**Машинное распознавание паттернов**

Олег получил инвестиции на стартап по машинному обучению, и занимается распознаванием закономерностей в тексте.

Его первый продукт будет анализировать текст, преобразованный из изображений точек и звёздочек, которые рисуют первоклассники в тетрадках. Последовательности всегда составлены по общему шаблону, но первоклассники пока часто ошибаются, и забывают поставить точку, рисуют лишнюю звёздочку, и т. п.

На вход программы поступают строки, состоящие из символов "." и "\*", всегда начинающиеся и завершающиеся звёздочкой. В них всегда повторяется единый шаблон, например:

\*..\*..\*..\*..\*..\*..\*

Такой пример считается корректным.

Однако первоклассники иногда ошибаются, и могут написать такие ошибочные строки:

\*..\*...\*..\*..\*..\*..\*

\*..\*..\*..\*..\*..\*\*..\*

Ещё примеры корректных строк:

\*

\*\*\*

\*.......\*.......\*

\*\*

\*.\*

Функция

bool LineAnalysis(string line)

получает на вход строку для анализа и возвращает логическое true/false, обозначающее корректность строки.

[как постить решение](https://skillsmart.ru/algo/lvl1/how.html)

навскидку нам надо определить длину подстроки между двумя \*

если они все совпадают – true

import java.util.\*;  
  
public class Patterns {  
  
 static boolean LineAnalysis(String line) {  
 boolean result = false;  
 String [] words = line.split("\\\*");  
  
 // if no dots present  
 if (!(line.contains("."))){  
 result = true;  
 }  
 // if string is "\*.\*"  
 if (line.equals("\*.\*")){  
 result = true;  
 }  
  
 for (int i = 0; i < words.length; i++){  
 System.*out*.println(words[i]);  
 if (!(words[i]).equals(words[1])){  
 System.*out*.println(words[i] + " doesn't match " + words[1]);  
 result = false;  
 break;  
 }  
 }  
  
 return result;  
 }  
  
  
 public static void main(String[] args) {  
 String test1 = "\*...\*....\*...\*...\*";  
  
 int firstDot = 0;  
 int marker = 0;  
 boolean match = true;  
  
 firstDot = test1.indexOf(".");  
 marker = test1.indexOf("\*", firstDot);  
  
 System.*out*.println("first dot at " + firstDot);  
 System.*out*.println("next star at " + marker);  
  
 String pattern = test1.substring(0, marker);  
  
 System.*out*.println("pattern: " + pattern + ", length: " + pattern.length());  
  
 StringBuilder sb1 = new StringBuilder(test1);  
 sb1.delete(0, marker);  
 System.*out*.println(sb1);  
  
 for (int i = 0; i < sb1.length() - 1; i++){  
  
  
 if (sb1.indexOf(pattern) == 0){  
 System.*out*.println("match");  
 sb1.delete(0, marker);  
 System.*out*.println("remaining string: " + sb1 + ", length: " + sb1.length());  
  
 if (sb1.length() > 1 && sb1.length() < pattern.length()){  
 System.*out*.println("remaining string shorter than pattern, no match");  
 match = false;  
 break;  
  
 }  
 }  
  
  
 else {  
 System.*out*.println("no match");  
 match = false;  
 break;  
  
 }  
 }  
  
 System.*out*.println("match = " + match);  
  
  
 //System.out.println(LineAnalysis(test1));  
 }  
}