Код программы продемонстрирован на рисунках 1,2,3:

1) Класс CountController:

```
| Dire | Set | Year | Services | Code | Sefector | Build Plan | Took | VC | Window | Help | ymmboin | CountController | Count | Dire | V | Reg Symbolideprication | V | Reg Symbolideprication
```

Рисунок 1

2) Класс CountResult:

```
De Delt Vew Namejare Code Beactor Build Fun Jook VCS Weedow Biele symbols - Countificant/java lymbols.counterand/sate/personal parameters and the property of the property of
```

Рисунок 2

Продолжение данного класса:

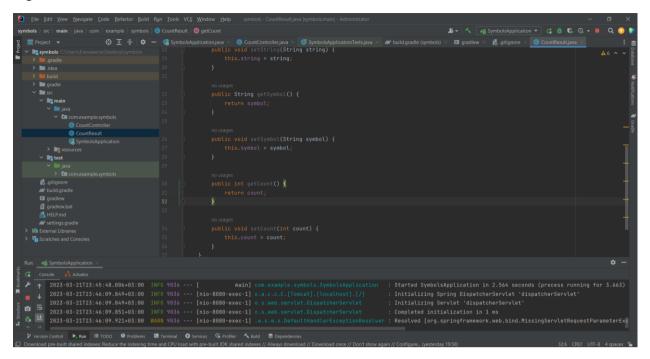


Рисунок 3

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 у - 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):

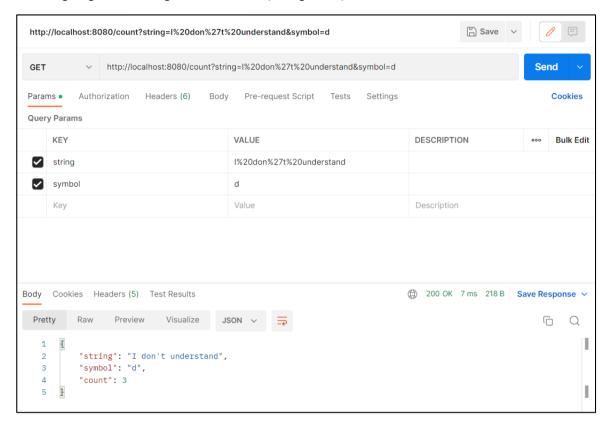


Рисунок 8

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

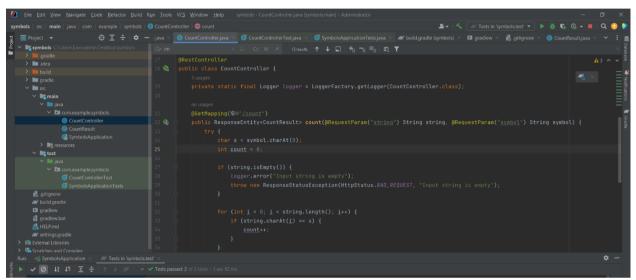


Рисунок 9

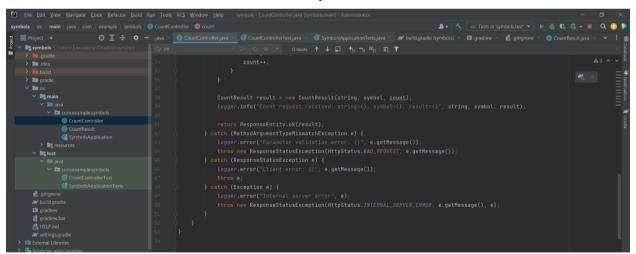


Рисунок 10

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

400 ошибка:

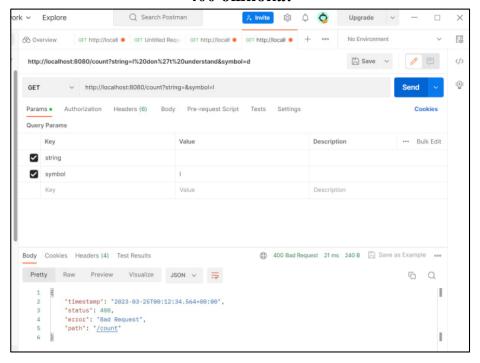


Рисунок 11



Рисунок 12

500 ошибка:

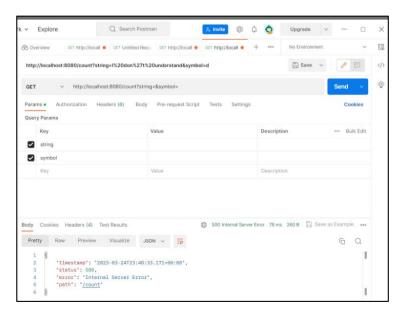


Рисунок 13



Рисунок 14

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

```
| Power | Confidence | Power | Power | Confidence |
```

Рисунок 15

```
| Second control | Seco
```

Рисунок 16



Рисунок 17

1) Класс CountController (см. рис.18, 19,20,21):

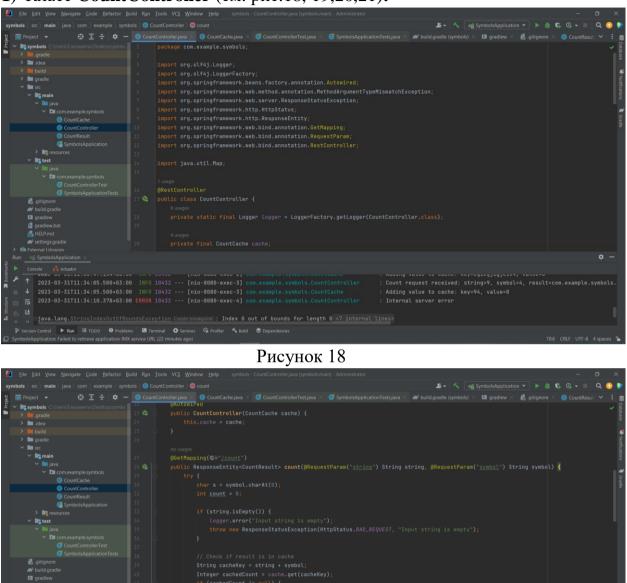


Рисунок 19

2023-03-31711:34:05.500+03:00 INFO 10432 --- [nio-8080-exec-3] com.example.symbols.CountController 2023-03-31711:34:05.500+03:00 INFO 10432 --- [nio-8080-exec-3] com.example.symbols.CountController 2023-03-31711:34:10.378+03:00 ERROR 10432 --- [nio-8080-exec-4] com.example.symbols.CountController

```
return ResponseEntity.ok(result);
catch (MethodArgumentTypeHismatchException e) {
  logger.error("Parameter validation error: {}", e.getHessage());
  throw new ResponseStatusException(HttpStatus.BAD_REQUEST, e.getHessage());
  catch (ResponseStatusException e) {
    logger.error("Client error: {}", e.getHessage());
    throw e;
  catch (Exception e) {
    logger.error("Internal server error", e);
    throw new ResponseStatusException(HttpStatus.INTERNAL_SERVER_ERROR, e.getHessage(), e);
2023-03-31711:34:80.5309-03:00 INFO 10432 --- [nio-8080-exec-3] com.example.symbols.countController : Count request received: string=9, symbols.4; result=com.example.symbols.2001.tCountController : Adding value to cache: key=94, value=0
2023-03-31711:34:10.378-03:00 ERROR 10432 --- [nio-8080-exec-4] com.example.symbols.CountController : Internal server error
                                                                                                                                                                                                  Рисунок 20
```

```
2023-03-31711:34:05.500+03:00 INFO 10432 --- [nio-8080-exec-3] com.example.symbols.CountController 2023-03-31711:34:05.500+03:00 INFO 10432 --- [nio-8080-exec-3] com.example.symbols.CountCoche 2023-03-31711:34:10.378+03:00 ERROR 10432 --- [nio-8080-exec-4] com.example.symbols.CountCoche 2023-03-31711:34:10.378+03:00 ERROR 10432 --- [nio-8080-exec-4] com.example.symbols.CountController
```

Рисунок 21

2) Класс CountCache (см.рис. 22,23):

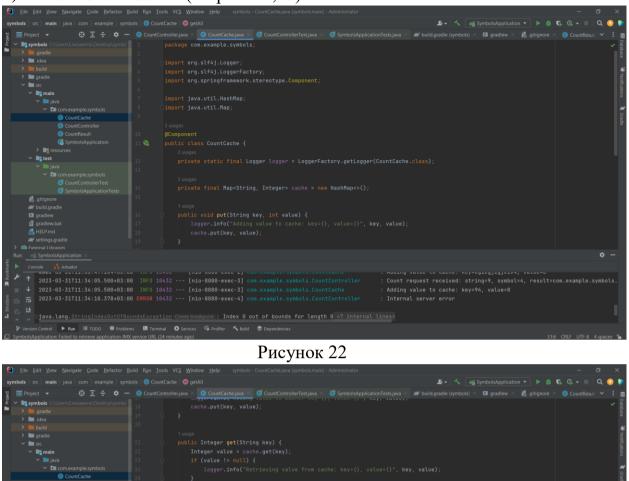


Рисунок 23

blic Map<String, Integer> getAll() {
 return new HashMap<>(cache);

Класс CountResult остаётся без изменений.

2023-03-31711:34:05.500+03:00 INFO 10432 --- [nio-8080-exec-3] com.example.symbols.CountController 2023-03-31711:34:05.500+03:00 INFO 10432 --- [nio-8080-exec-3] com.example.symbols.CountController 2023-03-31711:34:10.378+03:00 ERROR 10432 --- [nio-8080-exec-4] com.example.symbols.CountController

После того, как произойдёт переход по URL /cache в браузере, Postman получим (см.рис.24,25):

• Если 3:

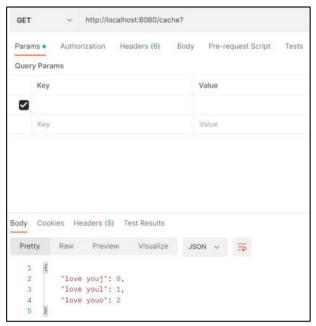


Рисунок 24

• Если 2:



Рисунок 25

1) Класс **Counter** (см.рис.26) :

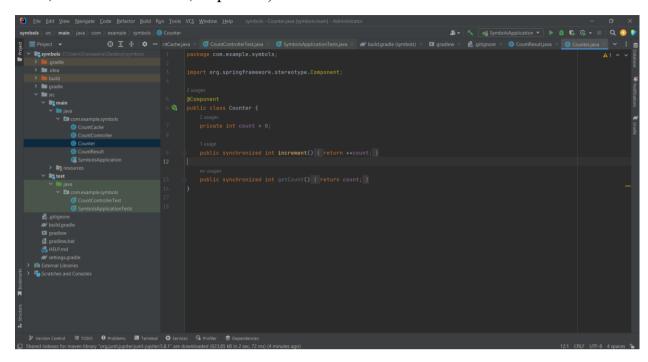


Рисунок 26

2) Класс CountController (см. рис. 27,28,29):

```
### Spin for Whoman Code Entance Balls Rep Dook VCS Whodow Jets memors CountController | Dook VCS Whoman Jets CountController | Dook Whoman Jets Whoman J
```

Рисунок 27

```
| The part of the
```

Рисунок 28

```
| Purpose Const. | Tool Of Proteins | Tool Of Proteins | Tool Const. | Purpose Const. | Pur
```

Рисунок 29

Рассмотрим, как работает программа (см.рис.30,31,32):

Если 3 итерации (рис.30):

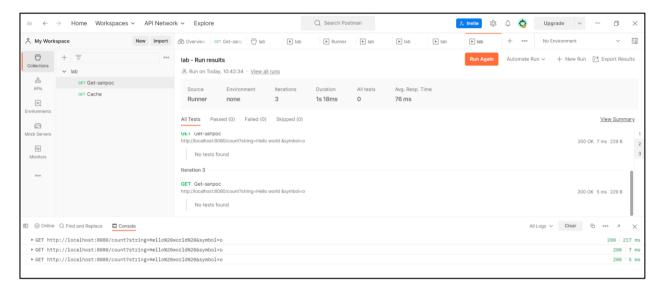


Рисунок 30

• В это время счётчик (рис.31):



Рисунок 31

Если мы снова проведём нагрузочный тест, добавив ещё 4 итерации, при всё том же запущенном коде, то count будет равен 7.

Пробовала и на большом кол-ве итерации, а именно -1000+. Всё работало так же, как и при маленьком, и при среднем кол-ве. Счётчик также считал корректно.

Для примера взяла 201 итерацию (рис.32) на 2 GET-запроса всё так же, как и до этого отработало без ошибок.

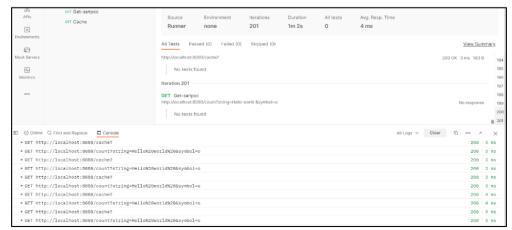


Рисунок 32