МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

**ОДЕСЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНОЛОГІЧНИЙ УНІВЕРСИТЕТ**

**ЛАБОРАТОРНА РОБОТА №1**

З дисципліни:

«Системний аналіз та проєктування комп’ютерних інформаційних систем»

Виконав студент I (скороченого) курсу

групи КН-333а

Смотрич Артем Жаннович

Одеса, 2022

**Задача:**

Існує набір рядків (рядок - це масив зі слів), умова форматування цього рядка та обмеження символів в рядку. Необхідно підготувати текст до публікації, так щоб він задовільняв умовам форматування й не перевищував кількості символів в рядку. Текст має бути розміщеним між символами \* з обох сторін.

**Хід роботи**

**Аналіз задачі та алгоритм її рішення.**

**Аналіз задачі**

На основі умови задачі було встановлено, що найкращім шляхом рішення буде створення рекурсивної функції, у якій буде проводитись форматування. Кожний рядок буде проходити через рекурсивну функцію і оброблятися. Якщо рядок перевищує, введені користувачем, обмеження, то слова, які проходять по кількістю символів – виводяться, а різниця між обмеженням і довжиною рядка стає кількістю пробілів, які додаються з певної сторони (залежить від обраного форматування) до рядка, що виводиться.

***Алгоритм рішення:***

1. Користувач задає обмеження, форматування та три рядки, всі ці змінні перевіряються на правильність введення. Для перевірки чи відповідає кожне слово у рядках обмеженням, викликається функція.
2. Якщо всі змінні пройшли перевірку, тоді зберігається та виводиться рядок, який складається з зірочок, кількість зірочок встановлюється обмеженням до якого додають ще дві одиниці.
3. Викликається рекурсивна функція, яка приймає у якості параметрів рядок, обмеження по символам та форматування.
4. У рекурсивній функції рядок розбивається по словам та через лічільник встановлюється чи відповідає набір слів обмеженням. Якщо набір слів у сумі дає значення більше за обмеженням, тоді слово або набір слів записуються у зміну за винятком того слова, через яке трапився перебір у кількості символів. Всі інші слова записуються у другу змінну.
5. Рядок, у які були записані слова які не перевищують обмеження, форматуються шляхом додавання певної кількості пробілів, яка пробілів встановлюється різницею між обмеженням та довжиною рядка. Місце додавання пробілів встановлюється умовою, яка перевіряє тип форматування.
6. Виводиться відформатований рядок на початок и на кінець якого, додаються по зірочці.
7. Викликається рекурсивна функція, яка приймає у якості параметрів рядок без слова або набору слів, які були відформатовані та виведені. Далі проходять ті ж самі операції, які були встановлені вище.
8. Рекурсивна функціє працює до тих пір поки обмеження не буде перевищувати розмір рядку.
9. Виводиться збережений рядок із зірочок.

**Обґрунтування мови програмування для вирішення задачі.**

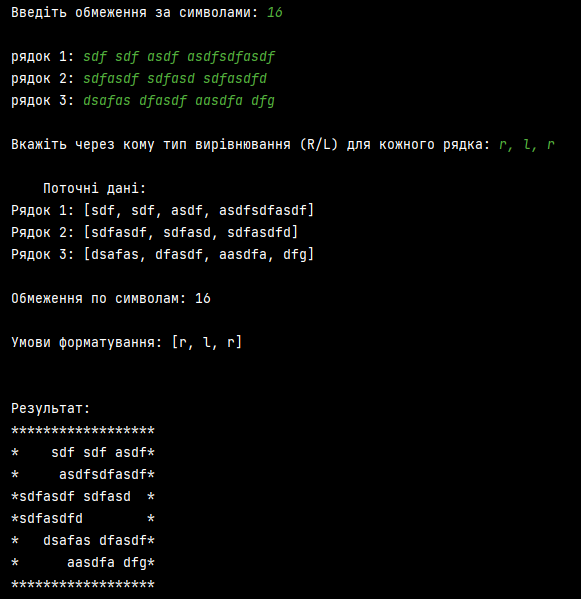
Вирішення поставленої задачі не потребує певної мови програмування, тому мій вибір впав на Java, як найбільш відому для мене мову, яка, крім цього, є одною з найпопулярніших мов програмування.

**Результати**

Лістинг програми:

public class Main {  
 public static void main(String[] args) throws IOException{  
 BufferedReader reader = new BufferedReader(new InputStreamReader(System.*in*));  
 boolean checkLength = false,  
 numCheck = false,  
 check = false;  
 int numQuant = 0;  
 String[] alignArr;  
 StringBuilder star = new StringBuilder();  
  
 do {  
 System.*out*.print("Введіть обмеження за символами: ");  
 try{  
 numQuant = Integer.*parseInt*(reader.readLine());  
 numCheck = true;  
 }catch (NumberFormatException e){  
 System.*out*.println("Число задано не правильно!");  
 }  
 }while (!numCheck);  
  
 String str1Temp = "",  
 str2Temp= "",  
 str3Temp= "";  
 System.*out*.println();  
  
 do {  
 str1Temp = ""; str2Temp = ""; str3Temp = "";  
  
 System.*out*.print("рядок 1: ");  
 str1Temp = reader.readLine();  
 if(!*checkStr*(str1Temp, numQuant)){  
 continue;  
 }  
  
 System.*out*.print("рядок 2: ");  
 str2Temp = reader.readLine();  
 if(!*checkStr*(str2Temp, numQuant)){  
 continue;  
 }  
  
 System.*out*.print("рядок 3: ");  
 str3Temp = reader.readLine();  
 if(*checkStr*(str3Temp, numQuant)){  
 checkLength = true;  
 }  
 }while (!checkLength);  
  
 do {  
 System.*out*.print("\nВкажіть через кому тип вирівнювання (R/L) для кожного рядка: ");  
 alignArr = reader.readLine().trim().replace(" ", "").toLowerCase().split(",");  
 if (alignArr.length == 3){  
 for (String s : alignArr) {  
 if (s.equals("r") || s.equals("l")) {  
 check = false;  
 }else {  
 System.*out*.println("Неправильно введено сторону вирівнювання!");  
 check = true;  
 break;  
 }  
 }  
 }else {  
 System.*out*.println("Кількість елементів менше аба більше трьох!");  
 check = true;  
 }  
 }while (check);  
  
 System.*out*.println("\n\tПоточні дані:" +  
 "\nРядок 1: " + Arrays.*toString*(str1Temp.trim().split(" ")) +  
 "\nРядок 2: " + Arrays.*toString*(str2Temp.trim().split(" ")) +  
 "\nРядок 3: " + Arrays.*toString*(str3Temp.trim().split(" ")) +  
 "\n\nОбмеження по символам: " + numQuant +  
 "\n\nУмови форматування: " + Arrays.*toString*(alignArr));  
  
  
 star.append("\*".repeat(Math.*max*(0, numQuant + 2)));  
 System.*out*.println("\n\nРезультат:\n" + star);  
 *getStr*(str1Temp, numQuant, alignArr[0]);  
 *getStr*(str2Temp, numQuant, alignArr[1]);  
 *getStr*(str3Temp, numQuant, alignArr[2]);  
 System.*out*.println(star);  
 }  
  
 public static boolean checkStr(String str, int num){  
 for (int i = 0; i < str.trim().split(" ").length; i++) {  
 if (str.trim().split(" ")[i].length() > num) {  
 System.*out*.println("Введене слово перевищує вказане обмеження!");  
 return false;  
 }  
 }  
 return true;  
 }  
  
 public static void getStr(String str, int num, String align ){  
 StringBuilder newStr = new StringBuilder();  
 if(str.length() > num){  
  
 String[] splitArr = str.split(" ");  
 int temp = 0, index = 0;  
  
 for (int i = 0; i < splitArr.length; i++) {  
 temp += splitArr[i].length();  
 if (temp >= num && splitArr.length != 1) {  
 index = i;  
 break;  
 } else if (temp <= num && splitArr.length == 1){  
 break;  
 }  
 temp++;  
 }  
 StringBuilder tempStr = new StringBuilder();  
 for(int i = index; i < splitArr.length; i++){  
 tempStr.append(" ").append(splitArr[i]);  
 }  
 newStr = new StringBuilder();  
 for(int i = 0; i <= index-1; i++){  
 newStr.append(" ").append(splitArr[i]);  
 }  
 newStr = new StringBuilder(newStr.toString().trim());  
 for (int y = 0; y < num - newStr.toString().trim().length(); y++) {  
 if(align.equals("l")) {  
 newStr.append(" ");  
 }  
 else if (align.equals("r")){  
 newStr = new StringBuilder(new StringBuilder(newStr.toString()).insert(0, " ").toString());  
 }  
 }  
 System.*out*.println(new StringBuilder(newStr.toString()).insert(0, "\*").toString() + "\*");  
 *getStr*(tempStr.toString().trim(), num, align);  
  
 }else{  
 newStr = new StringBuilder(str);  
 for (int y = 0; y < num - newStr.toString().trim().length(); y++) {  
 if(align.equals("l")) {  
 newStr.append(" ");  
 }  
 else if (align.equals("r")){  
 newStr = new StringBuilder(new StringBuilder(newStr.toString()).insert(0, " ").toString());  
 }  
 }  
 System.*out*.println(new StringBuilder(newStr.toString()).insert(0, "\*").toString() +"\*");  
 }  
 }  
}

Результат виконання програми:

**