|  |
| --- |
|  |
| 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 ПРАКТИЧЕСКИМ РАБОТАМ №12-15**

Выполнил студент группы ИНБО-02-20 Бойко К.В.

Принял Степанов П.В.

Практическая работа выполнена «\_\_\_»\_\_\_\_\_\_\_2021г.

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

Отметка о выполнении

**Москва – 2021 г.**

1. **Задание**

**Упражнение 1.**

Разработать класс Person, в котором имеется функция, возвращающая Фамилию И.О. Функция должна учитывать возможность отсутствия значений в полях Имя и Отчество.

Программу оптимизировать с точки зрения быстродействия.

**Упражнение 2.**

Доработать класс адреса, который из полученной строки формата “Страна[d] Регион[d] Город[d] Улица[d] Дом[d] Корпус[d] Квартира” ([d] – разделитель, например, «запятая») выбирает соответствующие части и записывает их в соответствующие поля класса Address. Учесть, что в начале и конце разобранной части адреса не должно быть пробелов. Все поля адреса строковые. Разработать проверочный класс не менее чем на четыре адресных строки. В программе предусмотреть две реализации этого метода:

а) разделитель – только запятая (использовать метод split()); Внимание, при разработке нужно учесть, что

б) разделитель – любой из символов ,.;- (класс StringTokenizer).

**Упражнение 3.**

Реализуйте класс Shirt: Метод toString() выводит объяснение и значение полей построчно. Дан также строковый массив: shirts[0] = «S001,Black Polo Shirt,Black,XL»; shirts[1] = «S002,Black Polo Shirt,Black,L»; shirts[2] = «S003,Blue Polo Shirt,Blue,XL»; shirts[3] = «S004,Blue Polo Shirt,Blue,M»; shirts[4] = «S005,Tan Polo Shirt,Tan,XL»; shirts[5] = «S006,Black T-Shirt,Black,XL»; shirts[6] = «S007,White T- Shirt,White,XL»; shirts[7] = «S008,White T-Shirt,White,L»; shirts[8] = «S009,Green T-Shirt,Green,S»; shirts[9] = «S010,Orange T-Shirt,Orange,S»; shirts[10] = «S011,Maroon Polo Shirt,Maroon,S»;

Преобразуйте строковый массив в массив класса Shirt и выведите его на консоль.

**Упражнение 4.**

Разработайте класс, который получает строковое представление телефонного номера в одном из двух возможных строковых форматов:

+«Код страны» «Номер 10 цифр», например “+79175655655”

или “+104289652211”

или 8«Номер 10 цифр» для России, например “89175655655”

и преобразует полученную строку в формат:

+«Код страны» «Три цифры»–«Три цифры»–«Четыре цифры»

**Упражнение 5.**

В методе main считай с консоли имя файла, который содержит слова, разделенные пробелом. В методе getLine() используя StringBuilder расставьте все слова в таком порядке, чтобы последняя буква данного слова совпадала с первой буквой следующего не учитывая регистр. Каждое слово должно участвовать 1 раз.

1. **Ход Работы**

(основной код доступен по ссылке <https://github.com/Zyablikitsme/Java_practice>)

В ходе выполнения работы были получены следующие исходные коды:

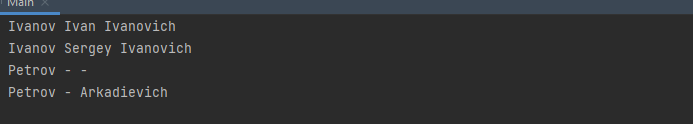
**Address.java**

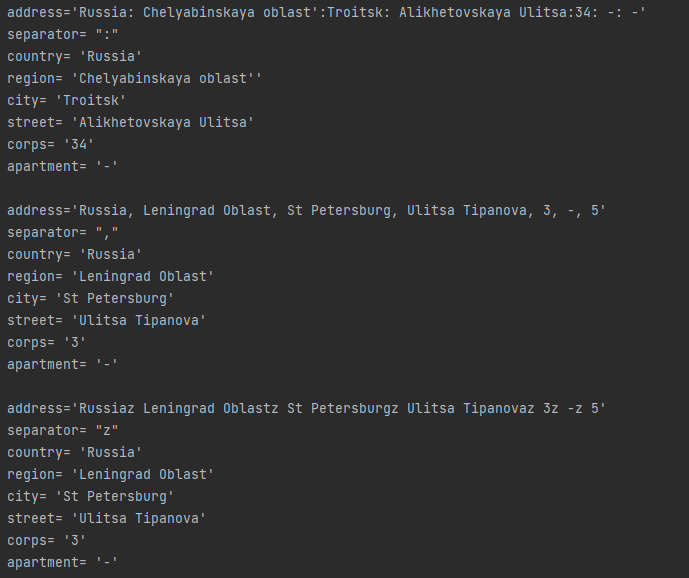
package ru.mirea.java.practice12\_15;  
  
import java.util.StringTokenizer;  
  
// Задание 2.  
  
// Доработать класс адреса, который из полученной строки формата  
// Страна[d] Регион[d] Город[d] Улица[d] Дом[d] Корпус[d] Квартира  
// ([d] – разделитель, например, «запятая») выбирает соответствующие  
// части и записывает их в соответствующие поля класса Address. Учесть, что в начале и конце разобранной  
// части адреса не должно быть пробелов. Все поля адреса строковые. Разработать проверочный класс не  
// менее чем на четыре адресных строки. В программе предусмотреть две реализации этого метода:  
//  
// а) разделитель – только запятая (использовать метод split()); Внимание, при разработке нужно учесть, что  
// б) разделитель – любой из символов ,.;- (класс StringTokenizer).  
  
  
public class Address {  
 String address, separator;  
  
 String country, region, city, street, corps, apartment;  
  
 public Address(String address, String separator) {  
 this.address = address;  
 this.separator = separator;  
  
 if(separator.equals(",")) separation1();  
 else separation2();  
 }  
  
 public void separation1(){  
 String[] separatedAddress = this.address.split(",");  
  
 for (int i = 0; i < 6; i++){  
 separatedAddress[i] = separatedAddress[i].replaceFirst("\\s", "");  
 switch(i){  
 case 0:  
 country = separatedAddress[i];  
 break;  
 case 1:  
 region = separatedAddress[i];  
 break;  
 case 2:  
 city = separatedAddress[i];  
 break;  
 case 3:  
 street = separatedAddress[i];  
 break;  
 case 4:  
 corps = separatedAddress[i];  
 break;  
 case 5:  
 apartment = separatedAddress[i];  
 break;  
 }  
 }  
 }  
  
 public void separation2(){  
 StringTokenizer separatedAddress = new StringTokenizer(address, separator);  
  
 for (int i = 0; i < 6; i++){  
 switch(i){  
 case 0:  
 country = separatedAddress.nextToken().replaceFirst("\\s", "");  
 break;  
 case 1:  
 region = separatedAddress.nextToken().replaceFirst("\\s", "");  
 break;  
 case 2:  
 city = separatedAddress.nextToken().replaceFirst("\\s", "");  
 break;  
 case 3:  
 street = separatedAddress.nextToken().replaceFirst("\\s", "");  
 break;  
 case 4:  
 corps = separatedAddress.nextToken().replaceFirst("\\s", "");  
 break;  
 case 5:  
 apartment = separatedAddress.nextToken().replaceFirst("\\s", "");  
 break;  
 }  
 }  
 }  
  
 @Override  
 public String toString() {  
 return  
 "\naddress='" + address + '\'' +  
 " \nseparator= \"" + separator + "\"" +  
 " \ncountry= '" + country + '\'' +  
 " \nregion= '" + region + '\'' +  
 " \ncity= '" + city + '\'' +  
 " \nstreet= '" + street + '\'' +  
 " \ncorps= '" + corps + '\'' +  
 " \napartment= '" + apartment + '\'';  
 }  
}

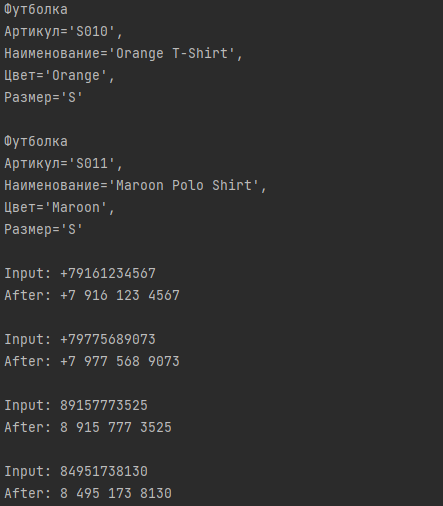
**PhoneNumber.java**

package ru.mirea.java.practice12\_15;  
  
// Задание 4.  
//Разработайте класс, который получает строковое представление телефонного номера в одном из  
//двух возможных строковых форматов:  
//+Код страны&gt;&lt;Номер 10 цифр&gt;, например “+79175655655” или  
//“+104289652211” или  
//8&lt;Номер 10 цифр&gt; для России, например “89175655655”  
//и преобразует полученную строку в формат:  
//+&lt;Код страны&gt;&lt;Три цифры&gt;–&lt;Три цифры&gt;–&lt;Четыре цифры&gt;  
  
  
public class PhoneNumber {  
 String number, readableNumber;  
  
 public PhoneNumber(String number) {  
 this.number = number;  
  
 if (number.charAt(0) == '+') readablePhoneNumberWithPlus();  
 else readablePhoneNumberWithEight();  
 }  
  
 public void readablePhoneNumberWithPlus(){  
 readableNumber = number.*format*("%.2s %.3s %.3s %.4s", number, number.substring(2), number.substring(5), number.substring(8));  
 }  
  
 public void readablePhoneNumberWithEight(){  
 readableNumber = number.*format*("%.1s %.3s %.3s %.4s", number, number.substring(1), number.substring(4), number.substring(7));  
 }  
  
 @Override  
 public String toString() {  
 return "\nInput: " + number  
 + "\nAfter: " + readableNumber;  
 }  
}

**Выходные данные программы** (при запуске Main.java)**:**







1. **Вывод**

В ходе данной работы были получены знания по работе со строками и их обработке в Java, были изучены методы класса String и других классов для обработки строк, были получены навыки по работе с файловым вводом/выводом в Java.