Ссылка на Git с заданием 5 (Задача 1 и Задача 2):

<https://github.com/ekbond/HW5>

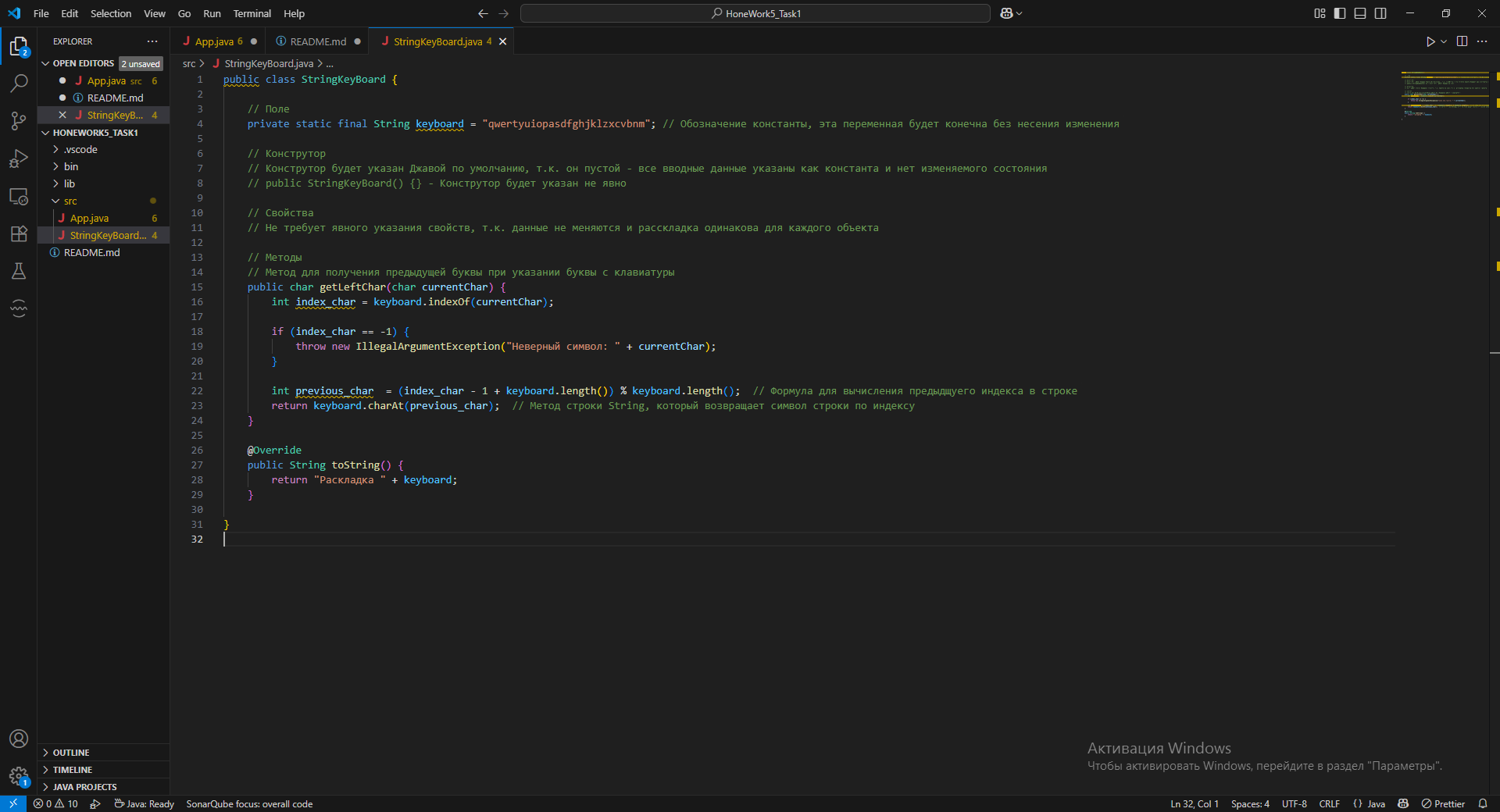
Задача №1

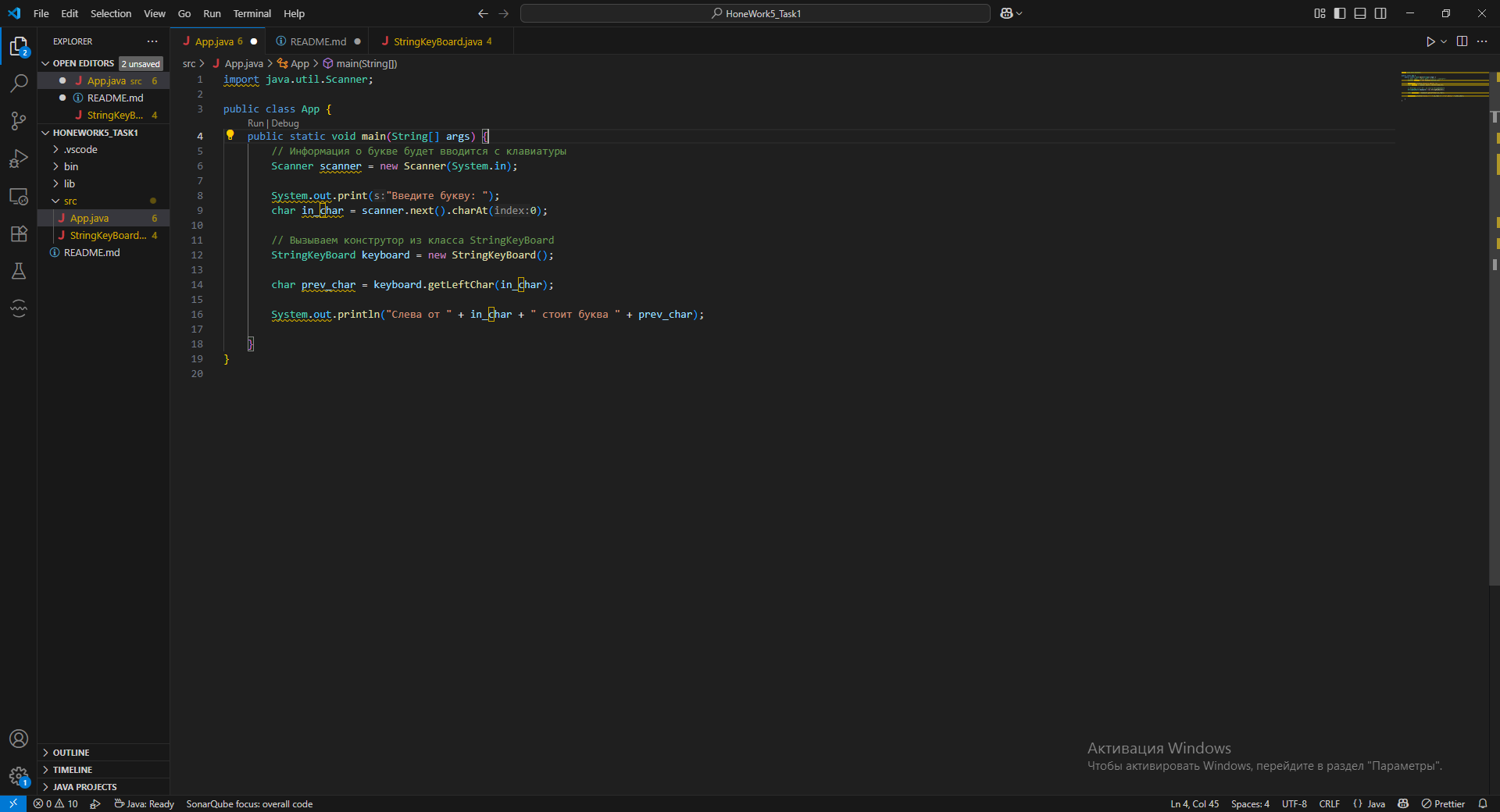
Для введенной с клавиатуры буквы английского алфавита нужно вывести слева стоящую букву на стандартной клавиатуре. При этом клавиатура замкнута, т.е. справа от буквы «p» стоит буква «a», а слева от "а" буква "р", также соседними считаются буквы «l» и буква «z», а буква «m» с буквой «q».

Входные данные: строка входного потока содержит один символ — маленькую букву английского алфавита.

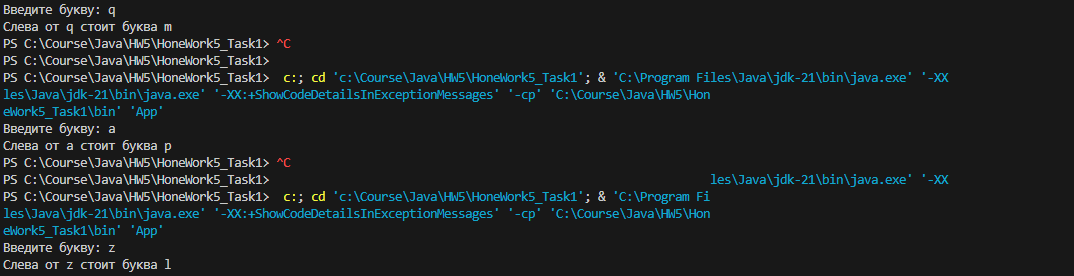
Выходные данные: следует вывести букву стоящую слева от заданной буквы, с учетом замкнутости клавиатуры.

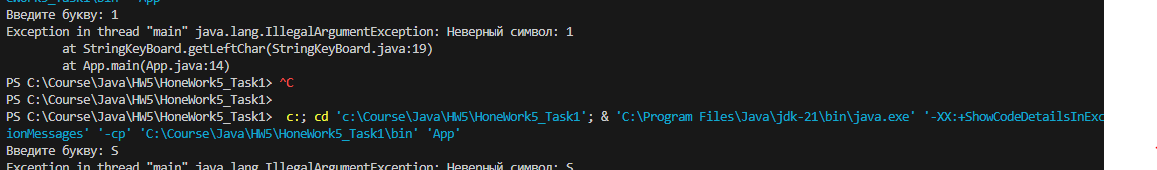
Скрипт:

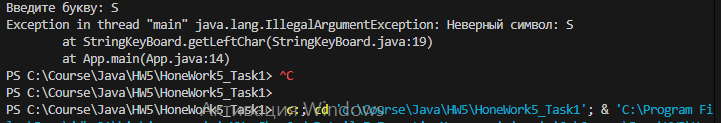




Вывод:







Readme:

**# Определение последующей буквы - Java реализация**

**## Описание проекта**

Проект представляет собой реализацию класса StringKeyBoard на Java для определения буквы, расположенной слева от заданной на стандартной QWERTY-клавиатуре с учётом её замкнутой структуры с демонстрацией работы в классе `App`.

**## Основные характеристики класса**

- **\*\*Поля класса\*\***:

`keyboard` - константная строка, содержащая последовательность букв клавиатуры в нижнем регистре (статическое и неизменяемое поле)

- **\*\*Основные методы\*\***:

- `getLeftChar(char currentChar)` - возвращает букву, расположенную слева от заданной

- `toString()` - переопределённый метод для вывода информации о раскладке клавиатуры

**## Особенности реализации**

1. Использование констант - раскладка клавиатуры задана как static final

2. Замкнутая логика - учитывает круговую структуру клавиатуры (после 'p' идёт 'a')

3. Способ создания телевизора:

   - С вводом параметров с клавиатуры

**## Как использовать**

1. Клонировать репозиторий

2. Открыть проект в IDE (IntelliJ IDEA или VScode.)

3. Запустить класс `App`

**## Пример вывода**

**### Пример\_1**

Введите букву: s

Слева от s стоит буква a

**### Пример\_2**

Введите букву: a

Слева от a стоит буква p

**### Пример\_3**

Введите букву: 1

Exception in thread "main" java.lang.IllegalArgumentException: Неверный символ: 1

**## Пример\_4**

Введите букву: S

Exception in thread "main" java.lang.IllegalArgumentException: Неверный символ: S

**### Ограничения**

1. Работает только с английскими строчными буквами

2. Не поддерживает специальные символы и цифры

3. Учитывает только стандартную QWERTY-расклад

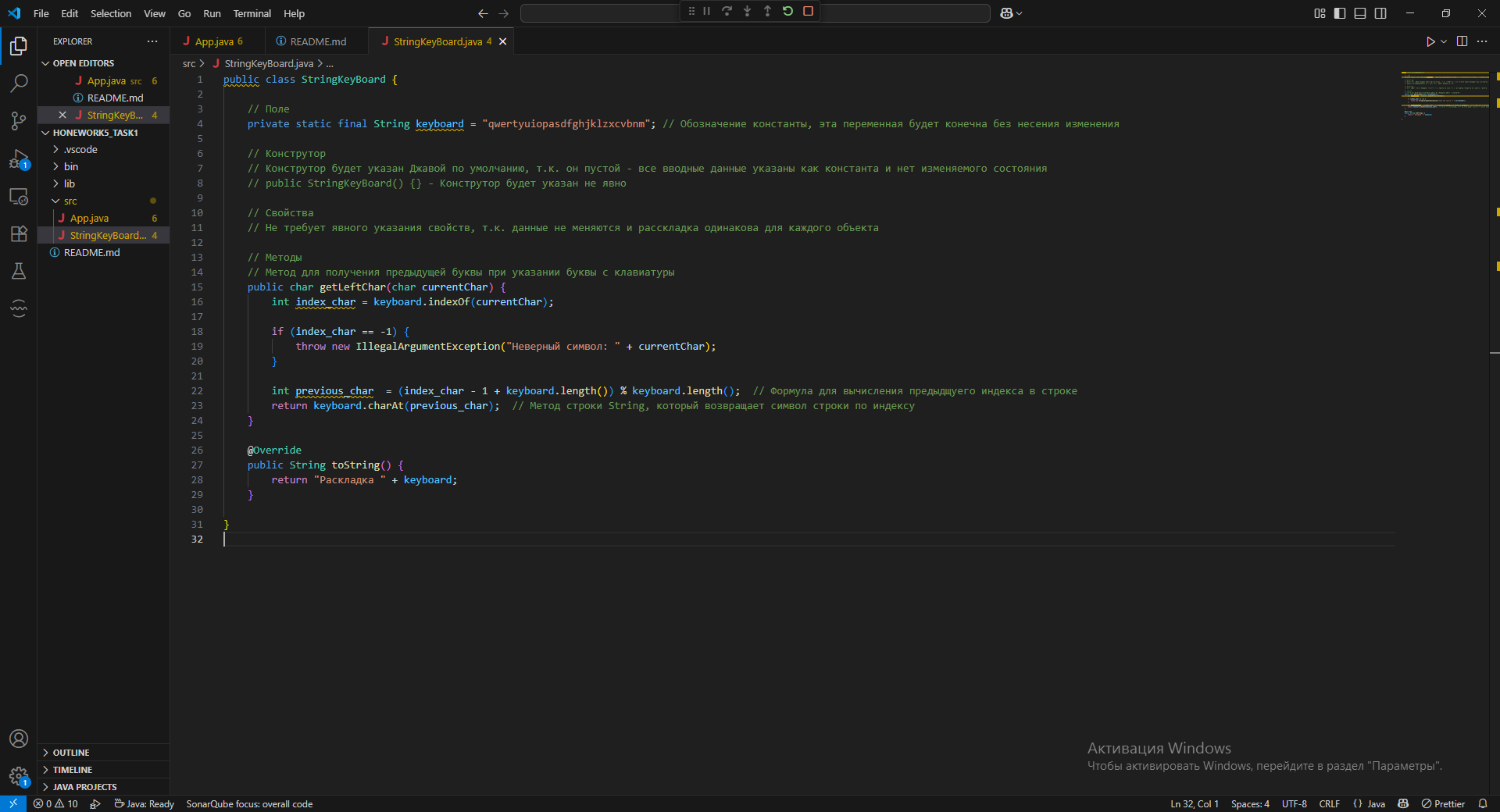
Задача №2

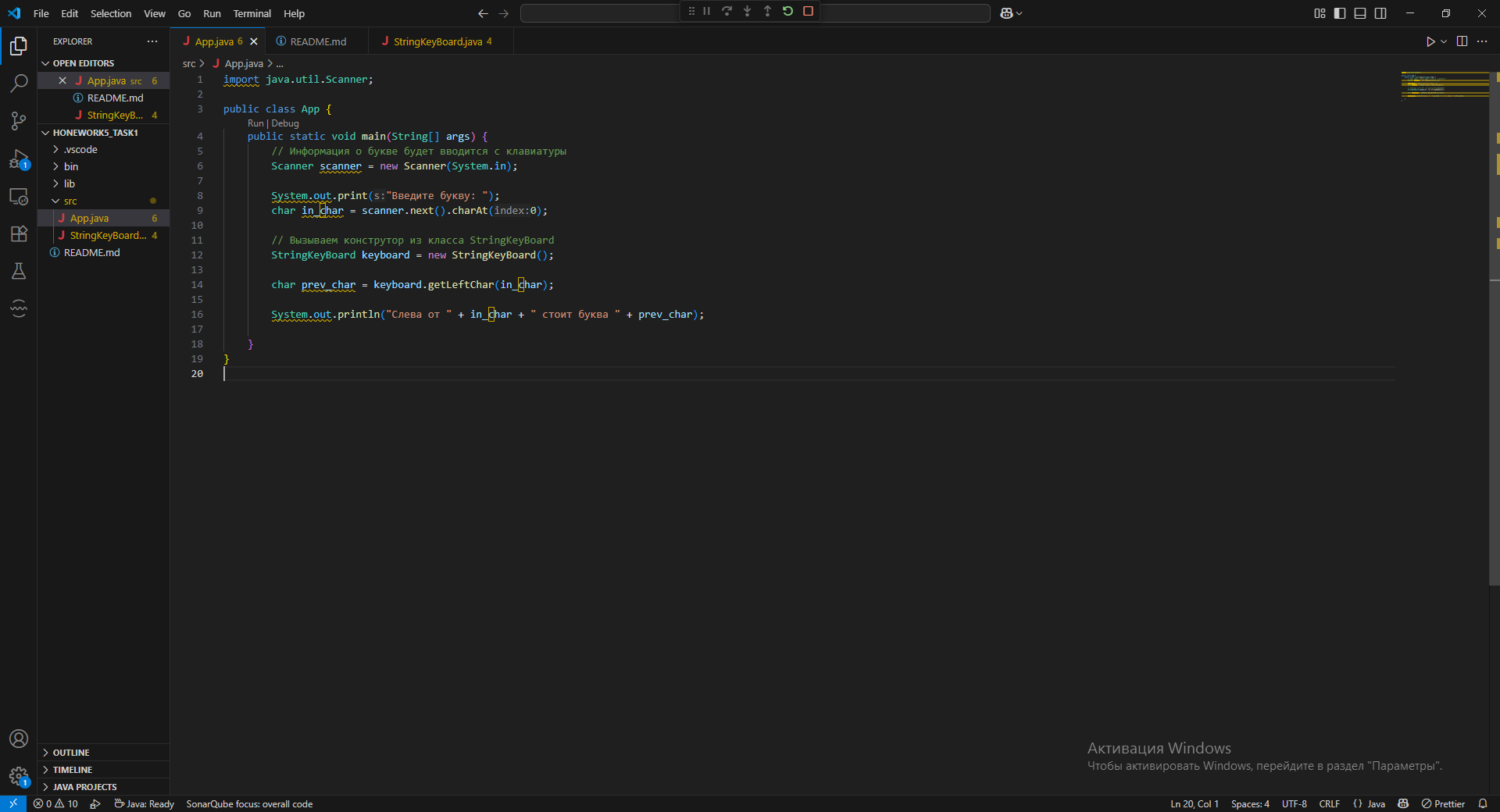
Задана последовательность, состоящая только из символов ‘>’, ‘<’ и ‘-‘. Требуется найти количество стрел, которые спрятаны в этой последовательности. Стрелы – это подстроки вида ‘>>-->’ и ‘<--<<’.

Входные данные: в первой строке входного потока записана строка, состоящая из символов ‘>’, ‘<’ и ‘-‘ (без пробелов). Строка может содержать до 106 символов.

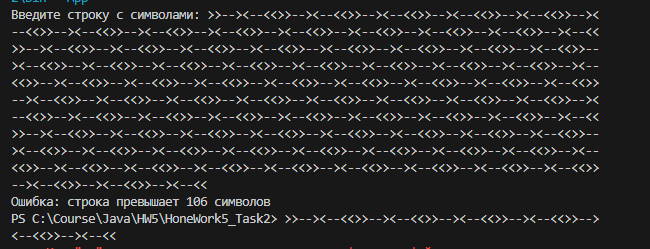
Выходные данные: в единственную строку выходного потока нужно вывести искомое количество стрелок

Скрипт:





Вывод:







Readme:

**# Количество стрел в строке - Java реализация**

**## Описание проекта**

Проект представляет собой реализацию класса `CountArrow` на Java для подсчёта специальных последовательностей (стрелок) в строке с демонстрацией работы в классе `App`.

**## Основные характеристики класса**

- **\*\*Поля класса\*\***:

Не содержит явных полей, так как не хранит состояние объектара

- **\*\*Основные методы\*\***:

`countArrows(String sequence)` - подсчитывает количество стрелок ">>-->" и "<--<<" в строке

**## Особенности реализации**

1. Простота использования - класс не требует инициализации состояния

2. Проверка допустимых значений (Ограничение на длину строки (до 106 символов))

3. Способ создания телевизора:

   - С вводом параметров с клавиатуры

**## Как использовать**

1. Клонировать репозиторий

2. Открыть проект в IDE (IntelliJ IDEA или VScode.)

3. Запустить класс `App`

**## Пример вывода**

**### Пример\_1**

Введите строку с символами: >>--><--<<

Количество стрел (>>--> или <--<<) в строке: 2

**### Пример\_2**

Введите строку с символами: >--><--<<>>--><--<<>>--><--<<>>--><--<<>>--><--<<>>--><--<<>>--><--<<>>--><--<<>>--><--<<>>--><--<<>>--><--<<>>--><--<<>>--><--<<>>--><--<<>>--><--<<>>--><--<<>>--><--<<>>--><--<<>>--><--<<>>--><--<<>>--><--<<>>--><--<<>>--><--<<>>--><--<<>>--><--<<>>--><--<<>>--><--<<>>--><--<<>>--><--<<>>--><--<<>>--><--<<>>--><--<<>>--><--<<>>--><--<<>>--><--<<>>--><--<<>>--><--<<>>--><--<<>>--><--<<>>--><--<<>>--><--<<>>--><--<<>>--><--<<>>--><--<<>>--><--<<>>--><--<<>>--><--<<>>--><--<<>>--><--<<>>--><--<<>>--><--<<>>--><--<<>>--><--<<>>--><--<<>>--><--<<>>--><--<<>>--><--<<>>--><--<<>>--><--<<>>--><--<<>>--><--<<>>--><--<<>>--><--<<>>--><--<<>>--><--<<>>--><--<<>>--><--<<>>--><--<<>>--><--<<>>--><--<<>>--><--<<>>--><--<<>>--><--<<>>--><--<<>>--><--<<>>--><--<<>>--><--<<>>--><--<<>>--><--<<>>--><--<<>>--><--<<>>--><--<<>>--><--<<>>--><--<<

Ошибка: строка превышает 106 символов

**### Пример\_3**

Введите строку с символами: <<->>--<->>

Количество стрел (>>--> или <--<<) в строке: 0

**### Ограничения**

1. Обрабатывает только строки длиной до 106 символов

2. Строка должна содержать только символы: '>', '<', '-' (без пробелов)