Основи технологій програмування

Лабораторна робота № 3

Вінницький В’ячеслав Андрійович

ІП-64, 2-ий курс

Кафедра обчислювальної техніки

ІП-6402

Варіант

С3 = 6402%3 = 0 => Тип StringBuilder

C17 = 6402%17 = 10 => З кожного речення заданого тексту видалити підрядок найбільшої довєини що починається та закінчуєтьсяи заданими літерами

Текст програмного коду

import java.util.Scanner;  
import java.util.regex.Matcher;  
import java.util.regex.Pattern;  
  
public class Main {  
 public static void main(String[] args) {  
 StringBuilder text = new StringBuilder(  
 "Inhabiting discretion the her dispatched decisively boisterous joy."+  
 " So form were wish open is able of mile of." +  
 " Waiting express if prevent it we an musical." +  
 " Especially reasonable travelling she son." +  
 " Resources resembled forfeited no to zealously." +  
 " Has procured daughter how friendly followed repeated who surprise." +  
 " Great asked oh under on voice downs." +  
 " Law together prospect kindness securing six." +  
 " Learning why get hastened smallest cheerful.");  
  
 System.*out*.println("String : " + text);//вывод текста  
  
 int dots\_count = 0;  
 for(int j = 0; j< text.length();j++ ) {  
 if(text.charAt(j) == '.')  
 dots\_count++;  
 }  
 System.*out*.println("Number of dots : " + dots\_count);  
  
 //Разделение на строки  
 Pattern pattern = Pattern.*compile*("\\."); // Best static final.  
 Matcher m = pattern.matcher(text);  
 int pos0 = 0;  
 int i = 0;  
  
 //создаем массив предложений  
 StringBuilder[] sentences = new StringBuilder[dots\_count];  
 CharSequence sequence;  
 //разделяем на предложения  
 while (m.find()) {  
 int pos1 = m.start();  
 //этот if убирает проблему удаления первой буквы  
 if(i == 0) {  
 sequence = text.subSequence(pos0, pos1);  
 }  
 else {  
 sequence = text.subSequence(pos0 + 1, pos1);  
 }  
 pos0 = pos1;//двигаемся от точки к точке  
 sentences[i] = new StringBuilder(String.*valueOf*(sequence));  
 i++;  
 }  
  
 //вводим букву  
 Scanner input = new Scanner(System.*in*);  
 for(int k = 0; k < dots\_count; k++) {  
 System.*out*.println(sentences[k]);  
 String symbol = input.nextLine().toLowerCase();  
 /\*для того, чтобы найти самую длинную подстроку достаточно искать первый символ слева и первый справа  
 можно также использовать методы indexOf(), lastIndexOf() - я использую их для изначальной инициализации  
 \*/  
 int del\_start = sentences[k].toString().toLowerCase().indexOf(symbol.toString());  
 int del\_end = sentences[k].toString().toLowerCase().lastIndexOf(symbol.toString());  
  
 for(int j = 0; j < dots\_count; j++) {  
 for(int l = 0; l < sentences[j].length(); l++) {  
 if(sentences[j].charAt(l) == symbol.toCharArray()[0]) {  
 del\_start = l;  
 }  
 break;  
 }  
 for(int n = sentences[j].length() - 1; n <= 0; n--) {  
 if(sentences[j].charAt(n) == symbol.toCharArray()[0]) {  
 del\_end = n;  
 break;  
 }  
 }  
 }  
 System.*out*.println("first : " + del\_start + " last : " + del\_end);  
  
 try {  
 sentences[k].delete(del\_start , del\_end + 1);  
 }  
 catch (StringIndexOutOfBoundsException e) {  
 System.*out*.println("Error : " + e);  
 }  
 System.*out*.println( "Предложение с вырезаной подстрокой : " + sentences[k]);  
 }  
 }  
}

Висновки

Як я зрозумів завдання : У нас є заданий текст, що складається з реченнь. При проходженні кожного речення ми задаємо літеру, і программа видаляє найдовший підрядок, що починається та закінчується заданою літерою.

Проблеми з якими я зіткнувся :

1)У мене був намір створити масив усіх можливих підрядків для кожного речення перебором масиву слів. Втім, оскільки всі елементи мають бути StringBuilder, зробити це не вийшно, бо append у пустий StringBuilder не працюють, а при додаванні сточатку деякого значення =>далі append потрібних слів у випадкових місцях цього StringBuilder виникали записи «null».

2)Оскільки використовувати треба StringBuilder, то я так зрозумів, що метод .toString() застосовуватися не повинен. Саме тому було прийнято рішення розділяти на речення не способом .toString().split(\\.), а за допомогою класів Pattern та Matcher.

3)Якщо ввести літеру, якої немає у реченні то буде помилка типу StringIndexOutOfBoundsException. Вирішив я це конструкцією try catch.

Загалом вважаю, що для того завдання, що було поставлено у ЛР доцільно було би використовувати String, бо StringBuilder використовується, як я зрозумів, для більш специфічних задач(де важлива швидкість, оптимізація пам’яті та однопоточність).