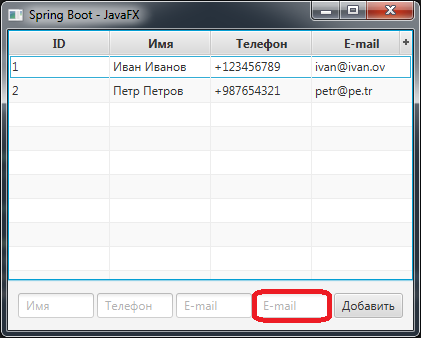
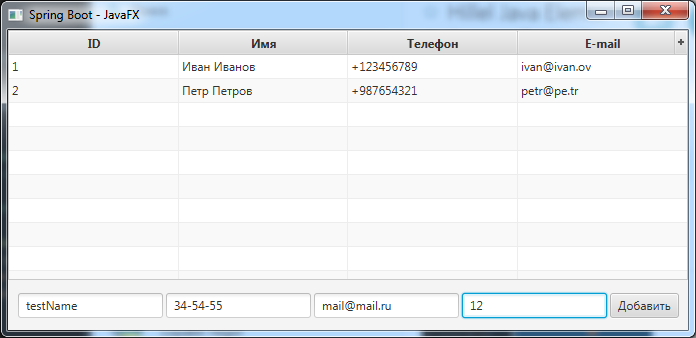
1. Исправить некорректный **label** для **age** параметра

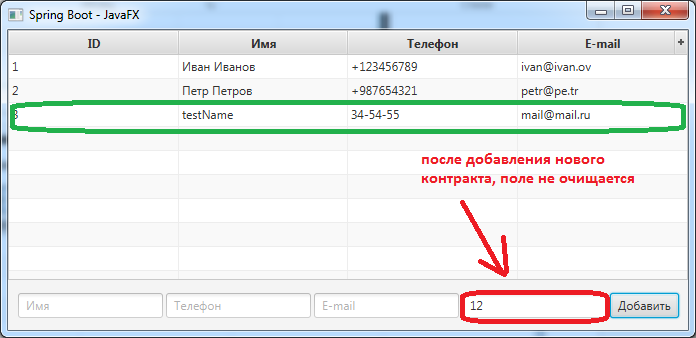
**actual** -> E-mail

**expected** -> Age

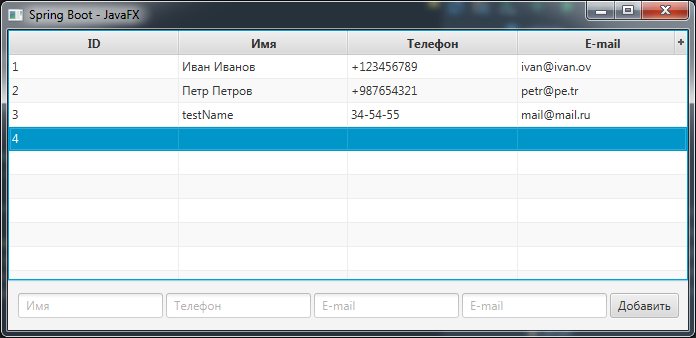


1. После добавления нового контракта - поле **age** не очищается





1. На текущий момент, пользуясь данной программкой - есть возможность добавить запись без каких-либо параметров: в случае, если Вы оставите поля пустыми и нажмете «Добавить», - на стороне бекенда валится exception следующего вида



Caused by: java.lang.reflect.InvocationTargetException

at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)

at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:62)

at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)

at java.lang.reflect.Method.invoke(Method.java:497)

at sun.reflect.misc.Trampoline.invoke(MethodUtil.java:71)

at sun.reflect.GeneratedMethodAccessor10.invoke(Unknown Source)

at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)

at java.lang.reflect.Method.invoke(Method.java:497)

at sun.reflect.misc.MethodUtil.invoke(MethodUtil.java:275)

at javafx.fxml.FXMLLoader$MethodHandler.invoke(FXMLLoader.java:1771)

... 48 more

Caused by: org.springframework.dao.DataIntegrityViolationException: could not execute statement; SQL [n/a]; constraint ["UK\_GNQWBWWCN7X0M5JLT4158DASS\_INDEX\_6 ON PUBLIC.CONTACT(EMAIL) VALUES ('', 4)"; SQL statement:

insert into contact (id, age, email, name, phone) values (null, ?, ?, ?, ?) [23505-187]]; nested exception is org.hibernate.exception.ConstraintViolationException: could not execute statement

**actual behavior** -> exception

**expected** -> как вариант

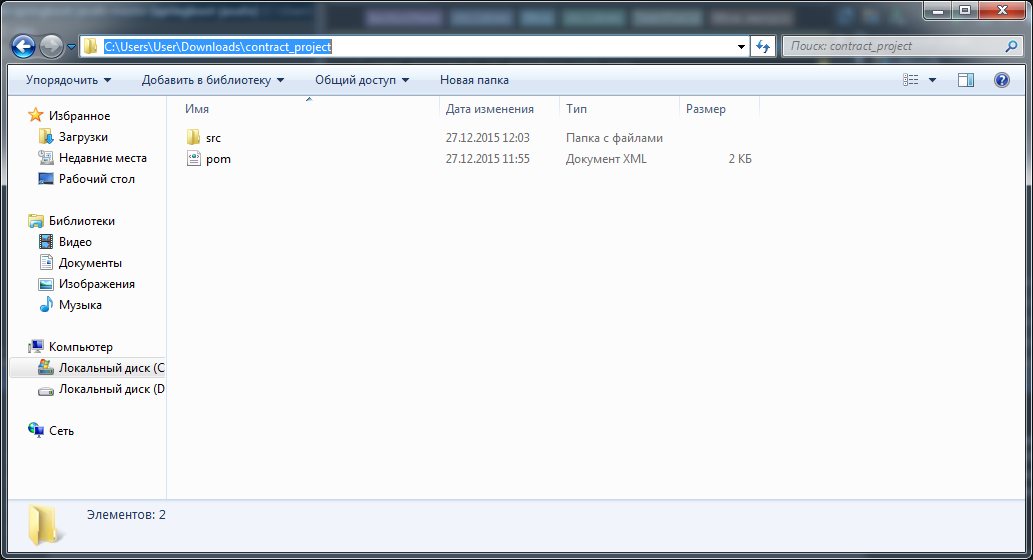
* Можно disabled button в случае, если поля не заполнены
* Можно проставить некий if и не сохранять просто подобные сущности(пустые контракты, контракт без параметров)
* Придумать какой-либо свой вариант)))

1. Сделать проверку на тип данных для поля возраст(**age** поле), в случае, если пользователь ввел вместо чисел – буквы, - контракт не сохранять

**Для запуска приложения Вы можете воспользоваться как минимум 2-мя вариантами, и 1-ый и 2-ой вариант первоочередными степами предполагает импорт проекта в IDE; Поэтому, давайте проделаем общий флоу и для 1-го и 2-го случаев, и уже после рассмотрим особенности запуска приложения обоими способами:**

**Напомню, что 1-ый общий степ – это импорт проекта в IDE >> то-есть**

1. Сделать clone remote repo
2. Import project в **IDE**

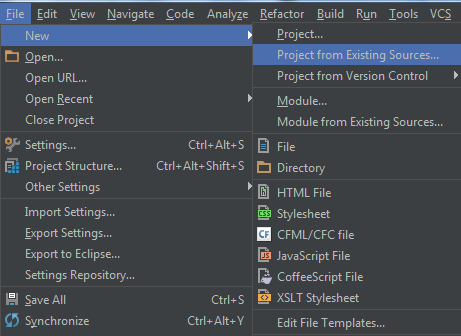


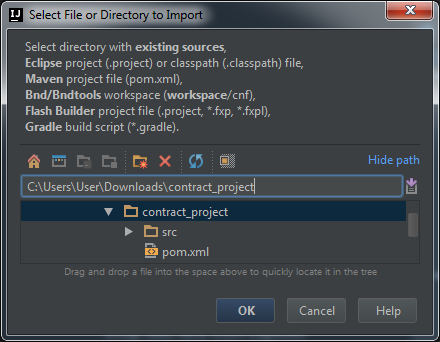
Наш проект

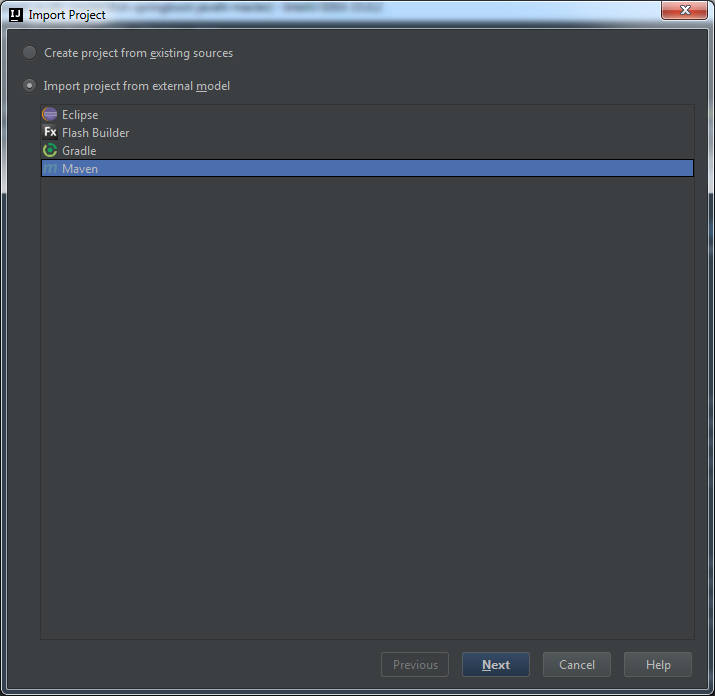
1. Запускаем IDE

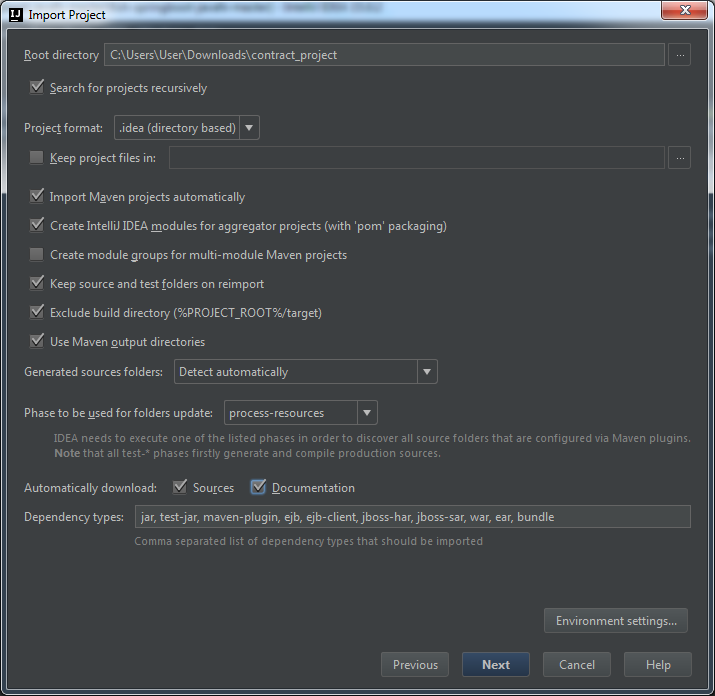


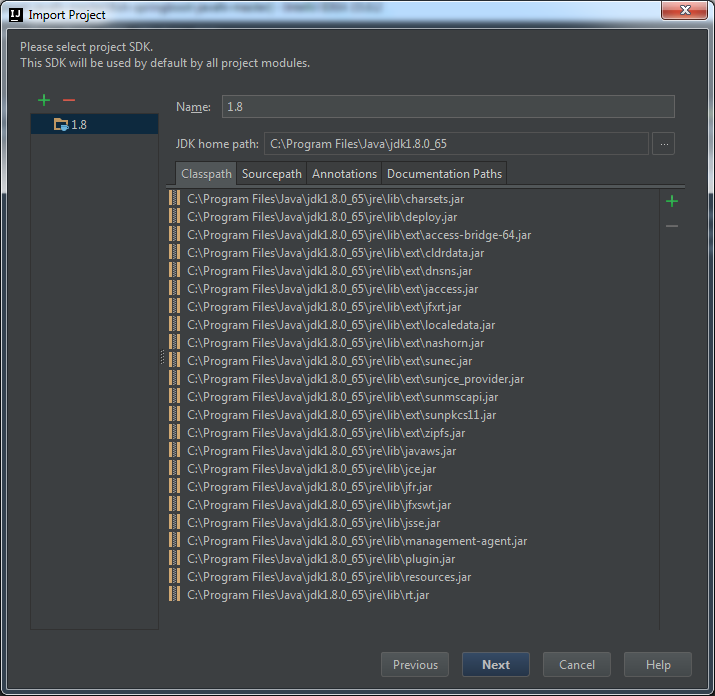
Производим import проекта

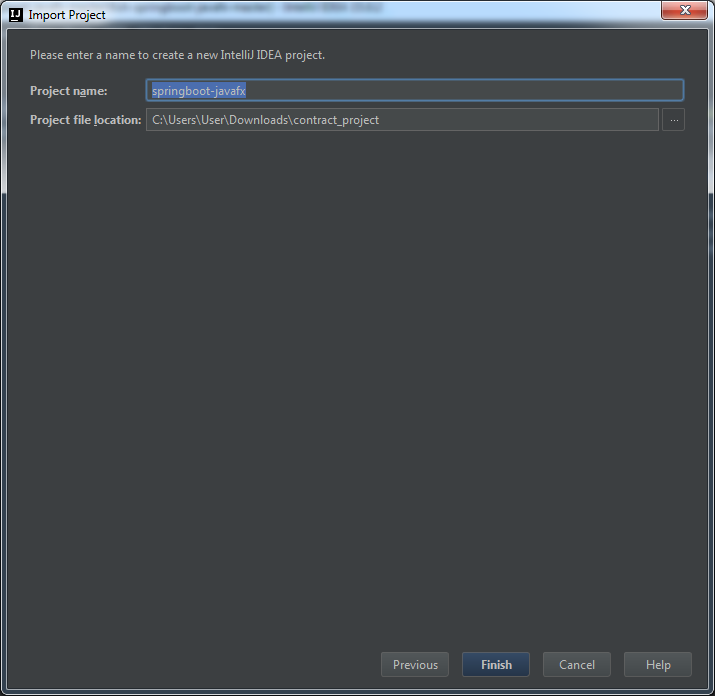






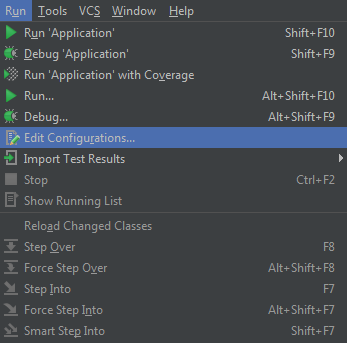






Создадим конфигурацию

Run -> Edit configuration



Нажимаем на **«+»**

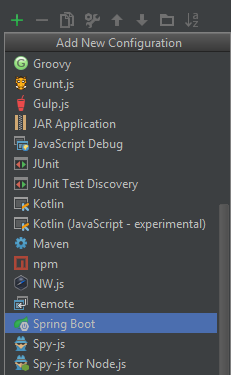


**С этого пункта давайте рассмотрим сначала запуск приложения 1-ым вариантом, чуть ниже (для практики) – 2-ым вариантом;**

**1 ВАРИАНТ:**

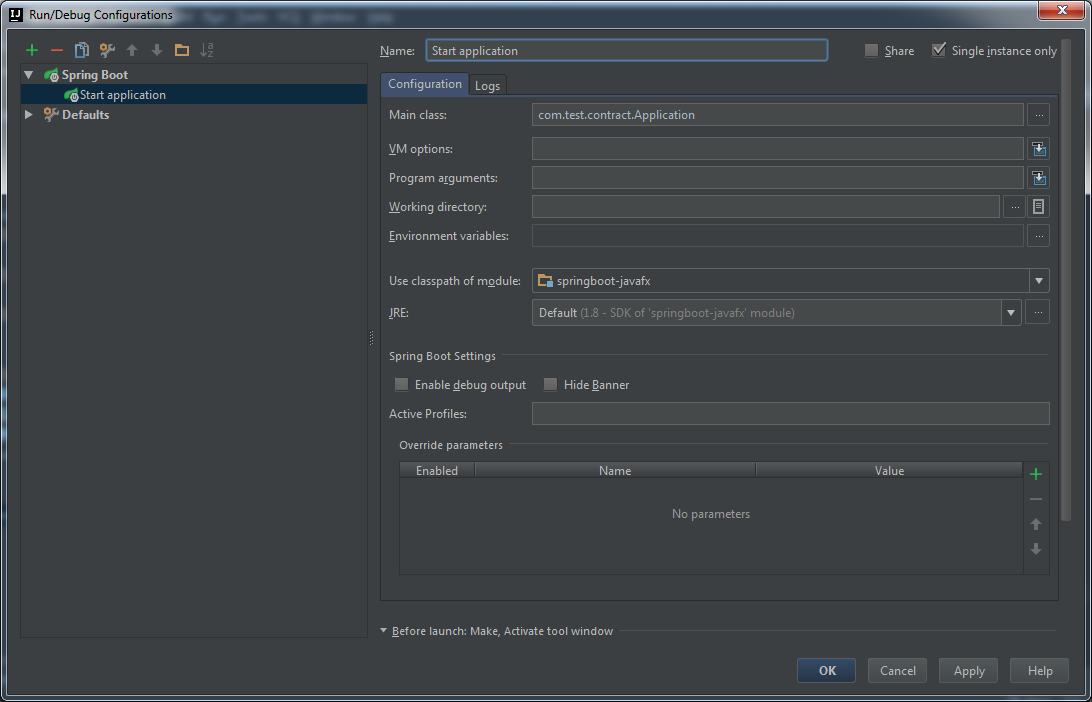
ЗАПУСК ПОСРЕДСТВОМ ИСПОЛЬЗОВАНИЯ **SPRINGBOOT**

Выбираем **«Spring Boot»**



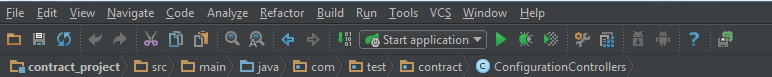
В качестве

**Main class** -> com.test.contract.Application

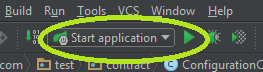


**Apply -> ok (BUTTONS)**

1. Запуск приложения

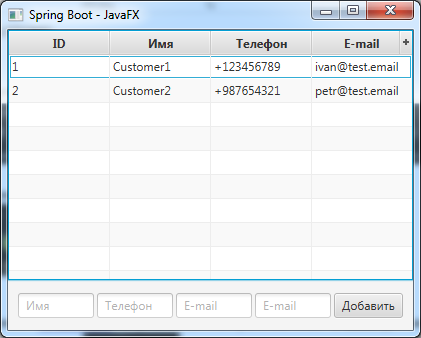


Нажимаем **«Start application»**



Нажимаем на  для запуска в обычном режиме либо на  для запуска в режиме дебаг(break point) ;

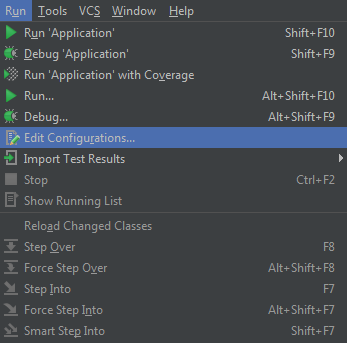
Все работает)))



**2 ВАРИАНТ:**

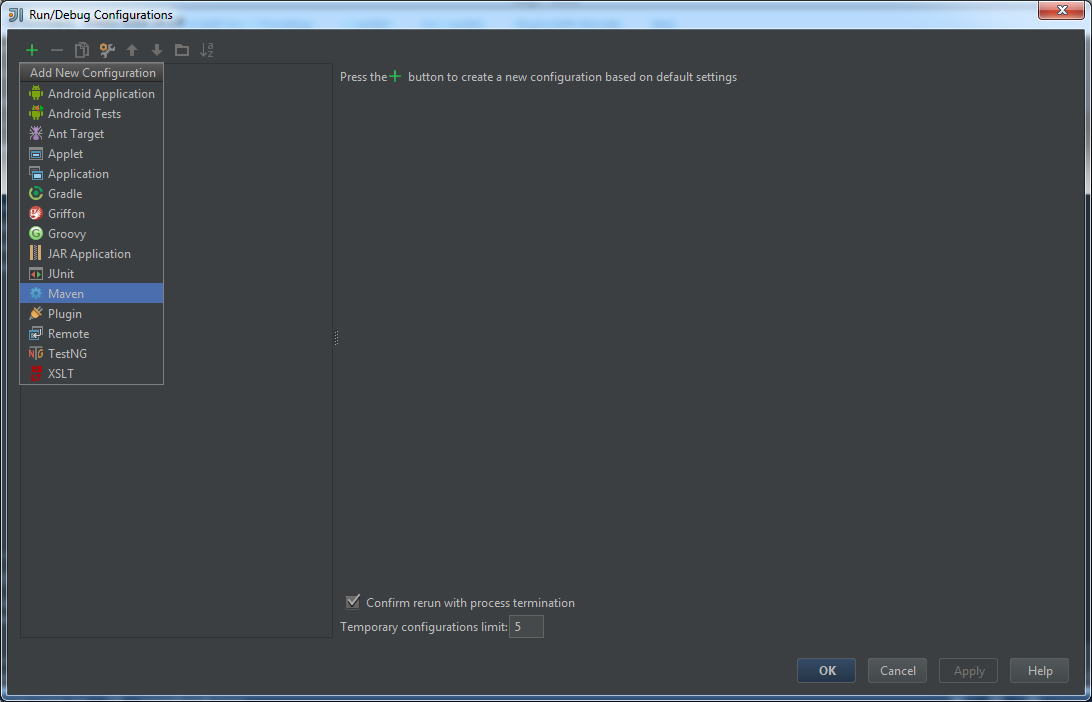
ЗАПУСК ПОСРЕДСТВОМ ИСПОЛЬЗОВАНИЯ КОНФИГУРАЦИИ **MAVEN**

Вернемся к пункту, когда мы создаем конфигурацию и перед нами появляется «+»



Нажимаем на **«+»**

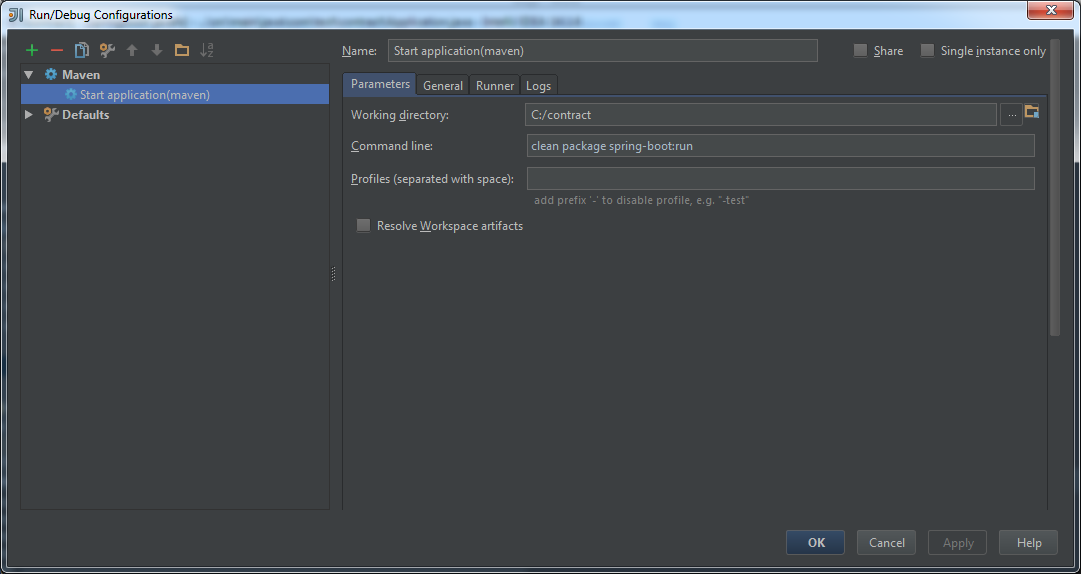
Давайте на этом этапе, вместо **SpringBoot** (который есть только в полной версии IntellijIdea) - выберем **Maven**



В качестве:

Name **-> Start application(maven)**

Commandline **-> clean package spring-boot:run**

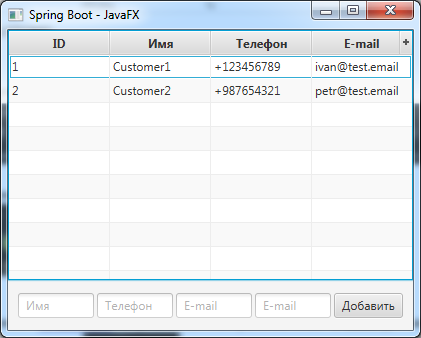


**Попробуем запустить:**

****

Аналогично: нажимаем на  для запуска в обычном режиме либо на  для запуска в режиме дебаг(break point) ;

Все по-прежнему работает)))



**p.s**

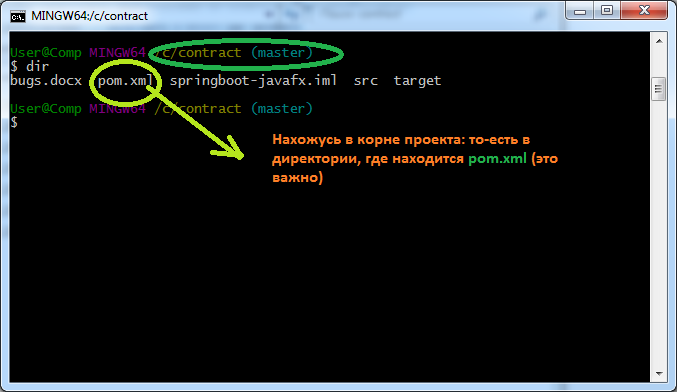
В случае, если у Вас настроены все необходимые переменные среды окружения: MAVEN, JAVA(cм. 1-ую лекцию, 1-ый туториал), - есть еще и **3** вариант – > запуск приложения из консольки

**Для того, чтобы запустить приложение из консоли Вам необходимо:**

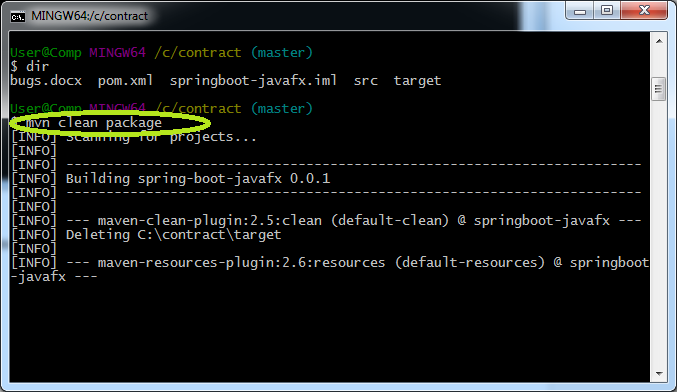
1. Собрать приложение -> получить в итоге **jar archive**
2. Запустить приложение -> т.к в результате сборки у нас получается **executable jar**, есть возможность запуска, посредством использование команды: **java –jar your\_executable\_jar\_name.jar**

**Давайте попробуем запустить из консольки?)**

* **Заходим в корень проекта:**

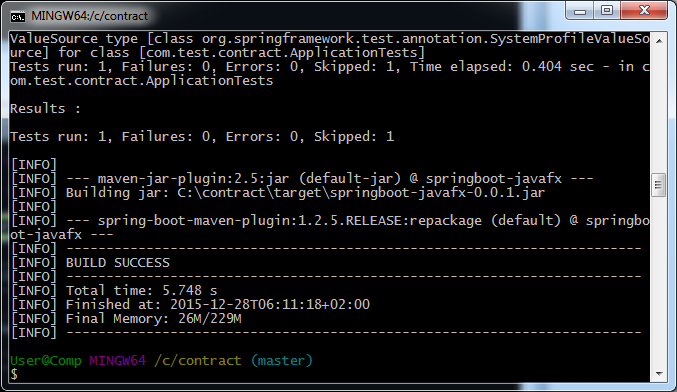
****

* Дальше выполняю сборку проекта - команда**: mvn clean package**

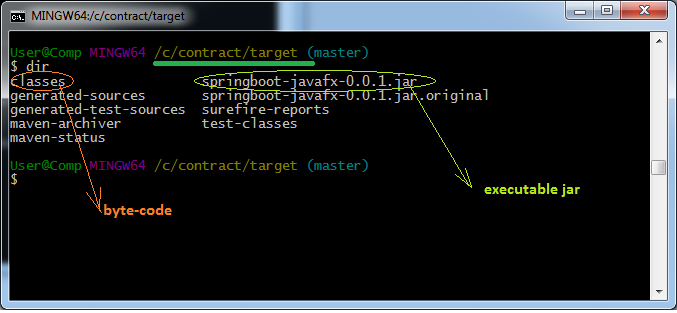
****

В результате успешной сборки:

* Увидите в консольке **message -> Build success**



* В случае перехода в **target** папочку: должны увидеть скомпилированные классы -> **target/classes**(там лежит наш байт-код) + в самой директории **target** -> будет собран нужный нам **executable jar**



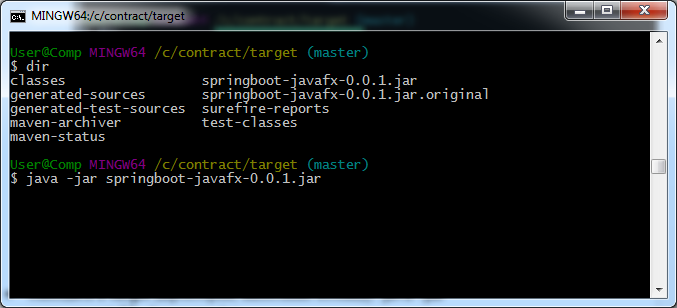
* Запуск приложение ⬄ Exec джара.

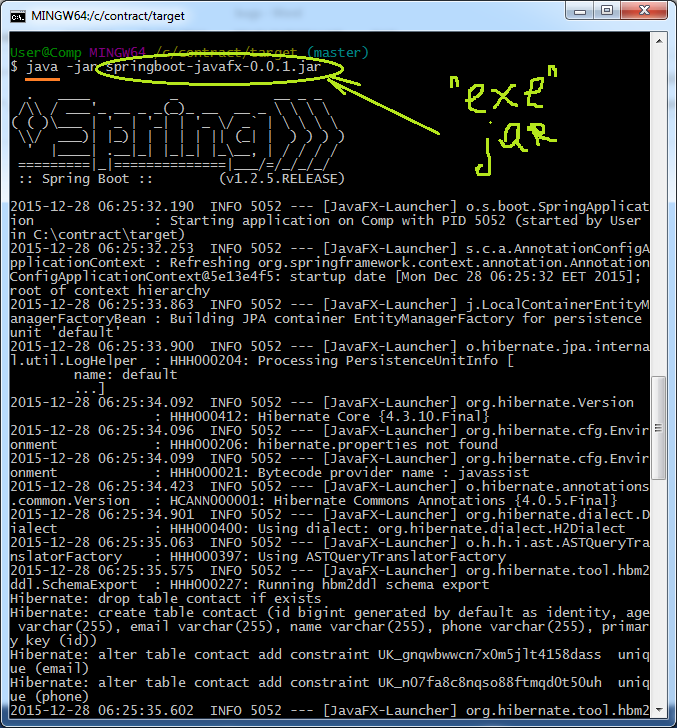
Находясь в **target** директории, выполняю команду:

**java –jar your\_executable\_jar\_name.jar**

**В нашем случае:**

**java –jar springboot-javafx-0.0.1.jar**



****