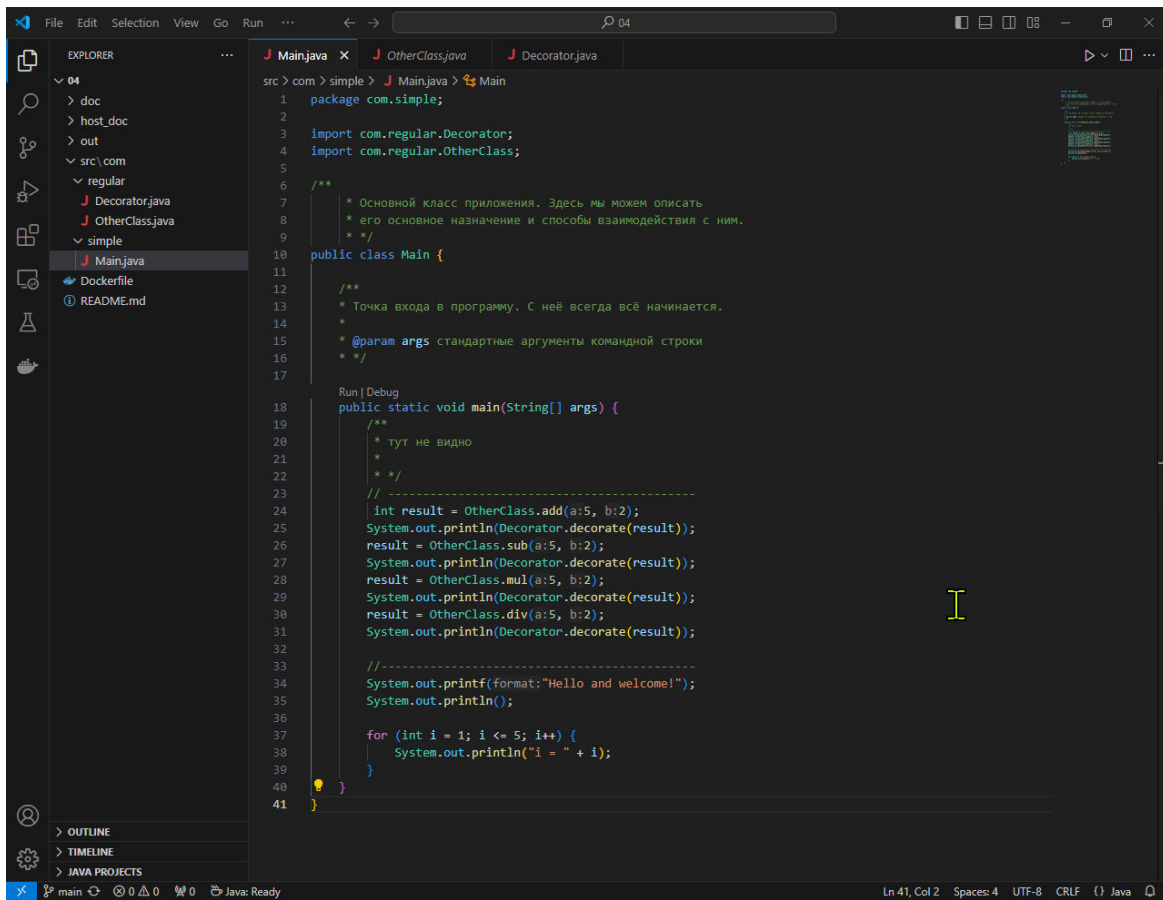
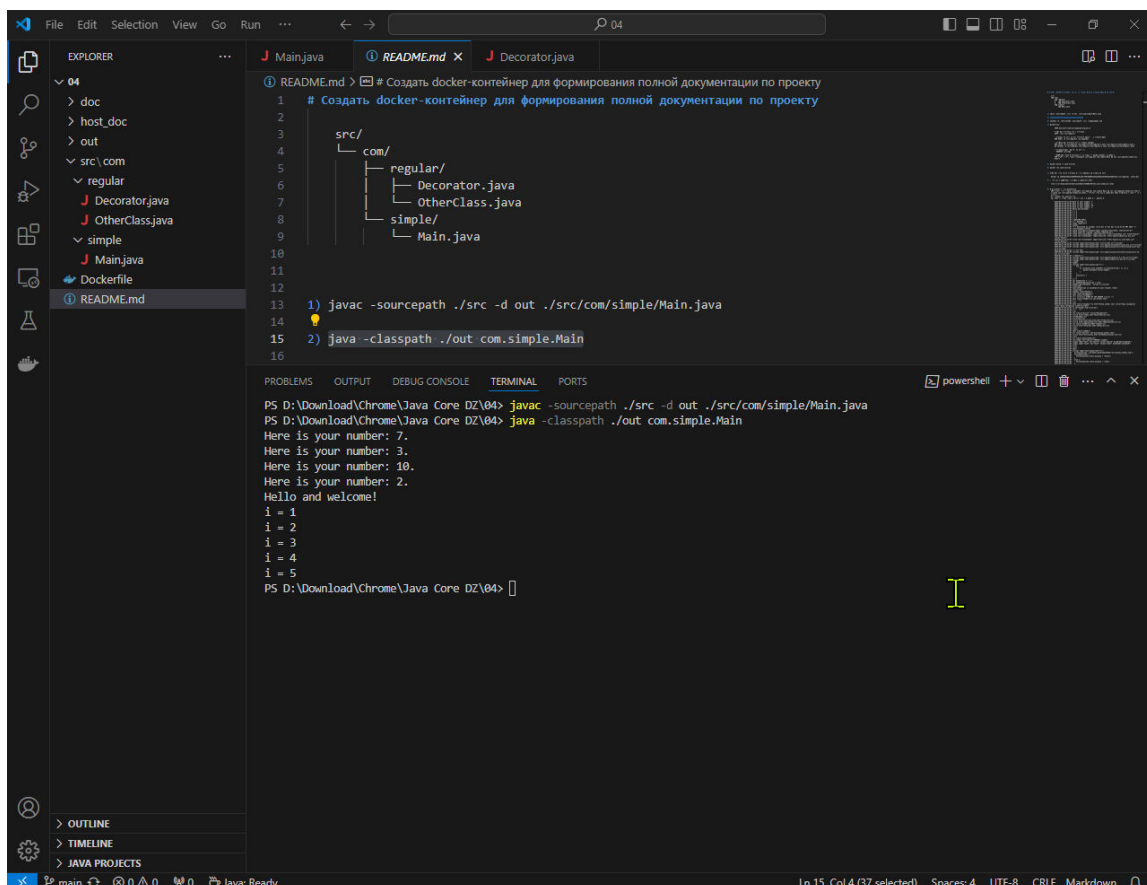


Создать docker-контейнер для формирования полной документации по проекту



```
src > com > simple > J Main.java > Main
1 package com.simple;
2
3 import com.regular.Decorator;
4 import com.regular.OtherClass;
5
6 /**
7  * Основной класс приложения. Здесь мы можем описать
8  * его основное назначение и способы взаимодействия с ним.
9  */
10 public class Main {
11
12     /**
13      * Точка входа в программу. С неё всегда всё начинается.
14      *
15      * @param args стандартные аргументы командной строки
16      */
17
18     Run | Debug
19     public static void main(String[] args) {
20         /**
21          * тут не видно
22          */
23         // -----
24         int result = OtherClass.add(a:5, b:2);
25         System.out.println(Decorator.decorate(result));
26         result = OtherClass.sub(a:5, b:2);
27         System.out.println(Decorator.decorate(result));
28         result = OtherClass.mul(a:5, b:2);
29         System.out.println(Decorator.decorate(result));
30         result = OtherClass.div(a:5, b:2);
31         System.out.println(Decorator.decorate(result));
32
33         //-----
34         System.out.printf(format:"Hello and welcome!");
35         System.out.println();
36
37         for (int i = 1; i <= 5; i++) {
38             System.out.println("i = " + i);
39         }
40     }
41 }
```



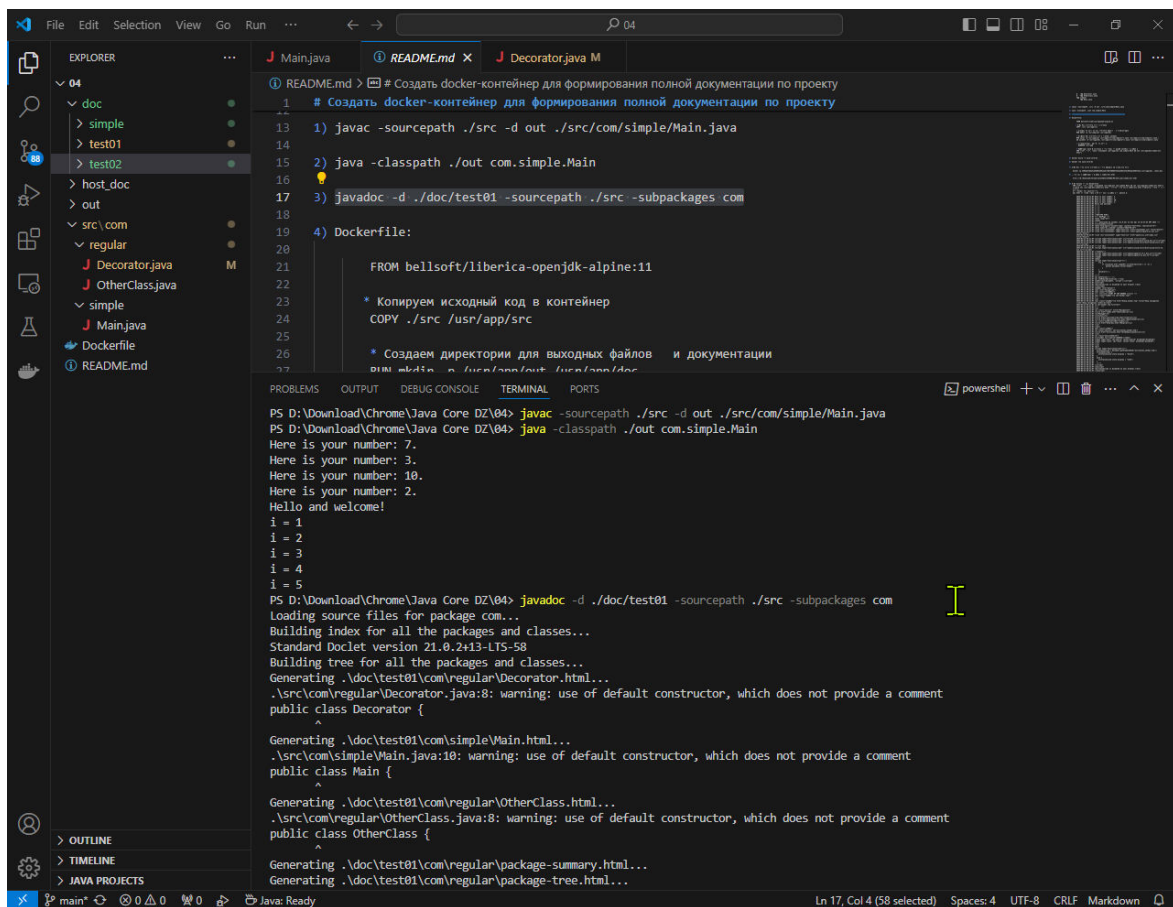
```
1 # Создать docker-контейнер для формирования полной документации по проекту
2
3 src/
4   └─ com/
5       └─ regular/
6           ├── Decorator.java
7           └─ OtherClass.java
8       └─ simple/
9           └─ Main.java
10
11
12
13 1) javac -sourcepath ./src -d out ./src/com/simple/Main.java
14
15 2) java -classpath ./out com.simple.Main
16
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS D:\Download\Chrome\Java Core DZ\04> javac -sourcepath ./src -d out ./src/com/simple/Main.java
PS D:\Download\Chrome\Java Core DZ\04> java -classpath ./out com.simple.Main
Here is your number: 7.
Here is your number: 3.
Here is your number: 10.
Here is your number: 2.
Hello and welcome!
i = 1
i = 2
i = 3
i = 4
i = 5
PS D:\Download\Chrome\Java Core DZ\04>
```

1) javac -sourcepath ./src -d out ./src/com/simple/Main.java

2) java -classpath ./out com.simple.Main



3) javadoc -d ./doc/test01 -sourcepath ./src -subpackages com

4) Dockerfile:

```
FROM bellsoft/liberica-openjdk-alpine:11
```

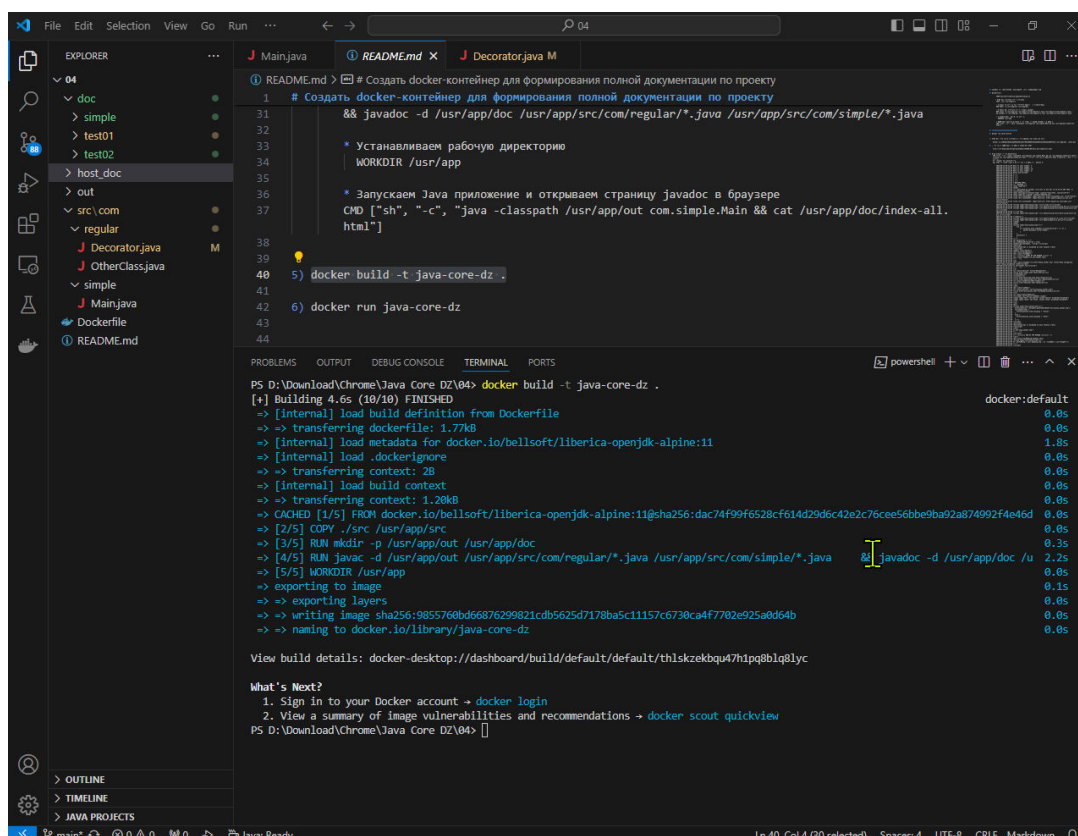
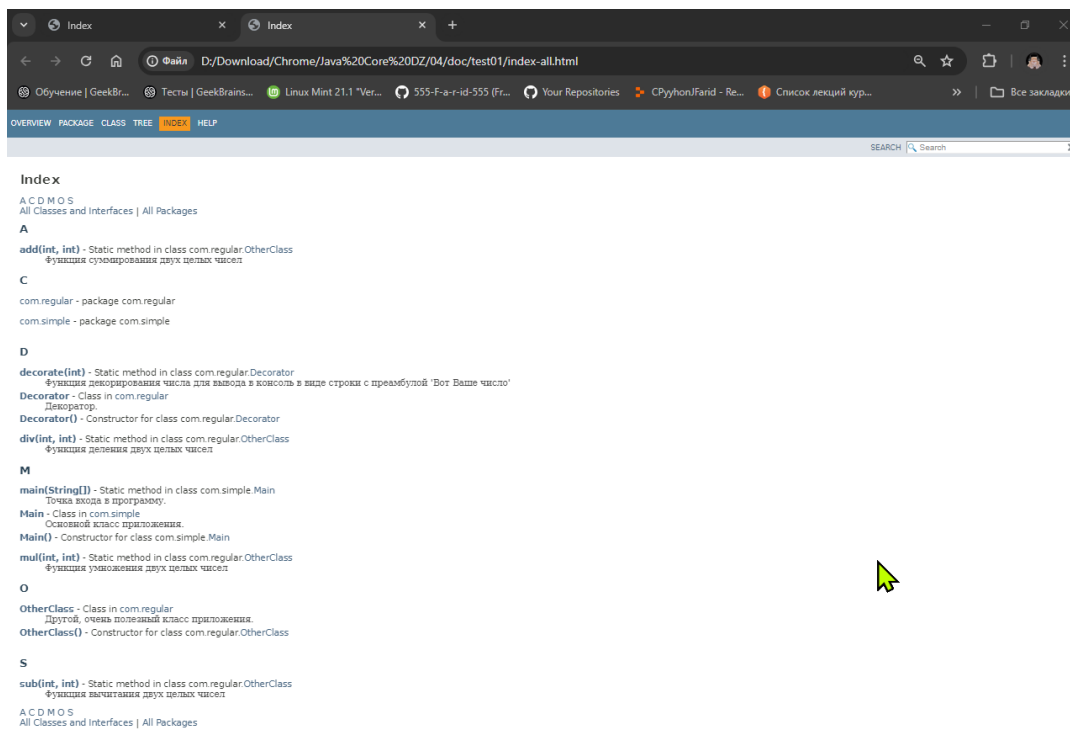
```
* Копируем исходный код в контейнер
COPY ./src /usr/app/src
```

```
* Создаем директории для выходных файлов и документации
RUN mkdir -p /usr/app/out /usr/app/doc
```

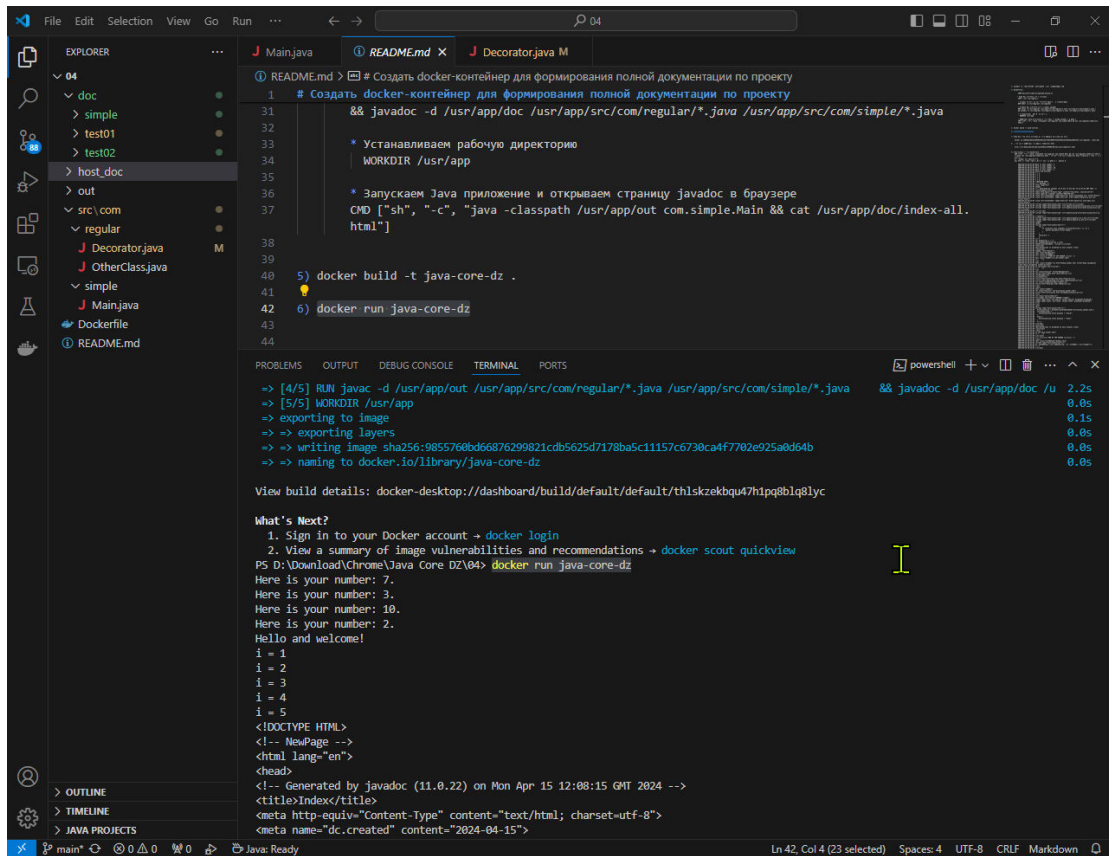
```
* Компилируем исходный код и создаем javadoc
RUN javac -d /usr/app/out /usr/app/src/com/regular/*.java /usr/app/src/com/simple/*.java \
&& javadoc -d /usr/app/doc /usr/app/src/com/regular/*.java /usr/app/src/com/simple/*.java
```

```
* Устанавливаем рабочую директорию
WORKDIR /usr/app
```

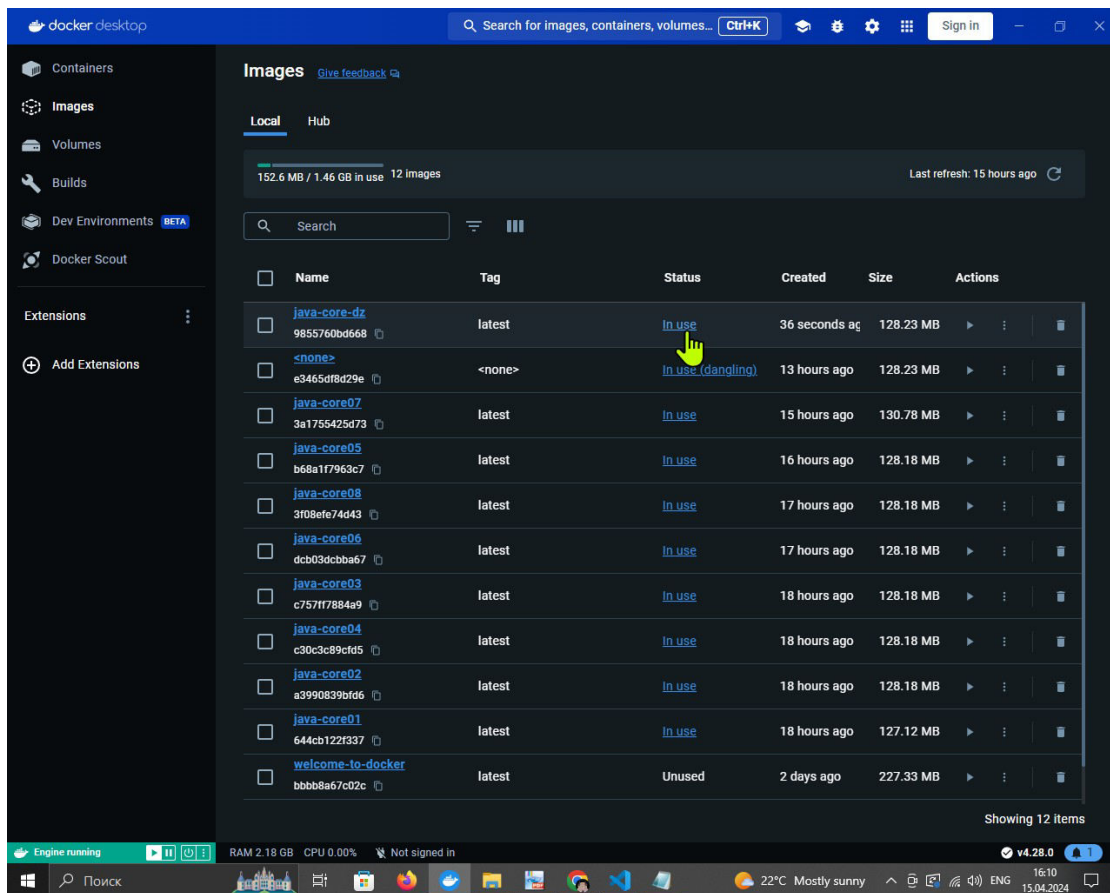
```
* Запускаем Java приложение и открываем страницу javadoc в браузере
CMD ["sh", "-c", "java -classpath /usr/app/out com.simple.Main && cat /usr/app/doc/index-all.html"]
```

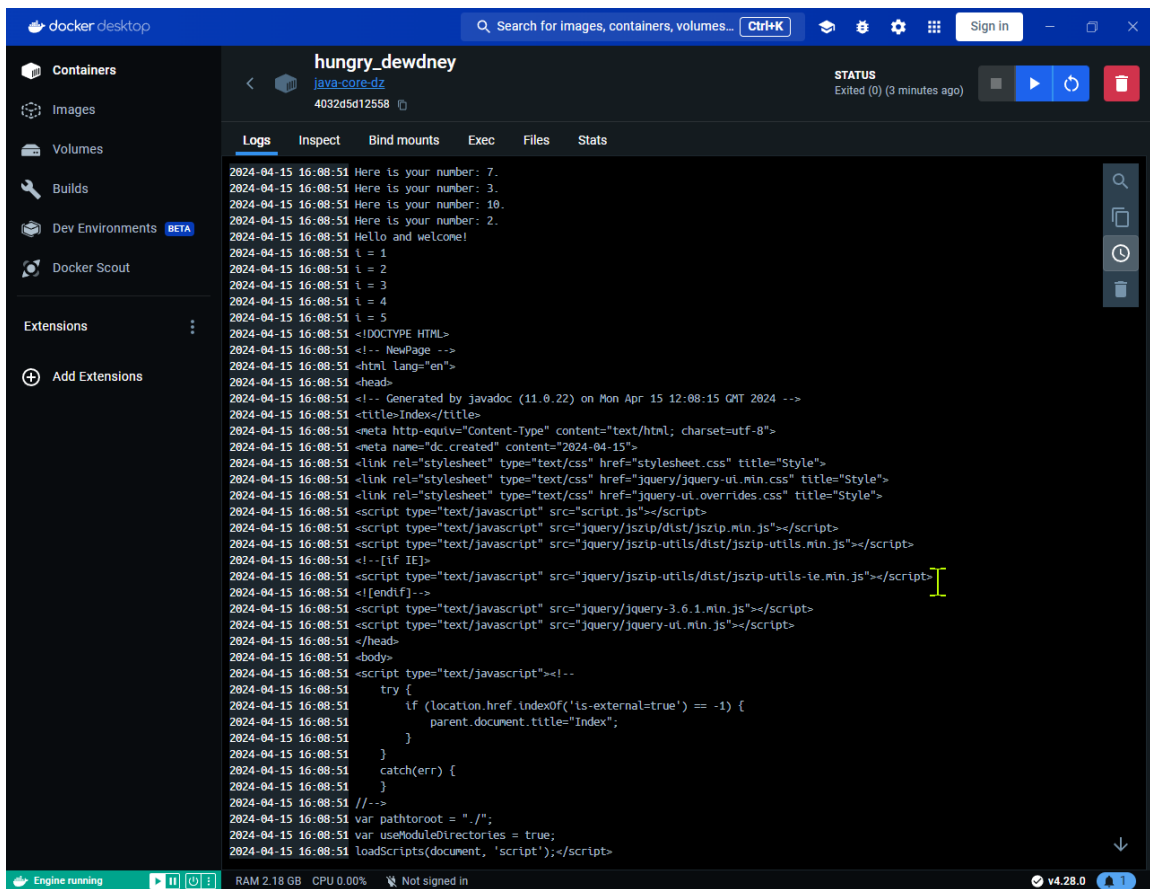
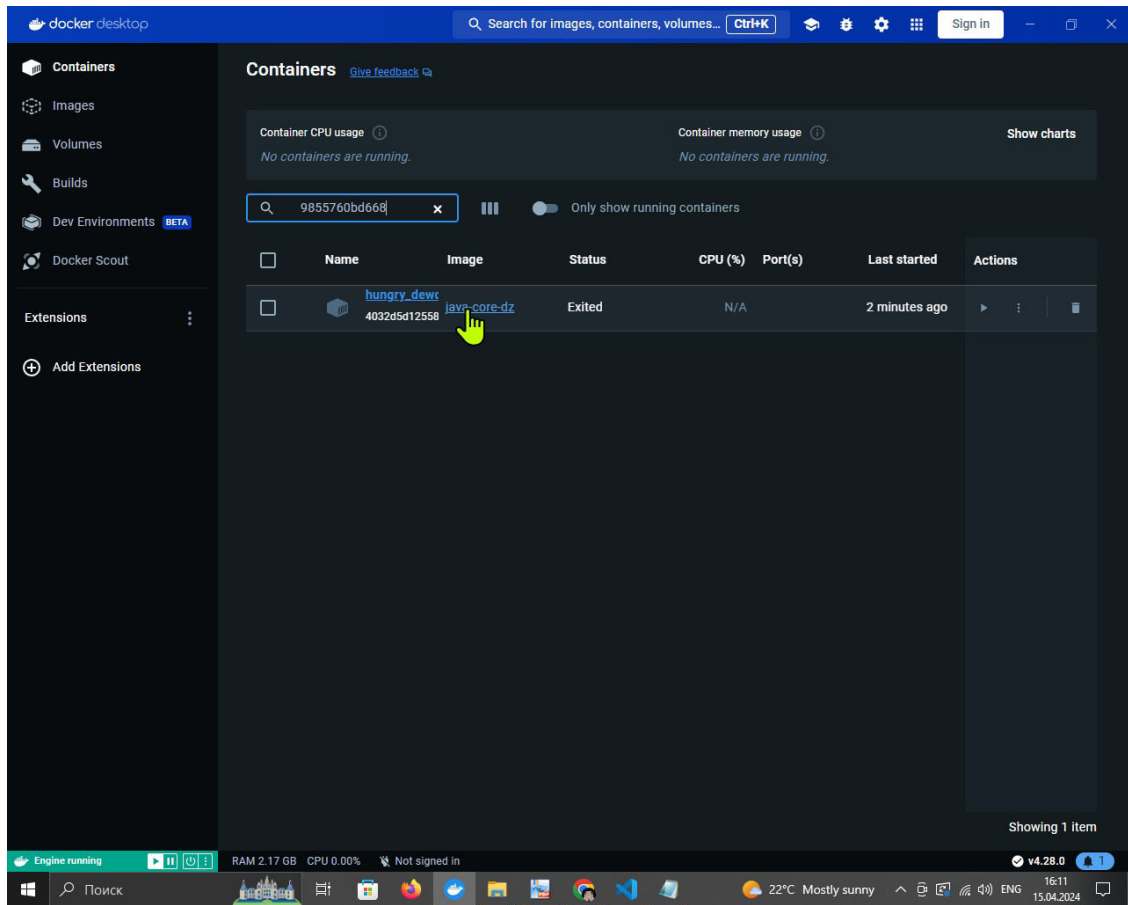


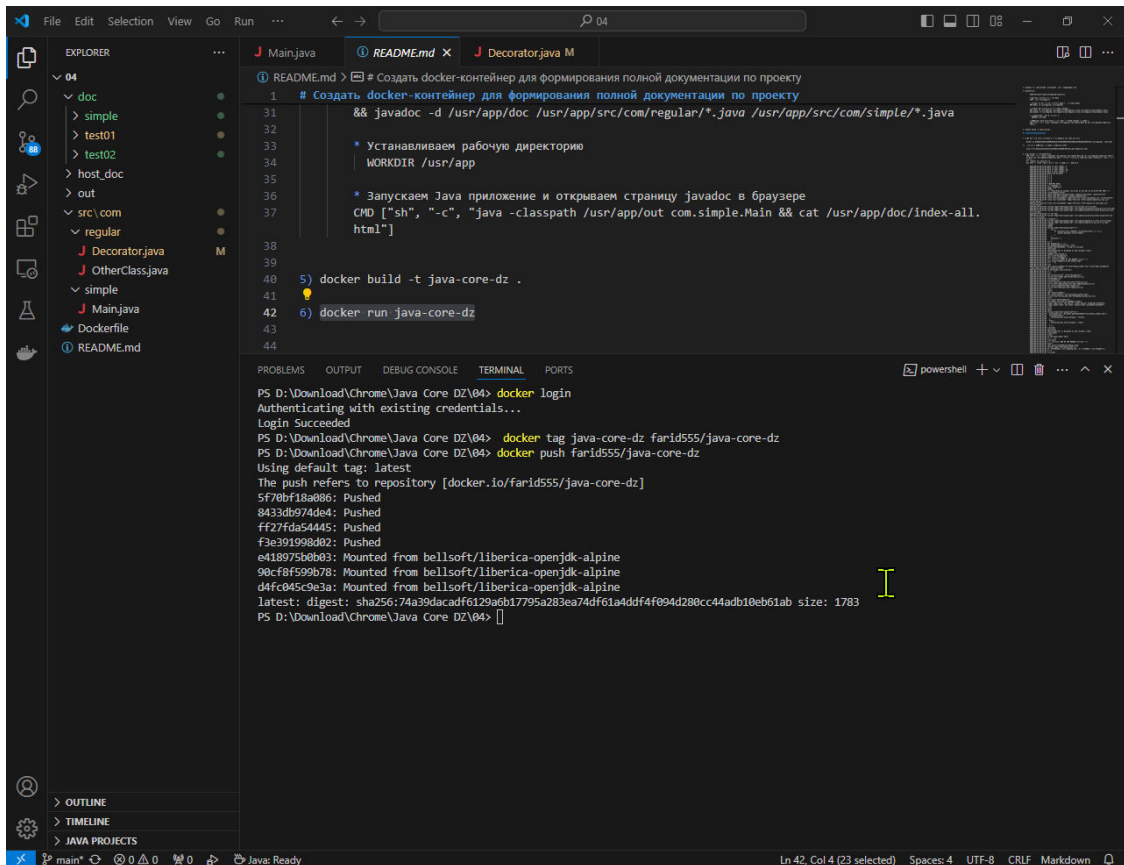
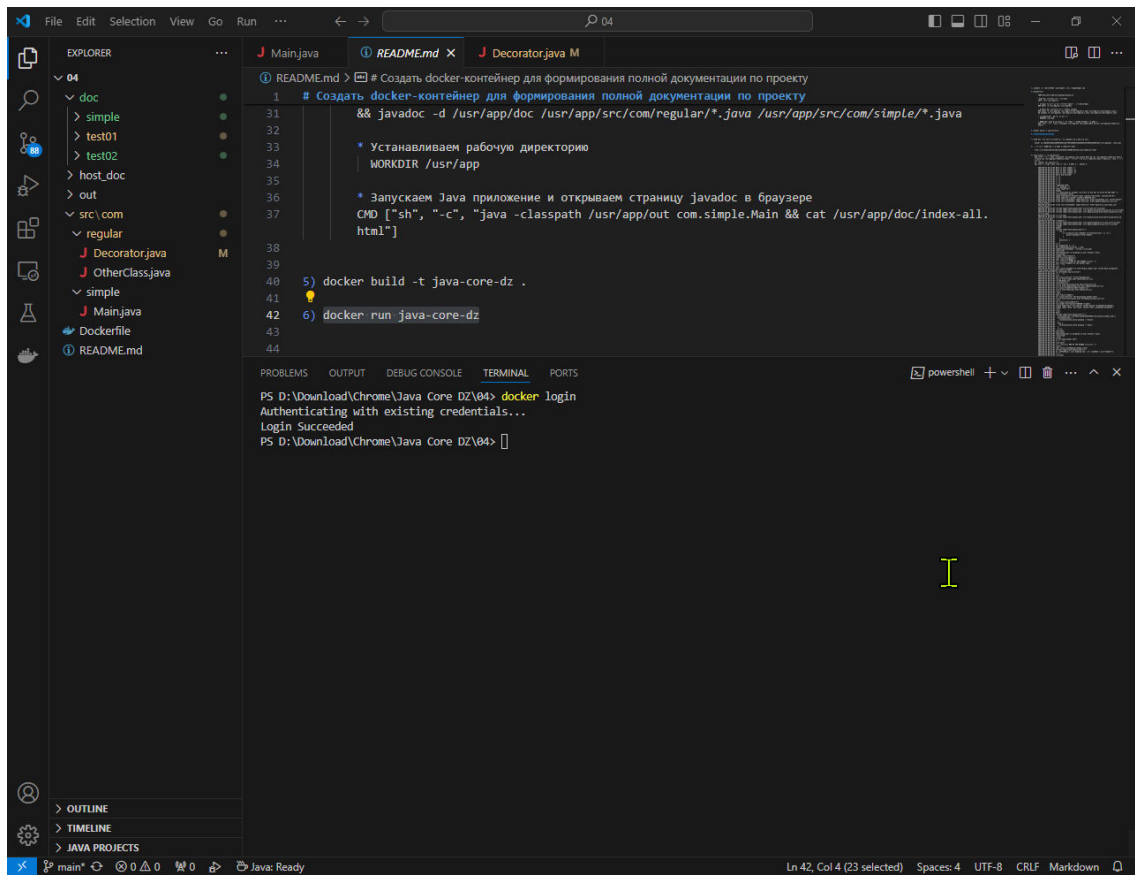
5) docker build -t java-core-dz .



6) docker run java-core-dz





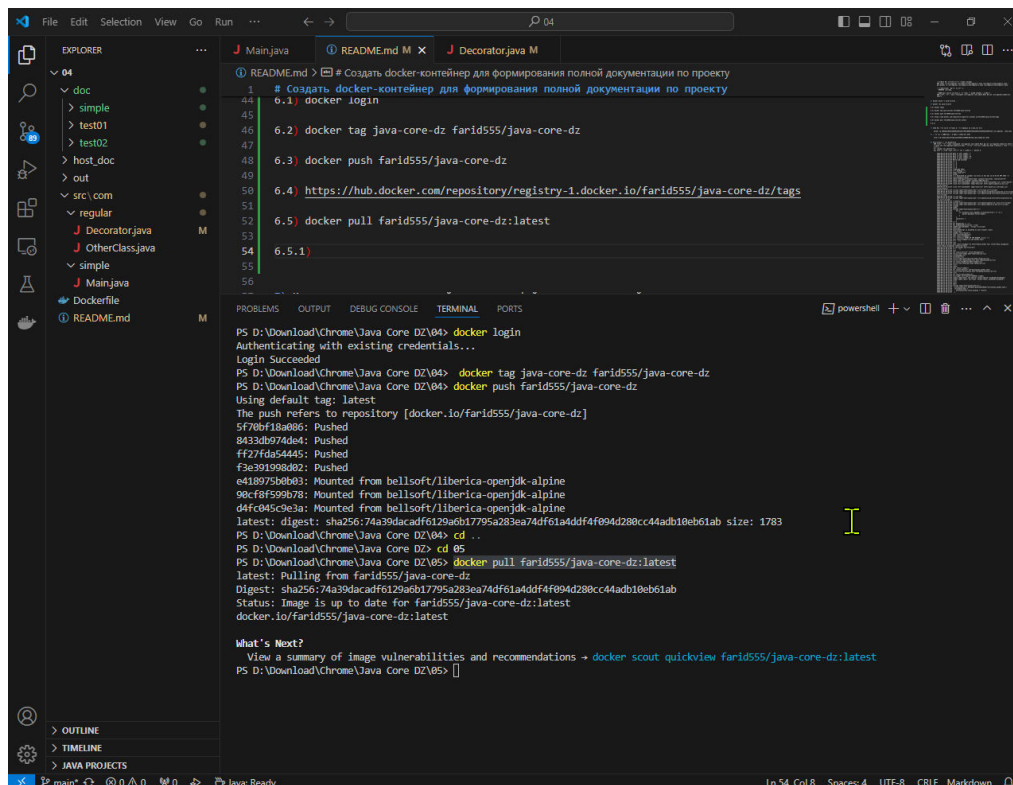
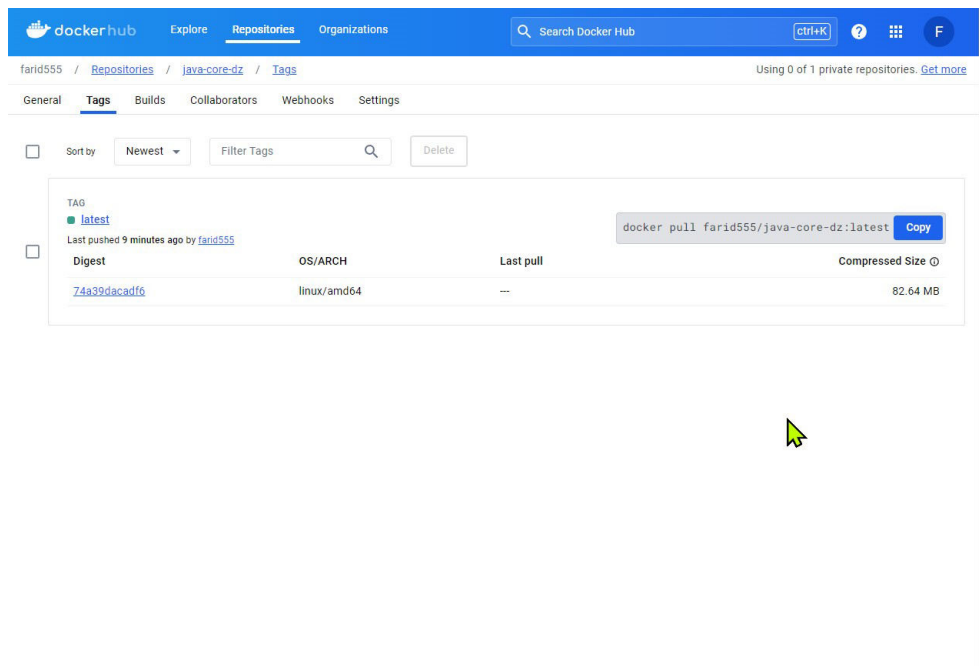


6.1) docker login

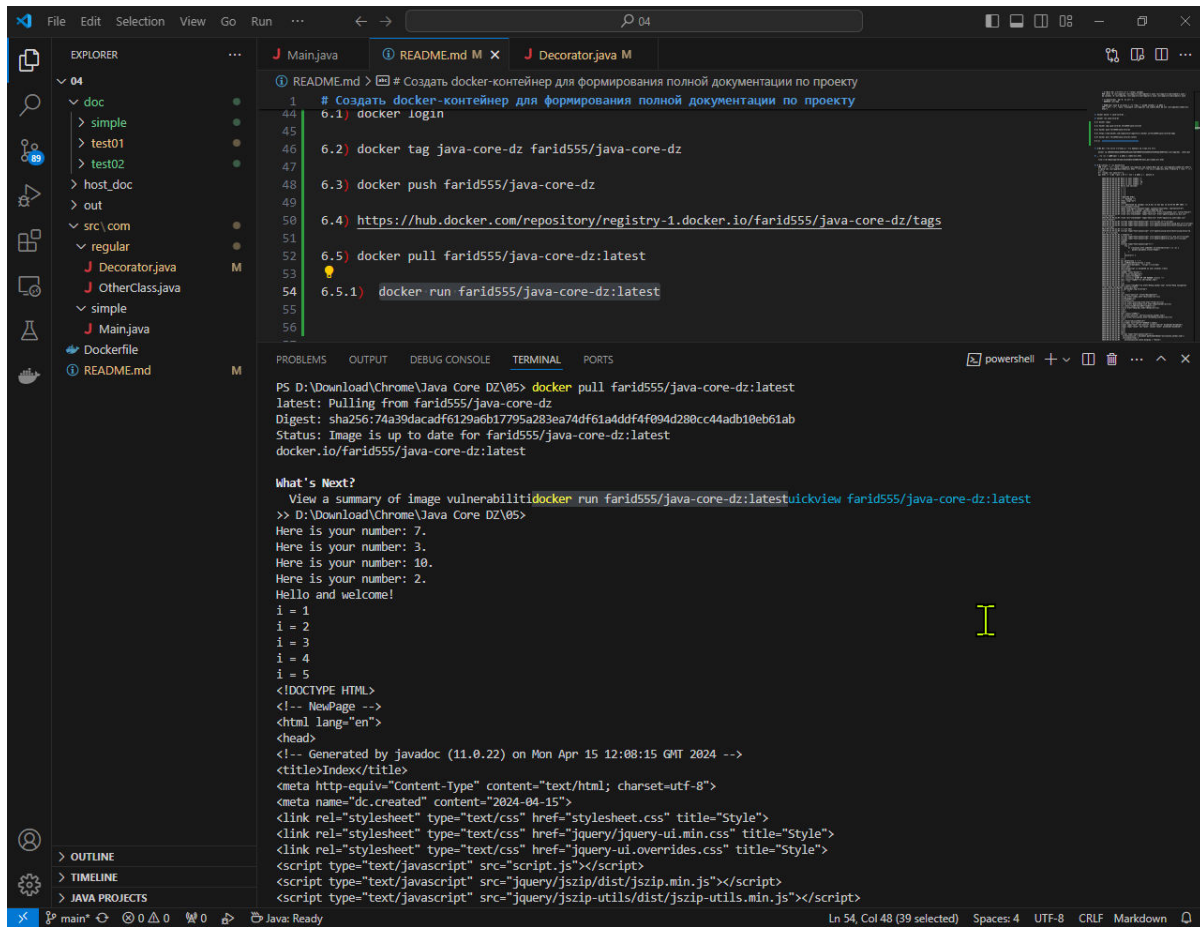
6.2) docker tag java-core-dz farid555/java-core-dz

6.3) docker push farid555/java-core-dz

6.4) <https://hub.docker.com/r/farid555/java-core-dz/tags>



6.5) docker pull farid555/java-core-dz:latest



The screenshot shows an IDE with a project structure on the left, a README file in the center, and a terminal window at the bottom. The README file contains a list of Docker commands for creating and running a container. The terminal window shows the execution of the 'docker pull' command, which successfully pulls the 'farid555/java-core-dz:latest' image from Docker Hub. The output of the command is displayed in the terminal, including the image's digest and status. Below the command output, there is a section titled 'What's Next?' which provides instructions on how to view the image's vulnerability information and how to run the container. The terminal window also shows the output of the 'docker run' command, which displays the container's output, including a list of numbers and a welcome message.

```
1 # Создать docker-контейнер для формирования полной документации по проекту
2
3 6.1) docker login
4
5 6.2) docker tag java-core-dz farid555/java-core-dz
6
7 6.3) docker push farid555/java-core-dz
8
9 6.4) https://hub.docker.com/repository/registry-1.docker.io/farid555/java-core-dz/tags
10
11 6.5) docker pull farid555/java-core-dz:latest
12
13 6.5.1) docker run farid555/java-core-dz:latest
```

```
PS D:\Download\Chrome\Java Core DZ\05> docker pull farid555/java-core-dz:latest
latest: Pulling from farid555/java-core-dz
Digest: sha256:74a39dacdf6129a6b17795a283ea74df61a4ddf4f094d280cc44adb10eb61ab
Status: Image is up to date for farid555/java-core-dz:latest
docker.io/farid555/java-core-dz:latest

What's Next?
View a summary of image vulnerability: docker run farid555/java-core-dz:latest quickview farid555/java-core-dz:latest
>> D:\Download\Chrome\Java Core DZ\05>
Here is your number: 7.
Here is your number: 3.
Here is your number: 10.
Here is your number: 2.
Hello and welcome!
i = 1
i = 2
i = 3
i = 4
i = 5
<!DOCTYPE HTML>
<!-- NewPage -->
<html lang="en">
<head>
<!-- Generated by javadoc (11.0.22) on Mon Apr 15 12:08:15 GMT 2024 -->
<title>Index</title>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
<meta name="dc.created" content="2024-04-15">
<link rel="stylesheet" type="text/css" href="stylesheet.css" title="Style">
<link rel="stylesheet" type="text/css" href="jquery/jquery-ui.min.css" title="Style">
<link rel="stylesheet" type="text/css" href="jquery-ui.overrides.css" title="Style">
<script type="text/javascript" src="script.js"></script>
<script type="text/javascript" src="jquery/jszip/dist/jszip.min.js"></script>
<script type="text/javascript" src="jquery/jszip-utils/dist/jszip-utils.min.js"></script>
```

6.5.1) docker run farid555/java-core-dz:latest - проверяем как контейнер работает :)