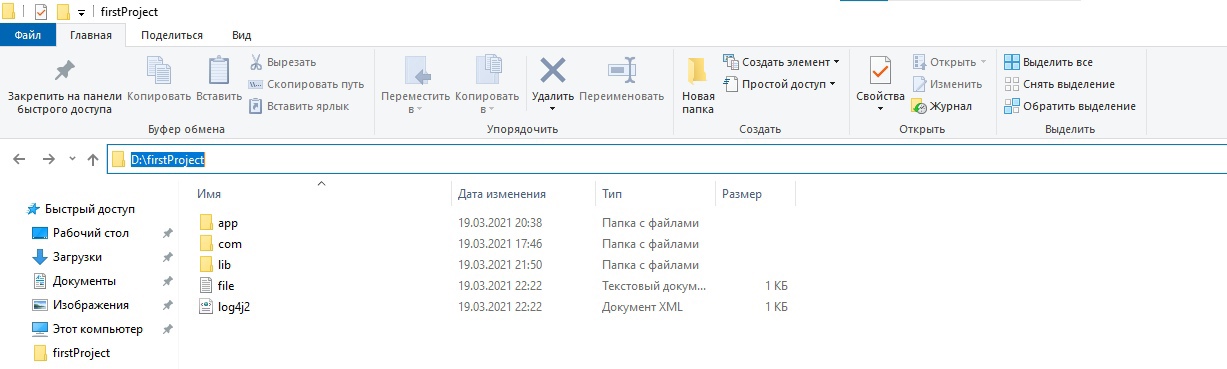
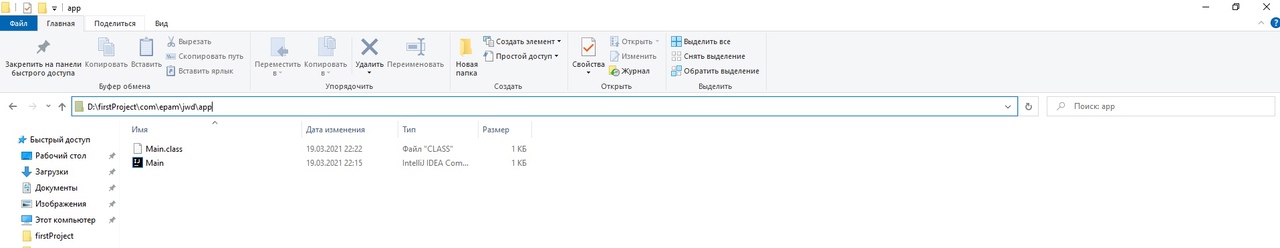
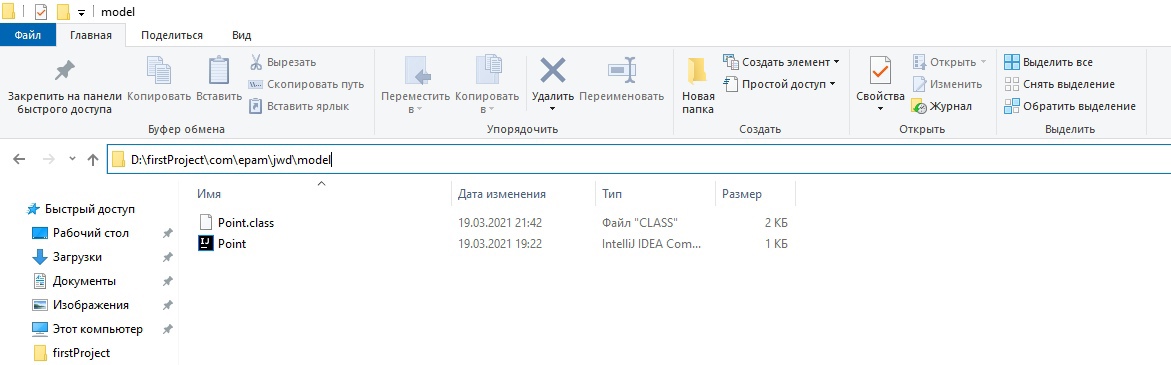
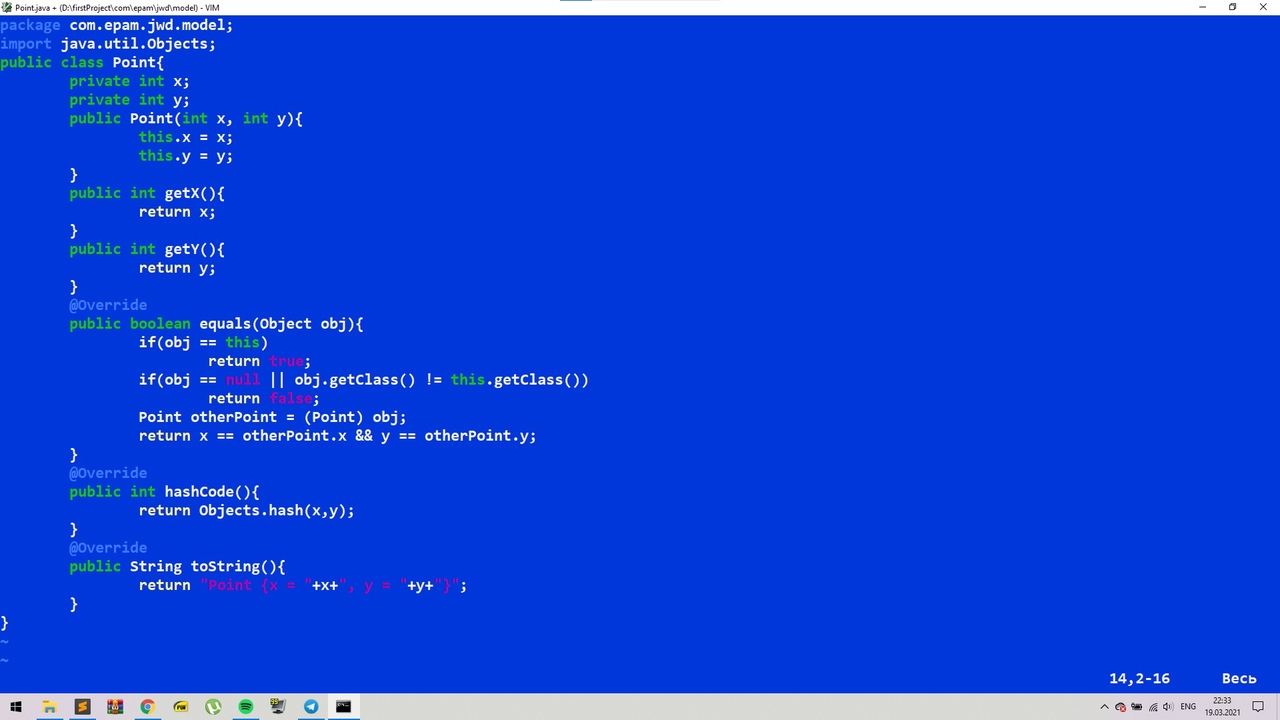
**Филютич Рома**

1. Создать два пакета com.epam.jwd.app и com.jwd.model.
   1. В пакете model положить простейшую модель (любой POJO класс с надлежащей инкапсуляцией),
   2. в пакете app разместить Main класс с простейшей логикой ( сложить точки в массив, в for вызвать на каждой toString() и записать в лог)

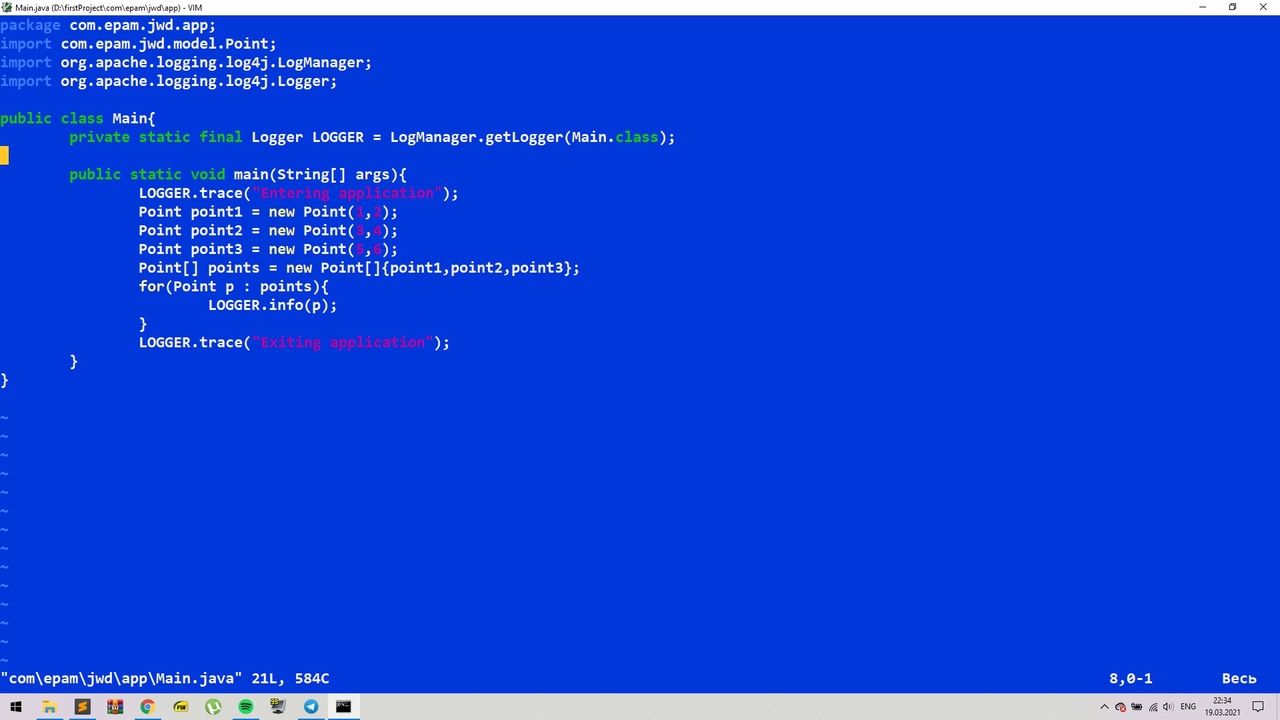
Корневая папка 

Папка com/epam/jwd/app

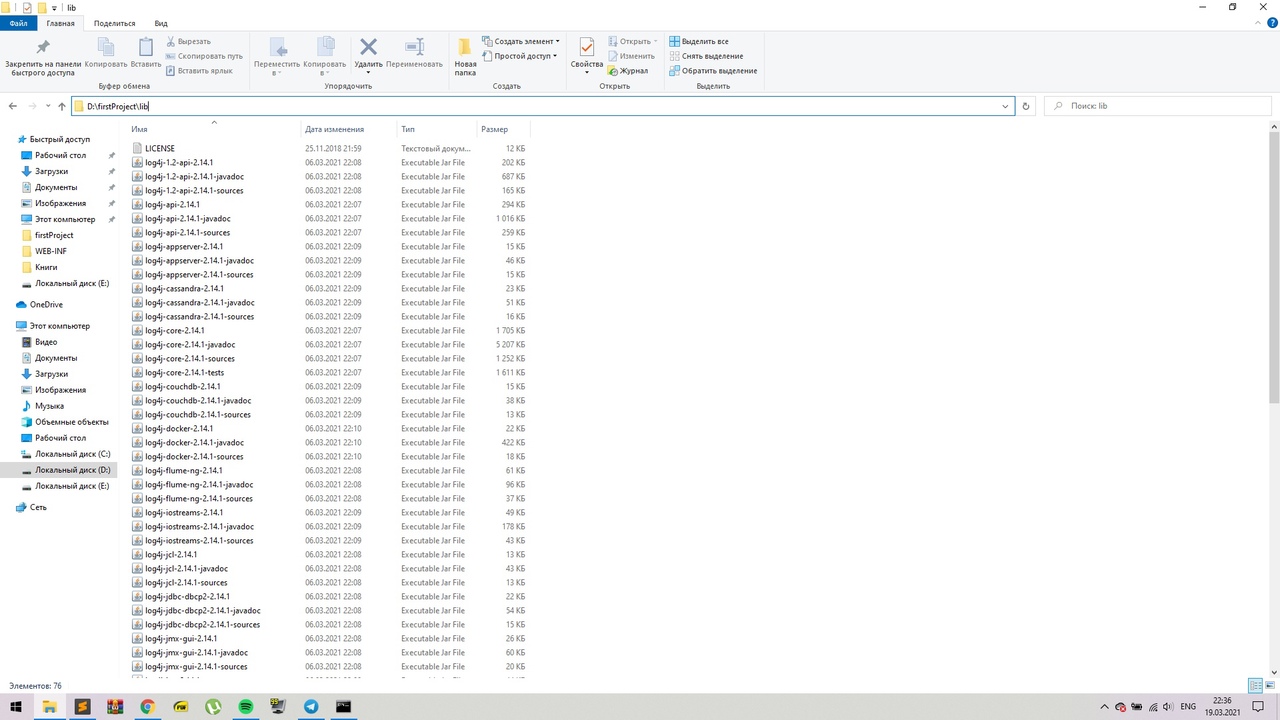
Папка com/epam/jwd/model

Pojo класс Point vim 

Главный класс Main vim



1. Все log-библиотеки разместить в виде jar-контейнеров в папке lib в корне проекта

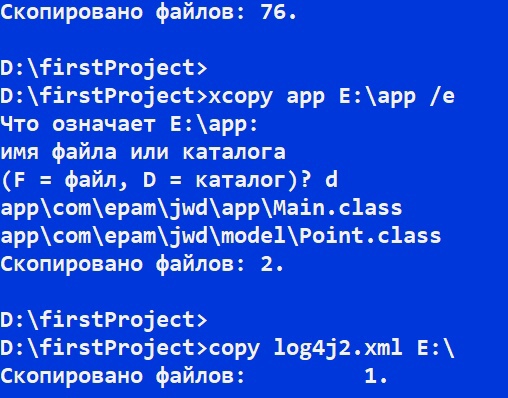
Библиотека log4j2 в папке lib в корне проекта 

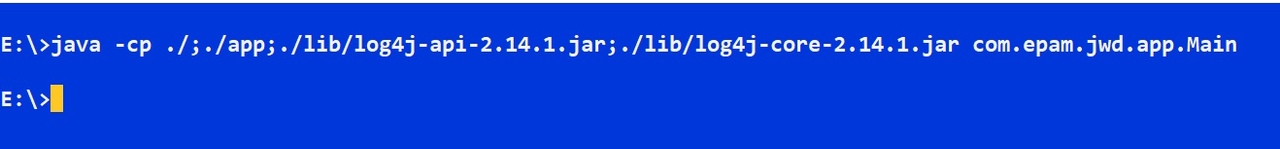
1. Скомпилировать приложение против библиотек из папки lib
2. положить скомпилированные .class файлы в папку /app в корне проекта

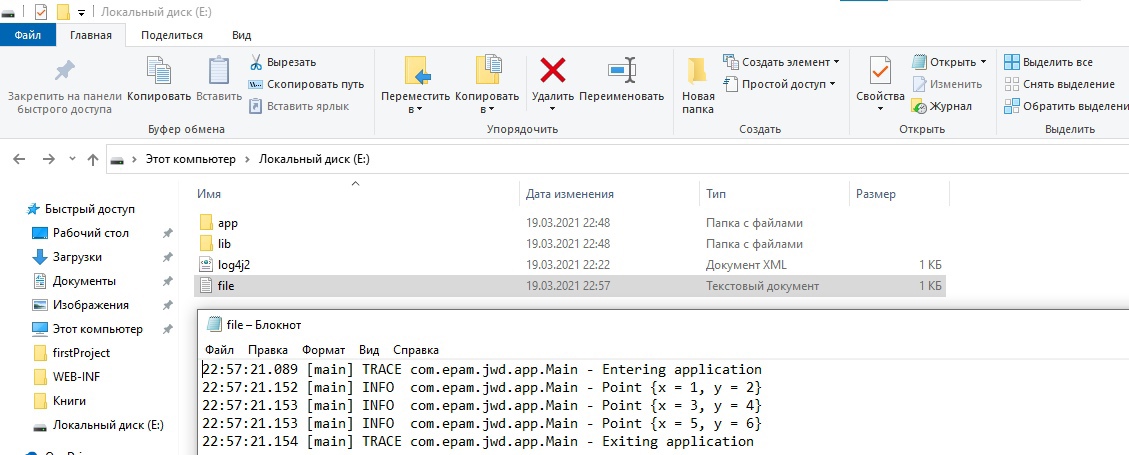


1. Папки /lib и /app скопировать в другое место и запустить используя консоль

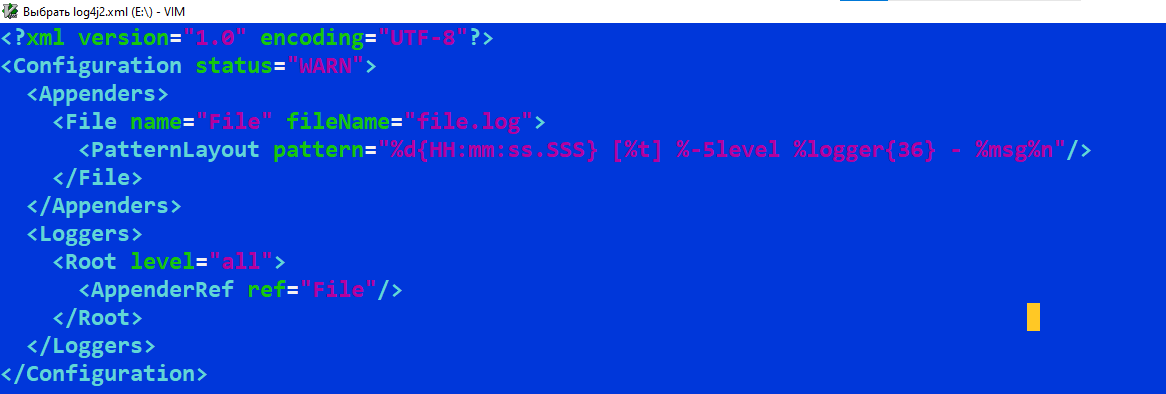
Копирование папок на диск E:



Запуск приложения с консоли

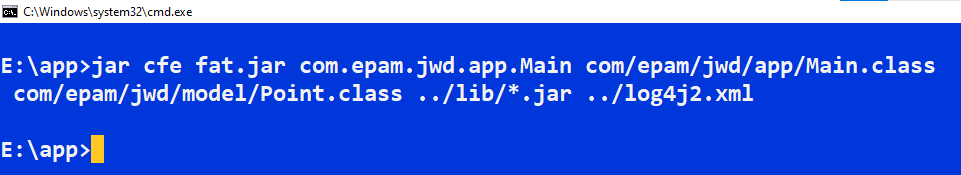
Результат работы программы ( Залогированны строки toString())

Файл конфигурации log4j2.xml в vim



1. (Optional) Скомпилировать приложение в “fat.jar”, которая будет включать в себя как скомпилированные классы приложения, так и jar библиотеки

Создание jar-файла



Результат создания jar-файла

