

Выполнение практического задания к семинару №1

Задание №1. Создать приложение с вложенностью пакетов не менее 3х, где будет класс для входа и несколько классов с логикой. Пример: приложение для внесения заметок во внешний файл с обязательной фиксацией времени

пример:

Введите заметку: Hello, world!

Дозапись в файл: 16.07.2023 -> Hello, world

Скомпилируйте и запустите посредством CLI

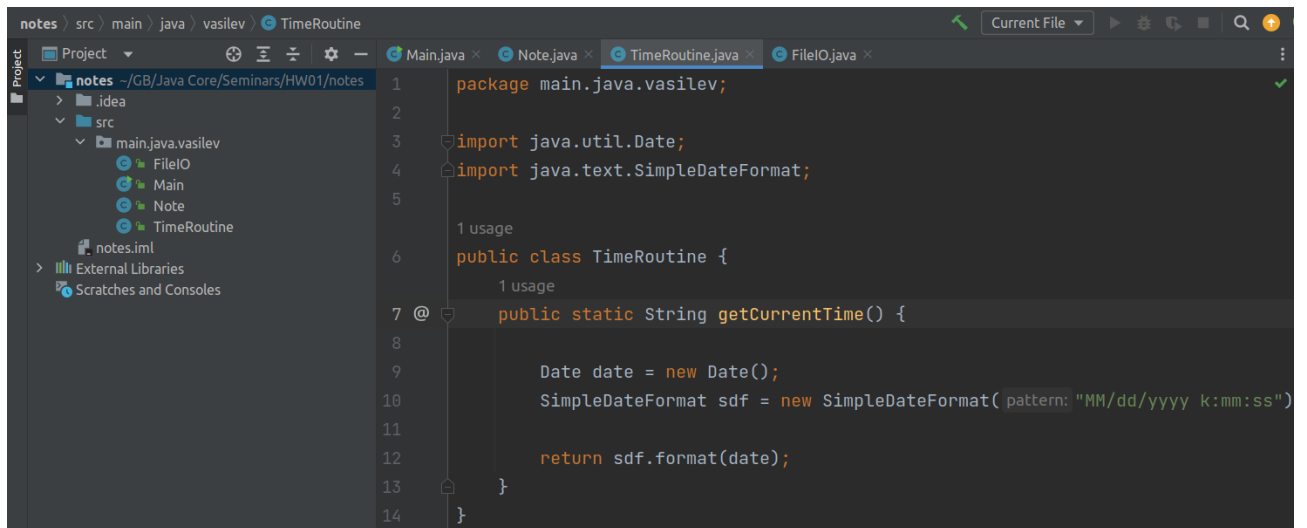
Выполнение:

а) Создан проект **notes**, в нем 4 класса в директории **/src/main/java/vasilev/**:

- **Main.java** – точка входа

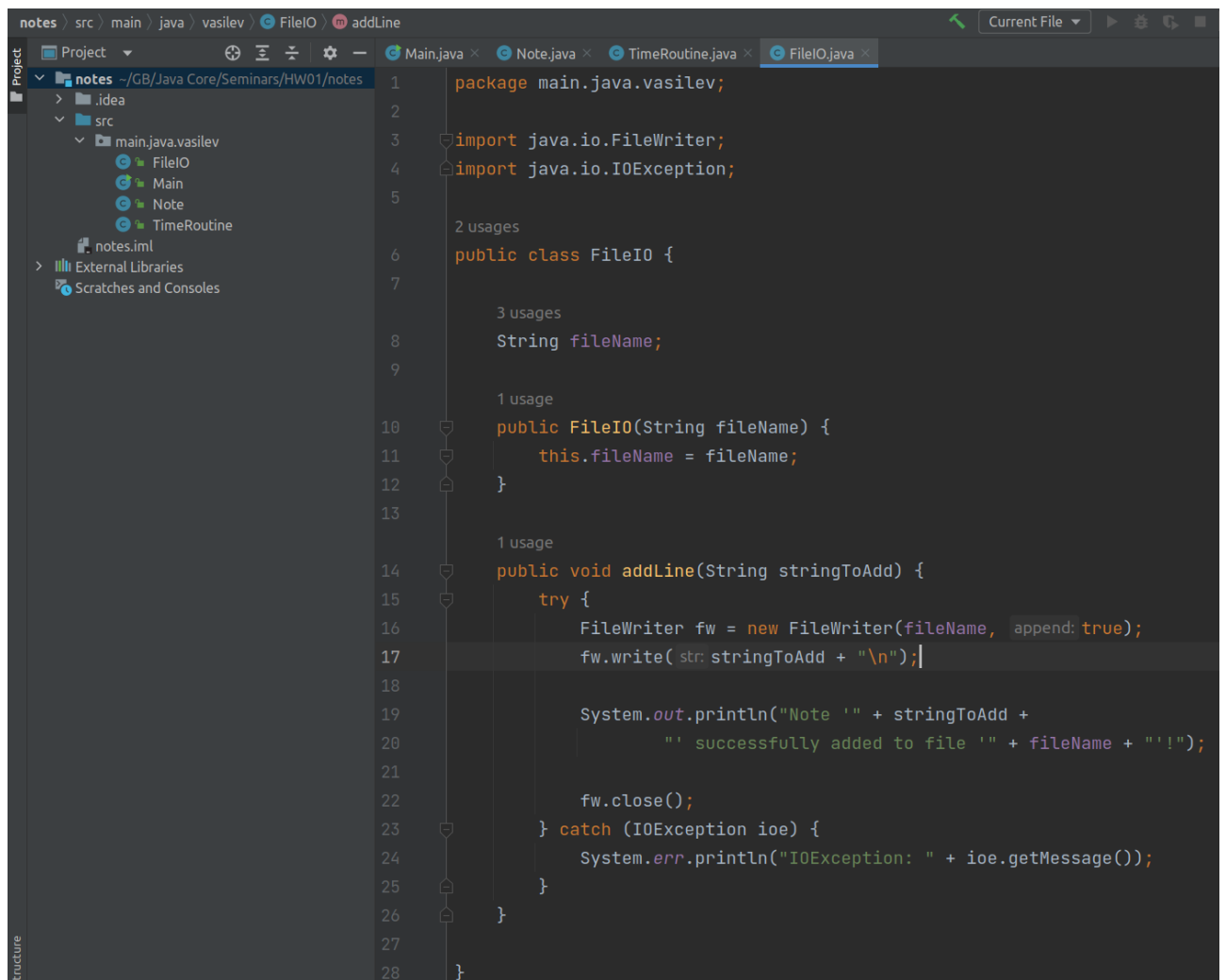
- **Note.java** – работа с заметками

- **TimeRoutine.java** - учет времени



```
1 package main.java.vasilev;
2
3 import java.util.Date;
4 import java.text.SimpleDateFormat;
5
6 1 usage
7 public class TimeRoutine {
8     1 usage
9     public static String getCurrentTime() {
10
11         Date date = new Date();
12         SimpleDateFormat sdf = new SimpleDateFormat(pattern: "MM/dd/yyyy k:mm:ss")
13
14         return sdf.format(date);
15     }
16 }
```

- **FileIO.java** - работа с файлами

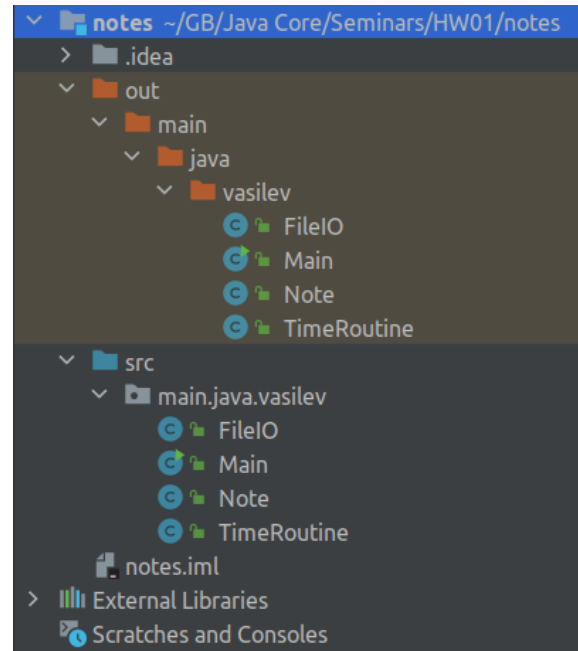


```
1 package main.java.vasilev;
2
3 import java.io.FileWriter;
4 import java.io.IOException;
5
6 2 usages
7 public class FileIO {
8
9     3 usages
10     String fileName;
11
12     1 usage
13     public FileIO(String fileName) {
14         this.fileName = fileName;
15     }
16
17     1 usage
18     public void addLine(String stringToAdd) {
19         try {
20             FileWriter fw = new FileWriter(fileName, append: true);
21             fw.write(str: stringToAdd + "\n");
22
23             System.out.println("Note '" + stringToAdd +
24                 "' successfully added to file '" + fileName + "'!");
25
26             fw.close();
27         } catch (IOException ioe) {
28             System.err.println("IOException: " + ioe.getMessage());
29         }
30     }
31 }
```

6) Компиляция проекта **notes**:

```
boris@lm-boris:~/GB/Java Core/Seminars/HW01/notes$ javac -sourcepath ./src/ -d ./out/ ./src/main/java/vasilev/Main.java
boris@lm-boris:~/GB/Java Core/Seminars/HW01/notes$
```

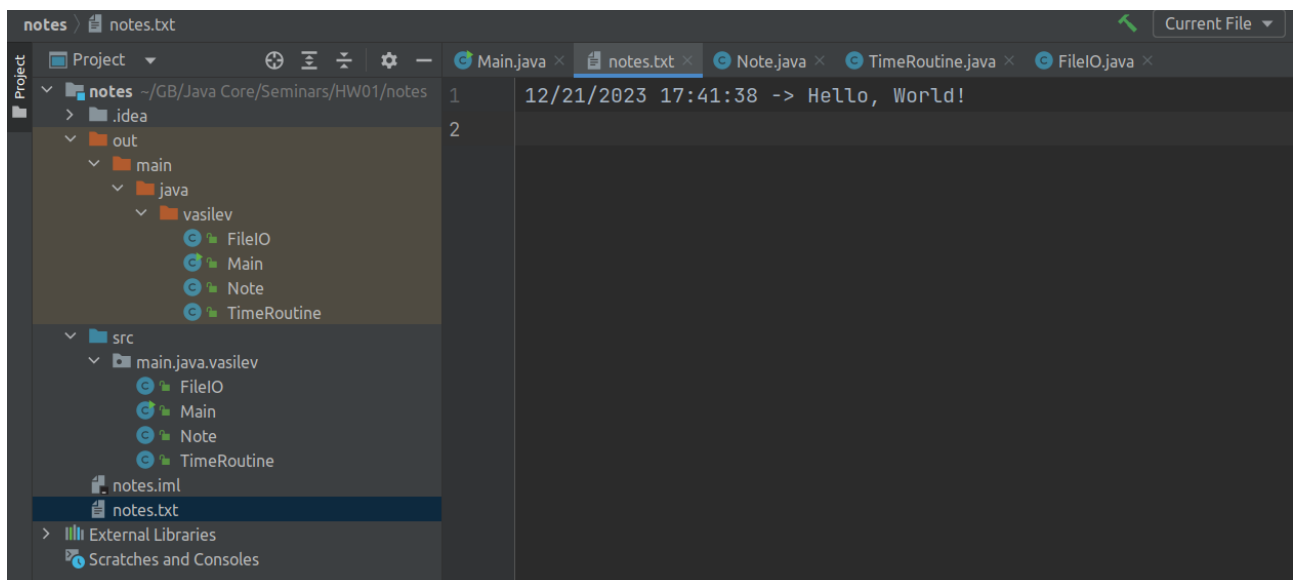
В директории проекта появилась папка **out**, структурой повторяющая иерархию размещения классов в проекте и содержащая байт-коды классов:



в) запуск проекта:

```
boris@lm-boris:~/GB/Java Core/Seminars/HW01/notes$ java -classpath ./out/ main.java.vasilev.Main
Your new note: Hello, World!
Note '12/21/2023 17:41:38 -> Hello, World!' successfully added to file 'notes.txt'!
boris@lm-boris:~/GB/Java Core/Seminars/HW01/notes$
```

Программа отработала корректно, пользователем создана новая заметка **“Hello, World!”**, к которой добавлено время ее создания; информация добавлена в файл **“notes.txt”**:



Задание №2. Создать два Docker-образа. Один должен компилировать Java-проект обратно в папку на компьютере пользователя, а второй забирать скомпилированные классы и исполнять их. Пример листинга для docker-compose приведен в презентации семинара

Выполнение:

а) созданы **yaml-файлы** для контейнеров компиляции и запуска проекта (из задания №1)

```
boris@lm-boris: ~/GB/Java Core/Seminars/HW01/notes
File Edit View Search Terminal Help
GNU nano 6.2 docker-compose-classes.yml
version: '3'

services:
  app:
    image: bellsoft/liberica-openjdk-alpine:11.0.16.1-1
    command: javac -sourcepath /app/src/ -d /app/out/ /app/src/main/java/vasilev/Main.java
    volumes:
      - ./out_dc:/app/out
      - ./src:/app/src
```

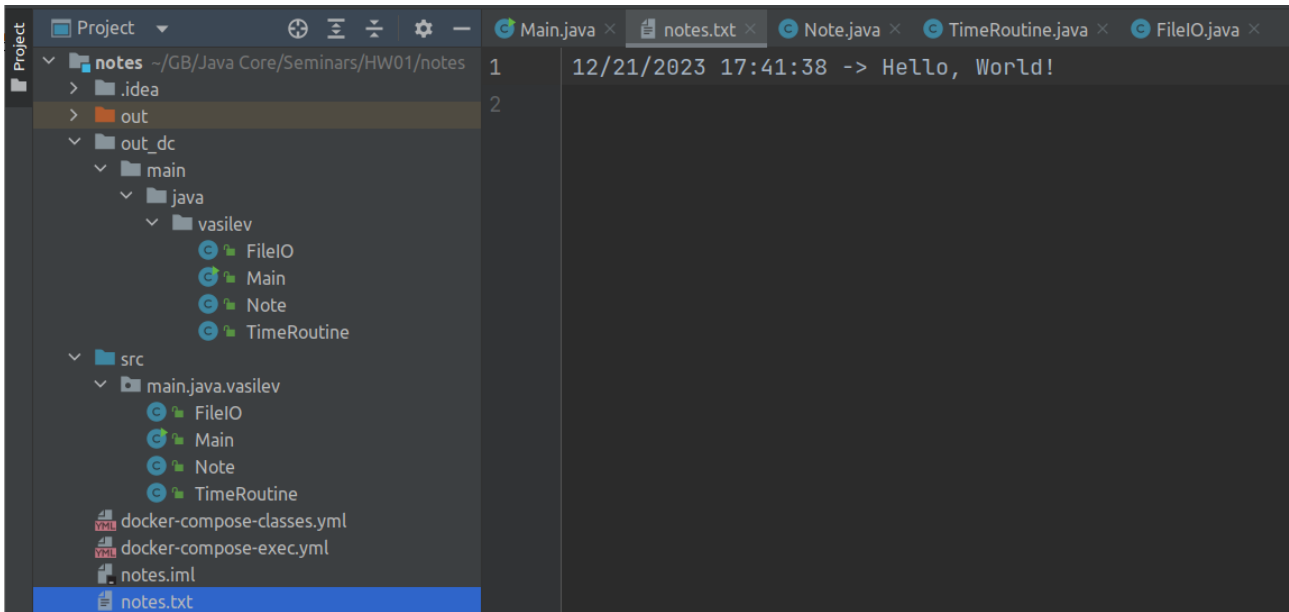
```
boris@lm-boris: ~/GB/Java Core/Seminars/HW01/notes
File Edit View Search Terminal Help
GNU nano 6.2 docker-compose-exec.yml
version: '3'

services:
  app:
    image: bellsoft/liberica-openjdk-alpine:11.0.16.1-1
    command: java -classpath /app/out main.java.vasilev.Main
    volumes:
      - ./out_dc:/app/out/
```

б) создан и запущен контейнер компиляции проекта

```
boris@lm-boris:~/GB/Java Core/Seminars/HW01/notes$ sudo docker-compose -f docker-compose-classes.yml up
Creating network "notes_default" with the default driver
Pulling app (bellsoft/liberica-openjdk-alpine:11.0.16.1-1)...
11.0.16.1-1: Pulling from bellsoft/liberica-openjdk-alpine
213ec9aee27d: Pull complete
f3b60082a773: Pull complete
85e7daf60581: Pull complete
Digest: sha256:050e6e620309772b4e886de339caf782f2e15dace820f4406749160410b7e290
Status: Downloaded newer image for bellsoft/liberica-openjdk-alpine:11.0.16.1-1
Creating notes_app_1 ... done
Attaching to notes_app_1
notes_app_1 exited with code 0
boris@lm-boris:~/GB/Java Core/Seminars/HW01/notes$
```

в) В соответствии с конфигурацией из файла **docker-compose-classes.yml** в результате работы контейнера создана директория **out_dc** с байт-кодами классов.



г) создан и запущен контейнер запуска программы:

```
boris@lm-boris:~/GB/Java Core/Seminars/HW01/notes$ sudo docker-compose -f docker-compose-exec.yml up
Recreating notes_app_1 ... done

Attaching to notes_app_1
app_1 | Your new note: Note '12/21/2023 18:33:22 -> Empty message line...' successfully added to file 'notes.txt'!
notes_app_1 exited with code 0
boris@lm-boris:~/GB/Java Core/Seminars/HW01/notes$
```

Программа успешно запускается в контейнере (что видно из вывода), однако так и не удалось реализовать в контейнере консольный ввод нового сообщения пользователя. Экземпляр класса Scanner, используемый для считывания строки, возвращает ошибку **java.util.NoSuchElementException: No line found**. Проект не отличается от использованного в задании №1 выше, там все работает корректно.

Вывод результат работы контейнера запуска (без заглушки для пустой строки):

```
boris@lm-boris:~/GB/Java Core/Seminars/HW01/notes$ sudo docker-compose -f docker-compose-exec.yml up
Creating notes_app_1 ... done

Attaching to notes_app_1
app_1 | Your new note: Exception in thread "main" java.util.NoSuchElementException: No line found
app_1 |     at java.base/java.util.Scanner.nextLine(Scanner.java:1651)
app_1 |     at main.java.vasilev.Note.getNote(Note.java:10)
app_1 |     at main.java.vasilev.Main.main(Main.java:10)
notes_app_1 exited with code 1
```