



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 №12-15**

Выполнил студент группы ИНБО-02-20

Деревянкин Н.А.

Принял

Степанов П.В.

Практическая работа выполнена «\_\_»\_\_\_\_\_ 2021 г.

«\_\_\_\_\_»

«\_\_»\_\_\_\_\_ 2021 г.

Отметка о выполнении

**Москва – 2021 г.**

## СОДЕРЖАНИЕ

Цель работы .....	3
Задание .....	3
Репозиторий .....	4
Выполнение работы .....	4
Код выполненной работы.....	4
Вывод.....	11

## Цель работы

Цель данной лабораторной работы – закрепить знания в области обработки строк, научиться применять методы класса String и других классов для обработки строк.

## Задание

### Задание 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. **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).

### Задание 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.

### Задание 4

Разработайте класс, который получает строковое представление телефонного номера в одном из двух возможных строковых форматах: +<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>

## Репозиторий

Ссылка: [https://github.com/neluckoff/mirea-javalessons/tree/master/src/ru/luckoff/mirea/practice\\_12](https://github.com/neluckoff/mirea-javalessons/tree/master/src/ru/luckoff/mirea/practice_12)

## Выполнение работы

### Решение задания 1

Задание достаточно легкое. Я реализовал класс `Person`, создал в нем два конструктора, один с полным ФИО, другой только с фамилией. В `toString()` я реализовал проверку наличия Имени и Отчества в объекте, в случае, если отсутствуют, то вывожу только Фамилию, иначе полное ФИО.

### Решение задания 2

Мной был реализован пустой конструктор и метод `addAddress`, который передает всю полученную строку в определенные переменные, характерные им. В первом подпункте этого задания был реализован ввод адреса только через запятую, во втором через любые знаки препинания.

### Решение задания 3

Здесь я реализовал класс `Shirt`. Опять же таки создал пустой конструктор и метод `addShirt()`, который работает также, как и во втором задании.

### Решение задания 4

Первым делом я создал переменную `formattedNumber`, в которой буду хранить отформатированный номер. Мной был создан конструктор класса, в котором я как раз и форматирую номер, который нам дают. И, конечно же, метод, возвращающий этот отформатированный номер.

## Код выполненной работы

Здесь в нескольких скриншотах можно увидеть как выглядит код выполненного задания и результат его работы.

### Полученный код для задания №1

```

public class Person {
    private String firstName;
    private final String lastName;
    private String patronymic;

    public Person(String lastName) {
        this.lastName = lastName;
    }

    public Person(String firstName, String lastName, String patronymic) {
        this.firstName = firstName;
        this.lastName = lastName;
        this.patronymic = patronymic;
    }

    public String toString() {
        if (firstName != null || patronymic != null) {
            return lastName + " " + firstName + " " + patronymic;
        } else {
            return lastName;
        }
    }

    public static void main(String[] args) {
        Person first_person = new Person( firstName: "Лукьяненко", lastName: "Дмитрий", patronymic: "Васильевич");
        System.out.println(first_person);
        Person second_person = new Person( lastName: "Лукьяненко");
        System.out.println(second_person);
    }
}

```

Рисунок 1 – Класс Person

```

Дмитрий Лукьяненко Васильевич
Лукьяненко

Process finished with exit code 0

```

Рисунок 2 – Результат запуска первой задачи

## Полученный код для задания №1

```

public class Address {
    private String country, region, city, street, house, block, flat;

    public Address() {
    }

    public void addAddress(String data, int i) {
        switch (i) {
            case 0 -> country = data;
            case 1 -> region = data;
            case 2 -> city = data;
            case 3 -> street = data;
            case 4 -> house = data;
            case 5 -> block = data;
            case 6 -> flat = data;
        }
    }

    public String toString() {
        return ("Адресс: " + country + " " + region + " - г. " + city + " ул. " +
            street + " д. " + house + " к. " + block + " кв. " + flat);
    }
}

```

Рисунок 3 – Класс Address

```

public static void main(String[] args) {
    int i = 0, j = 0;
    Address address = new Address();
    System.out.println("Первое задание:");
    String str1 = "Россия, 50, Долгопрудный, Спортивная, 11, Б, 235";
    System.out.println(str1);
    for (String data : str1.replace( target: " ", replacement: "").split( regex: ",")) {
        address.addAddress(data, i);
        i++;
    }
    System.out.println(address + "\n\n");

    System.out.println("Второе задание:");
    String str2 = "Россия, 50. Долгопрудный; Спортивная - 11: Б, 235";
    System.out.println(str2);
    str2 = str2.replace( target: " ", replacement: "");
    StringTokenizer st = new StringTokenizer(str2, " . ; - :");
    while (st.hasMoreTokens()) {
        address.addAddress(st.nextToken(), j);
        j++;
    }
    System.out.println(address);
}

```

Рисунок 4 – Ввод для класса Address

```

Первое задание:
Россия, 50, Долгопрудный, Спортивная, 11, Б, 235
Адрес: Россия 50 - г. Долгопрудный ул. Спортивная д. 11 к. Б кв. 235

Второе задание:
Россия, 50. Долгопрудный; Спортивная - 11: Б, 235
Адрес: Россия 50 - г. Долгопрудный ул. Спортивная д. 11 к. Б кв. 235

```

Рисунок 5 – Результат запуска второй задачи

### Полученный код для задания №3

```
public class Shirt {  
    private String num, name, color, size;  
  
    public Shirt() {  
    }  
  
    public void addShirt(String shirt, int i) {  
        switch (i) {  
            case 0 -> num = shirt;  
            case 1 -> name = shirt;  
            case 2 -> color = shirt;  
            case 3 -> size = shirt;  
        }  
    }  
  
    public String toString() {  
        return (num + " | " + name + " | " + color + " | " + size);  
    }  
}
```

Рисунок 6 – Класс Shirt



```

public static void main(String[] args) {
    Shirt str = new Shirt();

    String[] shirts = new String[11];
    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";

    for (int i = 0; i < 11; i++) {
        String[] spl = shirts[i].split(" ");
        for (int j = 0; j < spl.length; j++) {
            str.addShirt(spl[j], j);
        }
        System.out.println(str);
    }
}

```

Рисунок 7 – Ввод для класса Shirt

```

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

Process finished with exit code 0

```

Рисунок 8 – Результат запуска третьей задачи

### Полученный код для задания №3

```
public class PhoneNumber {
    private final String formattedNumber;

    PhoneNumber(String number) {
        StringBuilder stringBuilder = new StringBuilder(number);
        String code = stringBuilder.substring(0, stringBuilder.length() - 10);
        if (code.equals("8"))
            code = "+7";
        String firstDigit = stringBuilder.substring(stringBuilder.length() - 10, stringBuilder.length() - 7);
        String secondDigit = stringBuilder.substring(stringBuilder.length() - 7, stringBuilder.length() - 4);
        String thirdDigit = stringBuilder.substring(stringBuilder.length() - 4, stringBuilder.length());
        this.formattedNumber = code + " (" + firstDigit + ") " + secondDigit + "-" + thirdDigit;
    }

    public String getFormattedNumber() {
        return this.formattedNumber;
    }
}
```

Рисунок 9 – Класс PhoneNumber

```
public static void main(String[] args) {
    PhoneNumber phoneNumberFirst = new PhoneNumber("+79266715863");
    PhoneNumber phoneNumberSecond = new PhoneNumber("89266715859");
    System.out.println(phoneNumberFirst.getFormattedNumber());
    System.out.println(phoneNumberSecond.getFormattedNumber());
}
```

Рисунок 10 – Ввод для класса PhoneNumber

```
+7 (926) 671-5863
+7 (926) 671-5859

Process finished with exit code 0
```

Рисунок 11 – Результат запуска четвертой задачи

## **Вывод**

В результате выполнения данной практической работы я смог закрепить знания в области обработки строк, научиться применять методы класса String и других классов для обработки строк на языке программирования Java.

**GITHUB - <https://github.com/dronikosha/JavaPractice>**