

**Лабораторная работа №3. Обработка строк**

**Цели работы**

1. Освоить работу с классами String и StringBuilder
2. Освоить работу с составным оператором

**Используемое программное обеспечение**

1. JDK 1.5 и выше
2. Интегрированная среда разработки на Java

**Справка**

Произвести над строкой следующие операции:

1. Удалить первое слово (вместе с подчёркиванием)
2. Удалить символ с индексом 2
3. Заменить все буквы “т” на “о”
4. Поменять местами 4 и 2 слово
5. Инвертировать последнее слово

Программа должна считать из аргументов командной строки предложение, в котором слова разделены нижним подчёркиванием, преобразовать его по заданию и вывести результат в System.out

Класс String предназначен для хранения неизменяемых (immutable)  
строк  
Полезные методы этого класса:

* split(String regex) – разбивает строку на массив строк,  
  разделителем которых в исходной строке являлся переданный  
  аргумент
* replace(char oldChar, char newChar) – возвращает новую строку, в  
  которой все вхождения символа oldChar заменяются на символ  
  newChar

Класс StringBuilder предназначен для работы со строкой как с динамическим массивом символов  
Полезные методы этого класса:

* deleteCharAt(int index) – вырезает символ с индексом index
* append(String s) – дописывает в конец буфера строку s
* reverse() – заменяет последовательность символов на ту же  
  последовательность, идущую в обратном порядке
* toString() – преобразует содержимое буфера в объект класса String

**Пример выполнения лабораторной работы**

public class *Main* {

public static void main(String[] args) throws IOException {

// Исходная и преобразованая строка

String sourceText, processedText;

sourceText = args[0];

// Удаляем первое слово (цикл идёт с 1)

String words[] = sourceText.split("\_");

StringBuilder builder = new StringBuilder();

int i;

for (i = 1; i < words.length - 1; i++) {

builder.append(words[i]).append("\_");

}

builder.append(words[i]);

// Удаляем символ с индексом 2

builder.deleteCharAt(2);

// Заменяем все буквы 'т' на 'о'

// Это проще сделать, преобразовав StringBuilder в String

processedText = builder.toString().replace('т', 'о');

// Снова разбиваем на массив слов и меняем местами 4 и 2 слово

words = processedText.split("\_");

String cachedWord = words[3];

words[3] = words[1];

words[1] = cachedWord;

// Инвертируем последнее слово

// Это проще сделать, засунув его в StringBuilder

builder = new StringBuilder(words[words.length - 1]);

builder.reverse();

words[words.length - 1] = builder.toString();

// Собираем строку обратно

builder = new StringBuilder();

for (i = 0; i < words.length - 1; i++) {

builder.append(words[i]).append("\_");

}

builder.append(words[i]);

// Выводим результат на экран

System.out.println(builder.toString());

}

}

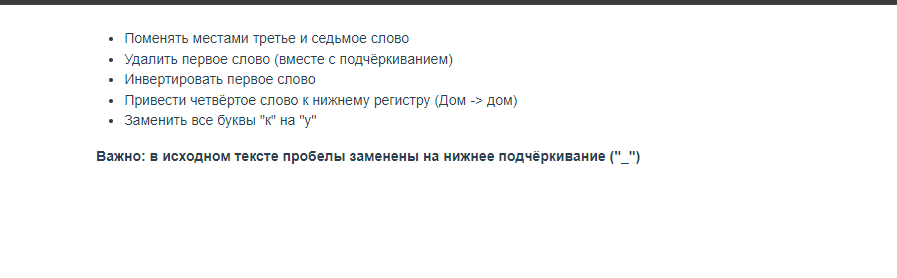
**Задание лабораторной работы**

1. Написать программу на Java, соответствующую заданию
2. Проверить правильность работы программы
3. Собрать исполняемый jar и отправить на проверку

**Требования к отчёту по лабораторной работе**

Отчёт должен содержать:

1. Титульный лист
2. Вариант задания
3. Листинг программы
4. Вывод



* Поменять местами третье и седьмое слово
* Удалить первое слово (вместе с подчёркиванием)
* Инвертировать первое слово
* Привести четвёртое слово к нижнему регистру (Дом -> дом)
* Заменить все буквы "к" на "у"

**Важно: в исходном тексте пробелы заменены на нижнее подчёркивание ("\_")**

**Дополнительная задача**

* Поменять местами третье и седьмое слово
* Удалить первое слово (вместе с подчёркиванием)
* Инвертировать первое слово
* Привести четвёртое слово к нижнему регистру (Дом -> дом)
* Заменить все буквы "к" на "у"
* Привести нечётные слова к нижнему регистру (Дом -> дом)
* Повернуть первое слово циклически на 2 символов вправо (Пример поворота на 2 символа: стена => насте)

**Важно: в исходном тексте пробелы заменены на нижнее подчёркивание ("\_")**