# **МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ**

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**

**«ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИНЖЕНЕРНЫХ ТЕХНОЛОГИЙ»**



##### **Факультет управления и информатики в технологических системах**

###### **Кафедра Информационной безопасности**

**Направление подготовки**

**(специальность) 10.05.03 Информационная безопасность автоматизированных систем**

**Отчет**

**по практике по технологиям и методам программирования**

наименование (вид) практики

Выполнил студент гр. УБ-42

Крылов Никита Романович

###### (Ф.И.О.)

###### \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

*(подпись)*

###### Проверили:

Маслов А.А.

*(Ф.И.О.)*

*(оценка) (подпись)*

*(дата)*

Воронеж - 2025

**Задание 15.**

Найти, каких букв, гласных или согласных, больше в каждом предложении текста.

Код программы.  
package *K*urs\_2.*P*r1;

public class Var15 {

private static boolean isVowel(char c) { //Гласные

c = Character.toLowerCase(c);

return "ауоыиэяюёеaeiouy".indexOf(c) != -1;

}

private static boolean isConsonant(char c) { //Согласные

c = Character.toLowerCase(c);

return (c >= 'а' && c <= 'я' && !isVowel(c)) ||

(c >= 'a' && c <= 'z' && !isVowel(c));

}

public static void main(String[] args) {

String text = "Привет, мир! Это тестовый текст. Hello world! Как дела?";

String[] sentences = text.split("[.!?]+");

System.**out**.println("Анализ текста:\n" + text + "\n");

for (int i = 0; i < sentences.**length**; i++) {

if (sentences[i].trim().isEmpty()) continue;

String sentence = sentences[i].trim();

int vowels = 0;

int consonants = 0;

for (char c : sentence.toCharArray()) {

if (isVowel(c)) vowels++;

else if (isConsonant(c)) consonants++;

}

System.**out**.println("Предложение " + (i + 1) + ": \"" + sentence + "\"");

System.**out**.println("Гласных: " + vowels + ", Согласных: " + consonants);

if (vowels > consonants) {

System.**out**.println("✓ Гласных больше\n");

} else if (consonants > vowels) {

System.**out**.println("✓ Согласных больше\n");

} else {

System.**out**.println("✓ Равное количество\n");

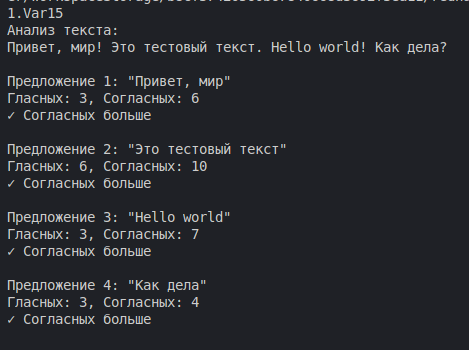
}

}

}

}

Вывод программы:



Вывод

Сегодня я изучил и научился решать задачу лингвистического анализа текста на уровне отдельных предложений, определяя количественное преобладание гласных или согласных букв в каждом из них. Этот практический пример позволил мне закрепить ключевые навыки работы со строками, символами, методами класса Character, регулярными выражениями и базовыми алгоритмами обработки и анализа текстовой информации на Java.

Изучив и реализовав это задание, я осознал важность начального этапа обработки текста – его корректного разделения на semantic units (семантические единицы), в данном случае – предложения. Для этого я использовал метод split() класса String с регулярным выражением "[.!?]+". Это выражение эффективно находит последовательности символов, обозначающих конец предложения (точку, восклицательный и вопросительный знаки).

Ключевым аспектом задачи была классификация каждого символа внутри предложения. Я реализовал эту логику через два вспомогательных приватных метода: isVowel(char c) для проверки, является ли символ гласной, и isConsonant(char c) для проверки на согласную. Это наглядный пример принципа декомпозиции – разбиения сложной задачи на более простые, логически законченные методы. Такой подход значительно улучшает читаемость кода, его структуру, облегчает процесс отладки и потенциального расширения функциональности. Важно отметить, что оба метода приводят символ к нижнему регистру с помощью Character.toLowerCase(c), что делает проверку регистронезависимой и корректно обрабатывает слова, начинающиеся с заглавной буквы.

Метод isConsonant(char c) реализован особенно интересно: он проверяет, принадлежит ли символ к диапазону букв русского или английского алфавита ((c >= 'а' && c <= 'я') или (c >= 'a' && c <= 'z')) и при этом не является гласной (!isVowel(c)). Это гарантирует, что знаки препинания, пробелы и цифры не будут учтены в подсчете, что критически важно для точности анализа.

В основном методе main после разделения текста на предложения для каждого из них осуществляется итерация по всем символам с помощью цикла for-each и преобразования строки в массив символов toCharArray(). Для каждого символа увеличивается соответствующий счетчик (vowels или consonants). По итогам подсчета происходит сравнение и формируется понятный вывод, указывающий на преобладающий тип букв или их равенство.