

Лабораторная работа №1

Код программы продемонстрирован на рисунках 1,2,3:

1) Класс CountController:

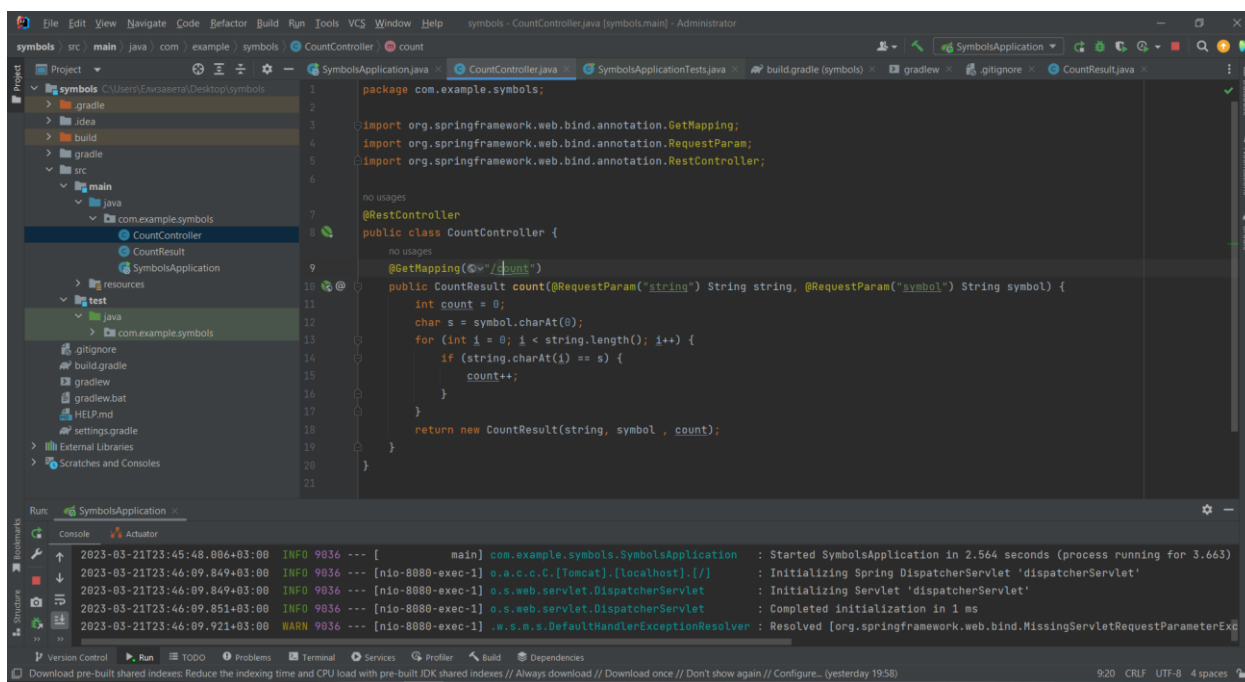


Рисунок 1

2) Класс CountResult:

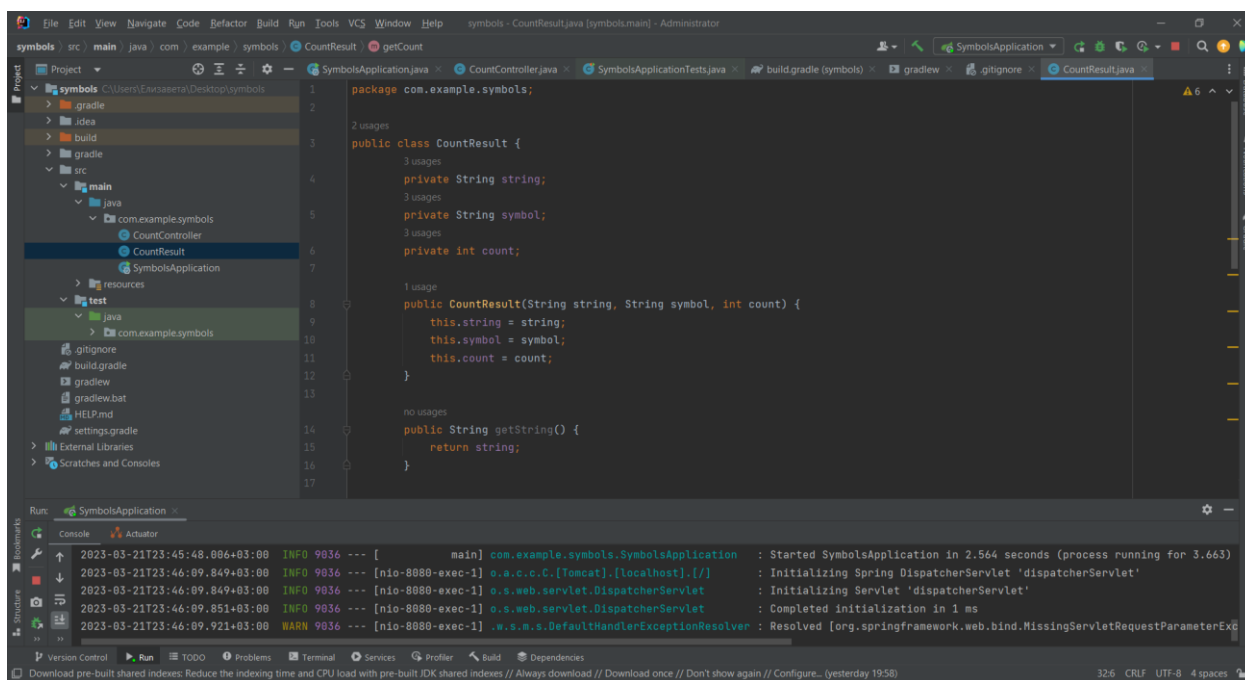


Рисунок 2

Продолжение данного класса:

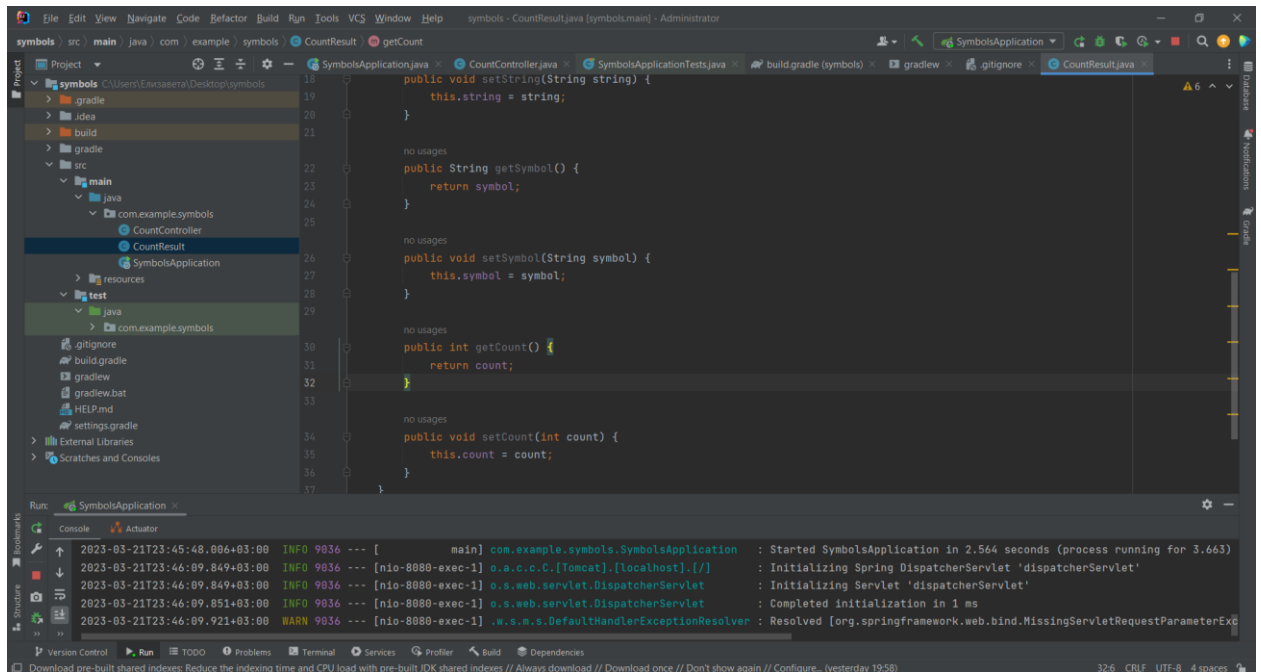


Рисунок 3

После запуска приложения, переходя по адресу <http://localhost:8080/count?string=yourstring&symbol=y> в веб-браузере, где `yourstring` - это строка, в которой по заданию нужно найти символ, а `y` - символ, который будем искать и подсчитывать, сколько раз он встречается в данной строке. Далее видим ответ в формате JSON в браузере.

Скриншоты, на которых продемонстрирована работа программы (см. рис. 4,5,6,7):



Рисунок 4



Рисунок 5



Рисунок 6



Рисунок 7

Работа программы через Postman (см. рис.8):

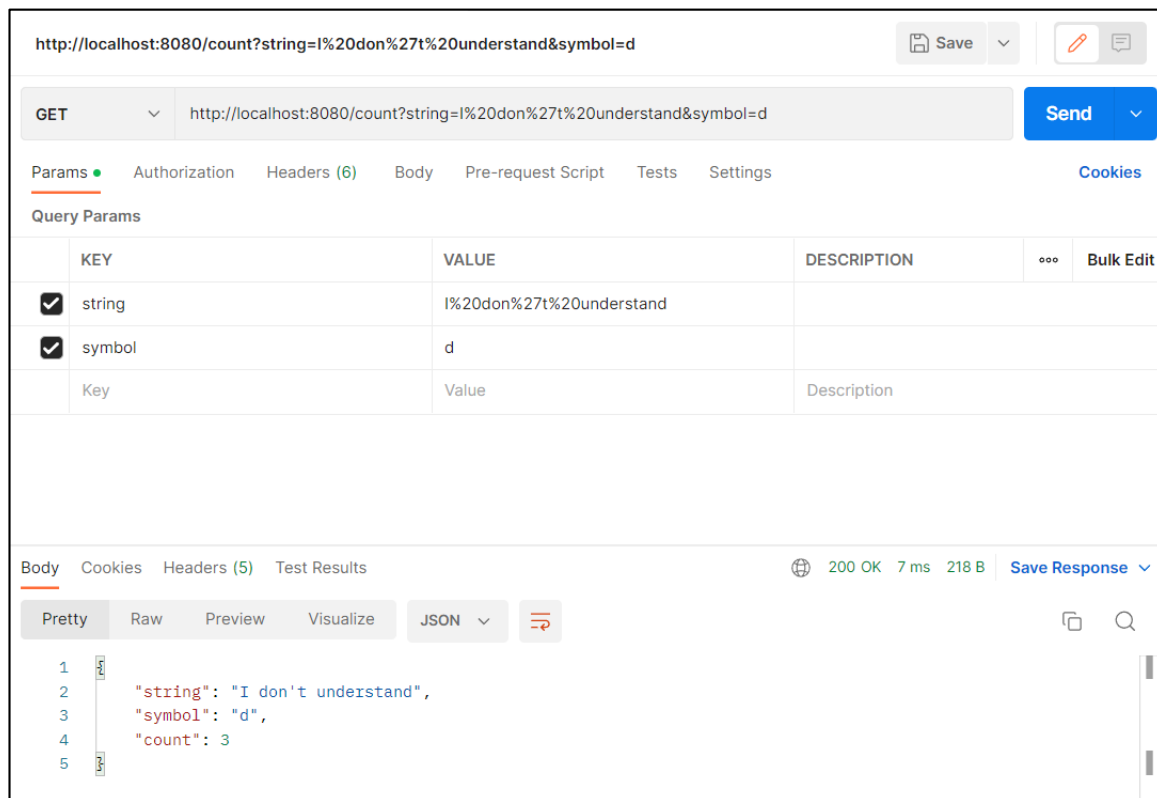


Рисунок 8

Лабораторная работа №2

В данном коде были добавлены:

- Валидация входных параметров с возвращением 400 ошибки;
- Обработка внутренних unchecked ошибок с возвратом 500 ошибки;
- Логирование действий и ошибок

Код изображён на рисунках 9,10

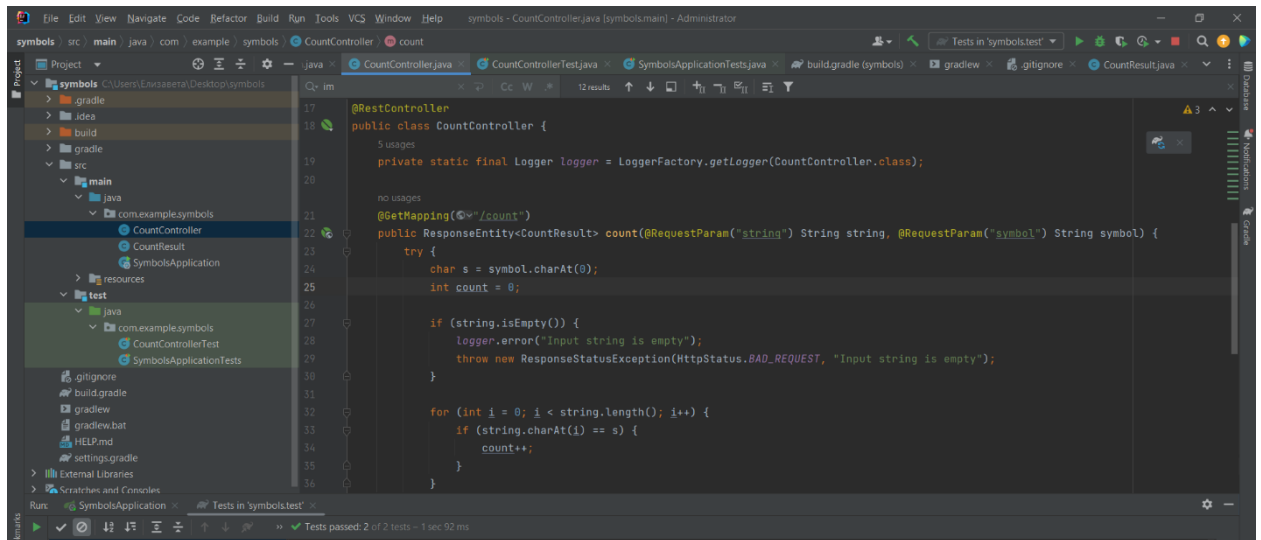


Рисунок 9

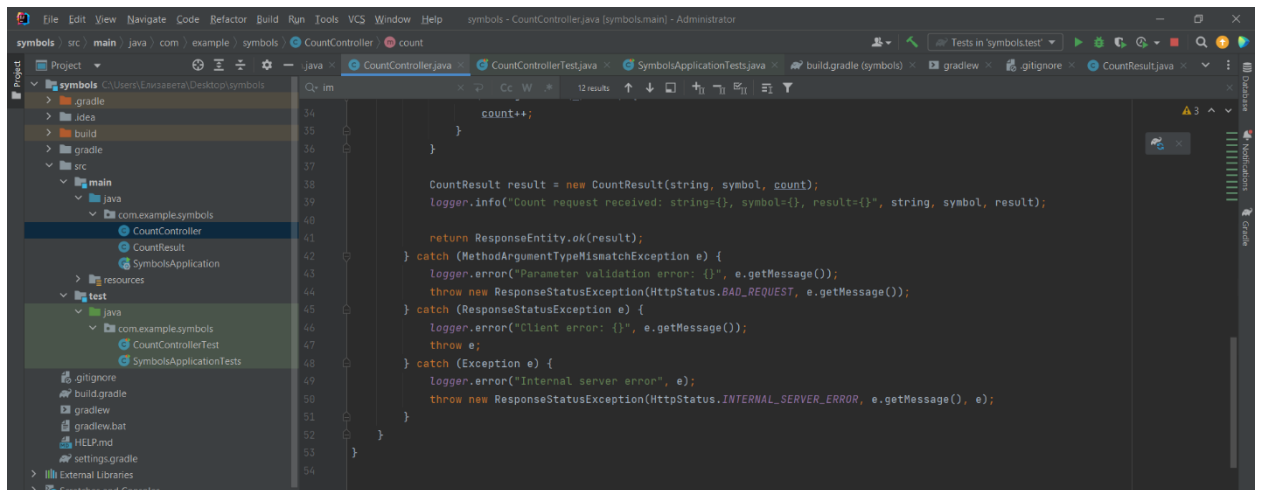


Рисунок 10

Скриншоты, на которых продемонстрирована работа программы (см. рис. 11,12,13,14):

400 ошибка:

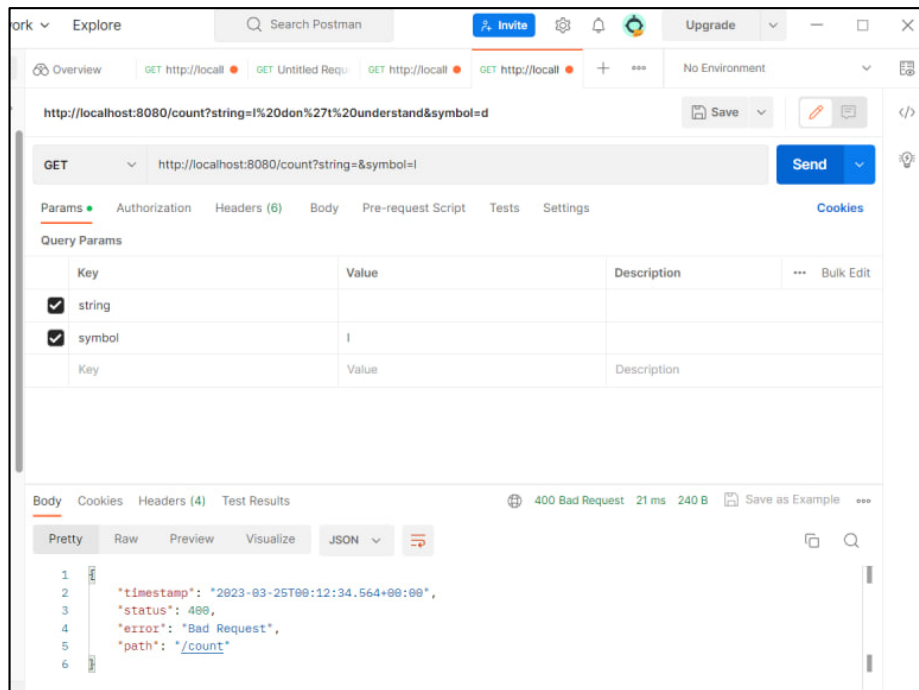


Рисунок 11

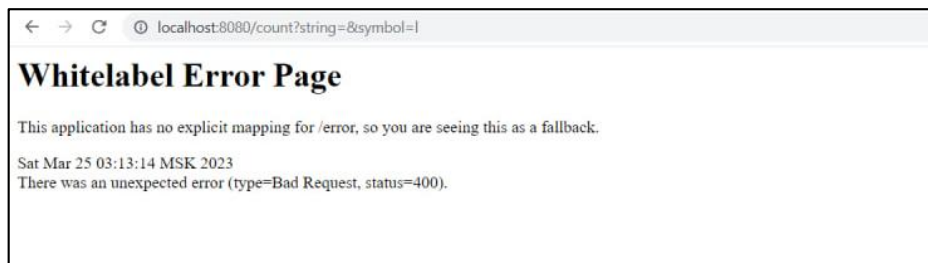


Рисунок 12

500 ошибка:

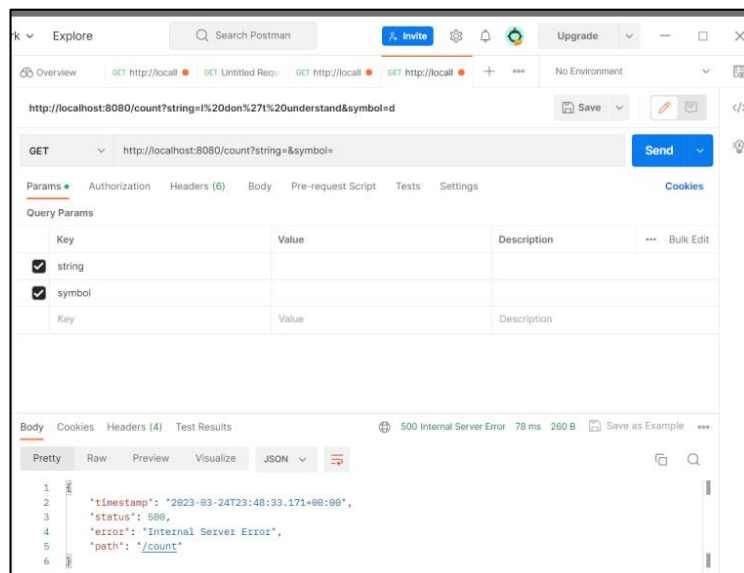


Рисунок 13



Рисунок 14

Далее отображено выполнение задания: написать unit test. См. рис.15,16,17.

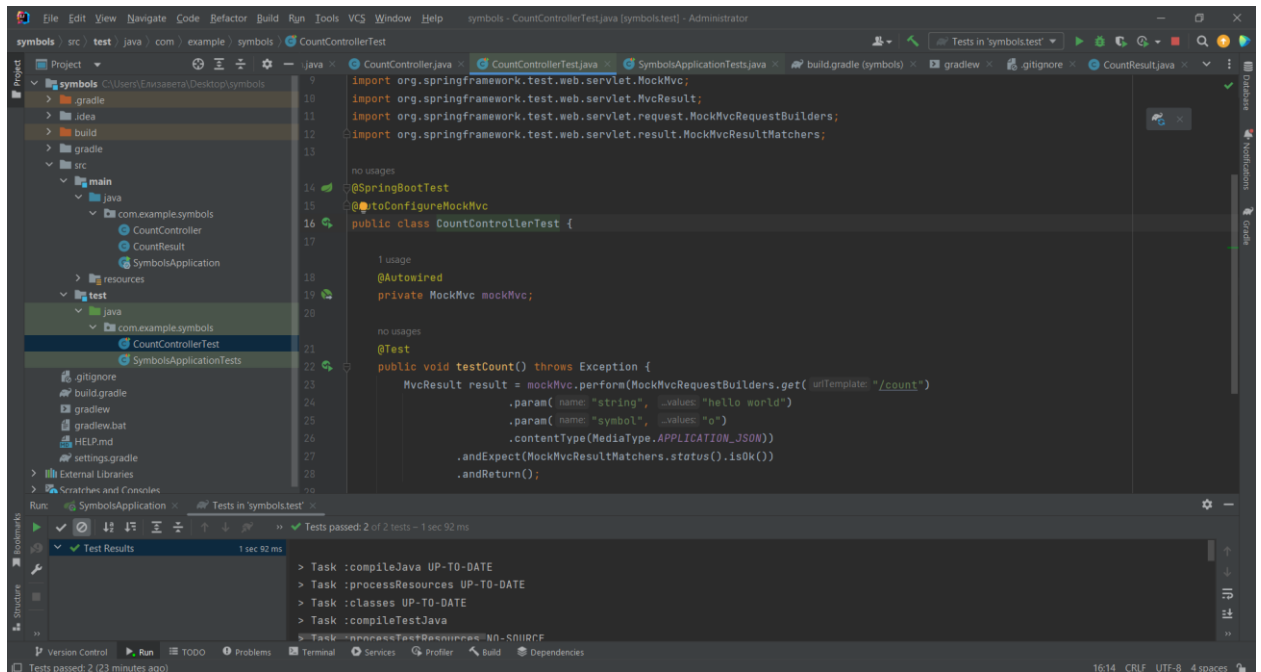


Рисунок 15

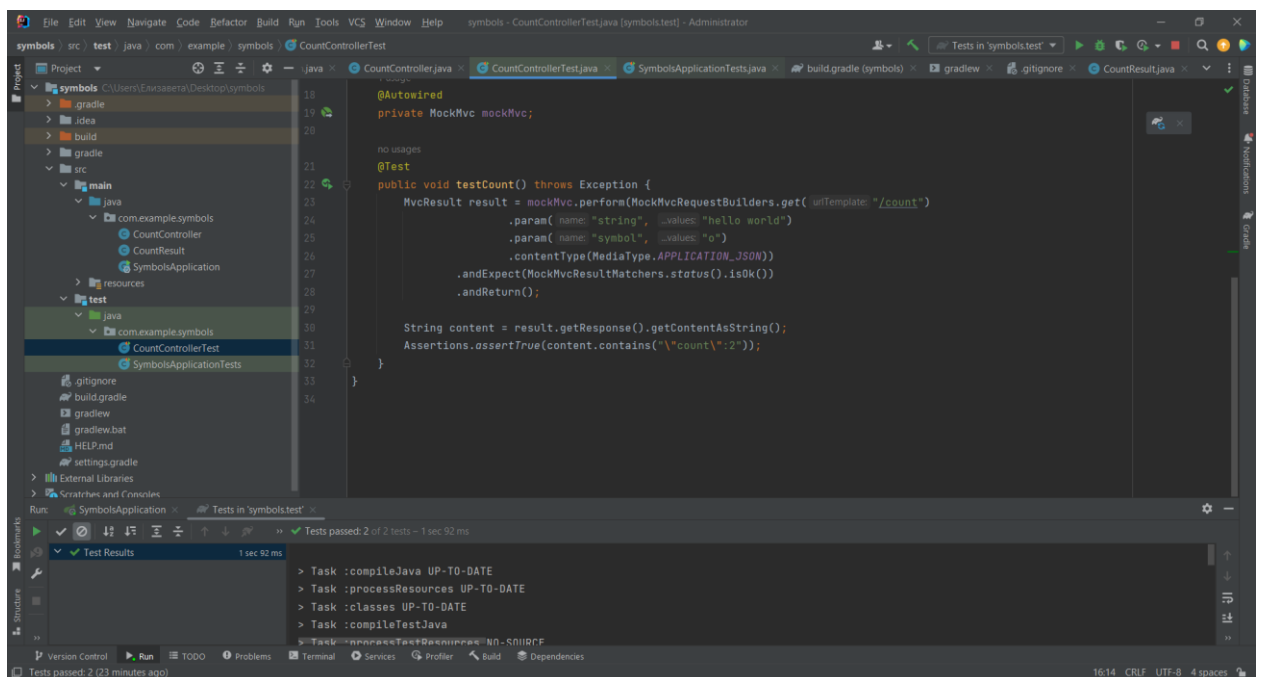


Рисунок 16

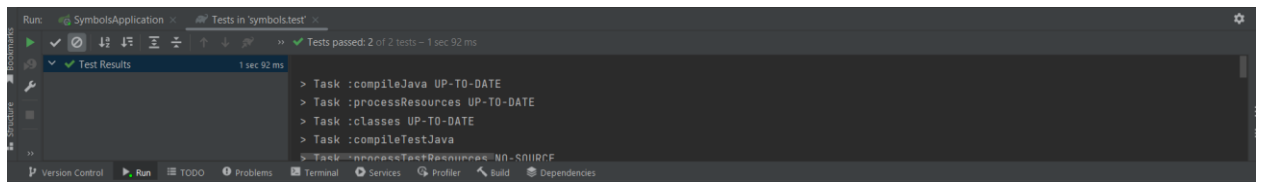


Рисунок 17