strs.length; i++) {  
 shirts[i] = new Shirts(strs[i]);  
 System.*out*.println(shirts[i]);  
 }  
 }  
}

}

----

package ru.mirea.lab.lab12\_13\_14\_15.ex4;  
  
public class PhoneNumbers {  
 private String code, city, triple, quadruple;  
  
 public PhoneNumbers(String phone){  
 if(phone.startsWith("8")){  
 code = "+7";  
 }else{  
 code = phone.substring(0, phone.length()-10);  
 }  
 city = phone.substring(phone.length()-10, phone.length()-7);  
 triple = phone.substring(phone.length()-7, phone.length()-4);  
 quadruple = phone.substring(phone.length()-4);  
 }  
  
 @Override  
 public String toString() {  
 return code + city + "-" + triple + "-" + quadruple;  
 }  
  
 public static void main(String[] args) {  
 System.*out*.println(new PhoneNumbers("89998887766"));  
 System.*out*.println(new PhoneNumbers("+79998887766"));  
 System.*out*.println(new PhoneNumbers("+1234567891234"));  
 System.*out*.println(new PhoneNumbers("+181112223335"));  
 }  
}

----

package ru.mirea.lab.lab12\_13\_14\_15.ex5;  
import java.io.FileReader;  
import java.io.IOException;  
import java.util.\*;  
  
public class ConnectingWords {  
 public static void main(String[] args) {  
 Scanner sc = new Scanner(System.*in*);  
 String path = sc.nextLine();  
 StringBuilder s = new StringBuilder();  
 try(FileReader f = new FileReader(path)){  
 int c;  
 while((c=f.read())!=-1){  
 s.append((char) c);  
 }  
 }catch (IOException e){  
 e.printStackTrace();  
 }  
 System.*out*.println(*getLine*(s.toString()));  
 }  
  
 private static String getLine(String s) {  
 Queue<String> Add = new ArrayDeque<>(Arrays.*asList*(s.split(" ")));  
 Set<String> added = new HashSet<>();  
 added.add(Add.peek());  
 StringBuilder sb = new StringBuilder(Add.remove());  
 boolean wasAdded = true;  
  
 while(!Add.isEmpty() && wasAdded){  
 wasAdded = false;  
 int i = 0;  
 for (; i < Add.size(); i++) {  
 String current = Add.remove();  
 if(added.contains(current)) continue;  
 if(current.startsWith(sb.substring(sb.length()-1).toLowerCase()) ||  
 current.startsWith(sb.substring(sb.length()-1).toUpperCase())){  
 sb.append(" ").append(current );  
 wasAdded = true;  
 added.add(current );  
 break;  
 }  
 Add.add(current );  
 }  
 for(; i< Add.size(); i++){  
 String cur = Add.remove();  
 if(added.contains(cur)) continue;  
 Add.add(cur);  
 }  
 }  
 return sb.toString();  
 }  
}

## **Вывод**

В данной практической работе мной были получены знания по обработке строк.