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

Отчет по выполнению практической работы 14

**Тема. ОБРАБОТКА СТРОК.**

Дисциплина Программирование на языке Джава

Выполнил

|  |  |
| --- | --- |
| студент | Болотов Михаил |
|  | Фамилия И.О. |
| группа | ИКБО-06-19 |
|  | Номер группы |

Москва 2020

**Содержание**

[​ Теория 2](#__RefHeading___Toc4128_811265792)

[​ Задание 2](#__RefHeading___Toc4136_811265792)

[​ Код 3](#__RefHeading___Toc35859_811265792)

[​ Main.java 3](#__RefHeading___Toc42736_811265792)

[​ Address.java 3](#__RefHeading___Toc42738_811265792)

[​ Скриншот 5](#__RefHeading___Toc3748_811265792)

[​ Заключение 5](#__RefHeading___Toc3750_811265792)

[​ Библиографический список 5](#__RefHeading___Toc3752_811265792)

# Теория

Для работы с текстовыми данными в Java есть три класса: String, StringBuffer и StringBuilder.

Особенности использования строк в Java

В Java строки представляют собой неизменяемую последовательность символов Unicode. В отличие от представления в C / C ++, где строка является просто массивом типа char, любая Java, строка является объектом класса java.lang.

Однако Java строка, представляет собой в отличие от других используемых классов особый класс, который обладает довольно специфичными характеристиками. Отличия класса строк от обычных классов:

• строка в Java представляет из себя строку литералов (текст), помещенных в двойные кавычки, например:

"Hello , World! ". Вы можете присвоить последовательность строковых литералов непосредственно переменной типа String, вместо того чтобы вызывать конструктор для создания экземпляра класса String.

• Оператор '+' является перегруженным, для объектов типа String, и всегда используется, чтобы объединить две строки операндов. В данном контексте мы говорим об операции конкатенации или сложения строк. Хотя '+' не работает как оператор сложения для любых других объектов, кроме строк, например, таких как Point и Circle.

• Строка является неизменяемой, то есть, символьной константой. Это значит, что ее содержание не может быть изменено после ее (строки как объекта) создания. Например, метод toUpperCase () – преобразования к верхнему регистру создает и возвращает новую строку вместо изменения содержания существующей строки.

Обратитесь к API JDK для того чтобы ознакомиться с полным списком возможностей класса String в java.lang.String.

# Задание

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).

# Код

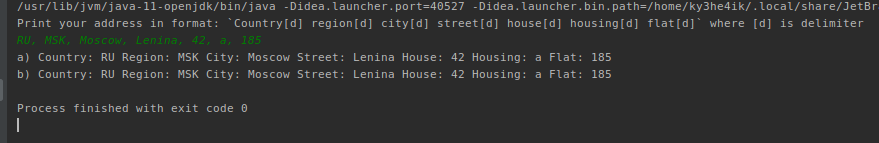
### Main.java

package dev.ky3he4ik.pr.pr14;  
  
import java.util.Scanner;  
  
public class Main {  
 public static void main(String[] args) {  
 System.out.println("Print your address in format: `Country[d] region[d] city[d] street[d] house[d] housing[d] flat[d]` where [d] is delimiter");  
 Scanner scanner = new Scanner(System.in);  
 String address = scanner.nextLine();  
 Address addressA = Address.fromStringA(address);  
 if (addressA == null)  
 System.out.println("a) Wrong address (the only supported delimiter is `,`");  
 else  
 System.out.println("a) " + addressA);  
 Address addressB = Address.fromStringB(address);  
 if (addressB == null)  
 System.out.println("b) Wrong address (the only supported delimiters is one of `,.;-`");  
 else  
 System.out.println("b) " + addressB);  
 }  
}

### Address.java

package dev.ky3he4ik.pr.pr14;  
  
import java.util.StringTokenizer;  
  
public class Address {  
 String country;  
 String region;  
 String city;  
 String street;  
 String house;  
 String housing;  
 String flat;  
  
 public Address(String country, String region, String city, String street, String house, String housing, String flat) {  
 this.country = country;  
 this.region = region;  
 this.city = city;  
 this.street = street;  
 this.house = house;  
 this.housing = housing;  
 this.flat = flat;  
 }  
  
 @Override  
 public String toString() {  
 return "Country: " + country +  
 " Region: " + region +  
 " City: " + city +  
 " Street: " + street +  
 " House: " + house +  
 " Housing: " + housing +  
 " Flat: " + flat;  
 }  
  
 public static Address fromStringA(String address) {  
 String[] tokens = address.split(", ");  
 if (tokens.length == 7)  
 return new Address(tokens[0], tokens[1], tokens[2], tokens[3], tokens[4], tokens[5], tokens[6]);  
 return null;  
 }  
  
 public static Address fromStringB(String address) {  
 StringTokenizer tokenizer = new StringTokenizer(address, ",.;-");  
  
 if (tokenizer.countTokens() == 7)  
 return new Address(tokenizer.nextToken().trim(), tokenizer.nextToken().trim(), tokenizer.nextToken().trim(),  
 tokenizer.nextToken().trim(), tokenizer.nextToken().trim(), tokenizer.nextToken().trim(), tokenizer.nextToken().trim());  
 return null;  
 }  
  
  
}

# Скриншот



# Заключение

В данной практической работе я закрепил знания в области обработки строк, научился применять методы класса String и других классов для обработки строк.

# Библиографический список

1. Зорина Н.В. Курс лекций по Объектно-ориентированному программированию на Java, МИРЭА, Москва, 2016
2. Программирование на языке Java: работа со строками и массивами. Методические указания. [Электронный ресурс] : Учебно-методические пособия — Электрон. дан. — СПб. : ПГУПС, 2015. — 24 с.
3. Кожомбердиева, Г.И. Программирование на языке Java: создание графического интерфейса пользователя: учеб. пособие. [Электронный ресурс] : Учебные пособия / Г.И. Кожомбердиева, М.И. Гарина. — Электрон. дан. — СПб.: ПГУПС, 2012. — 67 с.
4. Вишневская, Т.И. Технология программирования. Часть 1. [Электронный ресурс] / Т.И. Вишневская, Т.Н. Романова. — Электрон. дан. — М. : МГТУ им. Н.Э. Баумана, 2007. — 59 с.