**Национальный исследовательский университет**

**Высшая школа экономики**

Факультет **«Бизнеса и менеджмента»**

***Командный проект по дисциплине   
«Programming II»***

***Пояснительная записка.***

Выполнили: студенты группы 151:

Тюхова Елена Михайловна

(Canvas: Lena Tyukhova)

Бучко Сергей Алексадрович

(Canvas: Sergey Buchko)

Москва, 2016 год

1. **Название проекта:**

**MoviesBot –**  бот на платформе Telegram, который позволяет получать различную информацию о сфере кино.

1. **Адрес центрального репозитория:**

<https://github.com/CepBuch/MoviesBot>

1. **Краткое описание функций:**

Данная программа обеспечивает непрерывное соединение с сервисом Telegram и обрабатывает каждое пришедшее указанному (при помощи api\_token) боту сообщение. Определенные типы сообщений (команды) обрабатываются особенным образом – вызываются соответствующие этому сообщению функции. Например, если пользователь вводит команду поиска фильмов /moviesearch, то бот отвечает этому пользователю приглашением к вводу фильма и записывает статус этого пользователя как «Ищет фильм», чтобы потом успешно обработать ответ от этого пользователя с названием фильма. Если в процессе выполнения запроса необходимо запоминать выбор пользователя (например, пользователь должен выбрать один из предложенных фильмов), то для этого используются словари <ID чата, данные>. Вся работа с запросами о получении информации о фильмах производится посылкой запросов на веб-сервисы omdb.com и themoviedb.com.

Команды для бота:

/info – Посылает информацию о боте с описанием основных его

/moviesearch – Производит поиск по названию фильма

/peoplesearch – Производит поиск актера, а затем предлагает список наиболее популярных фильмов, в которых этот актер участвовал.

/getbygenre – Возвращает случайный фильм на выбранный пользователем жанр.

/getnowplaying – Возвращает афишу фильмов на текущий день

/getfromtop250 – Возвращает случайный фильм из списка лучших фильмов по версии IMDB

/getsimilars – Возвращает «похожие» фильмы на введенный.

Для формирования ответов бота используется статический класс BotAnswer.

1. **Члены команды и их роли:**

Елена Тюхова, 151(2)

Ответственна:

-Связь с веб-сервисами, предоставляющими информацию о фильмах

-Логика методов, связанных с фильмами.

-UI (ответы бота)

-Обработка сообщений пользователя

Бучко Сергей 151(2)

Ответственен:

-Соединение с ботом Telegram. Получение и посылка сообщений и других запросов.

- Логика

- Обеспечение абстракции

- Обработка сообщений пользователя

1. **Список классов, использованных в программе**

Программа реализована в консольном приложении.

* 1. **MoviesBot**

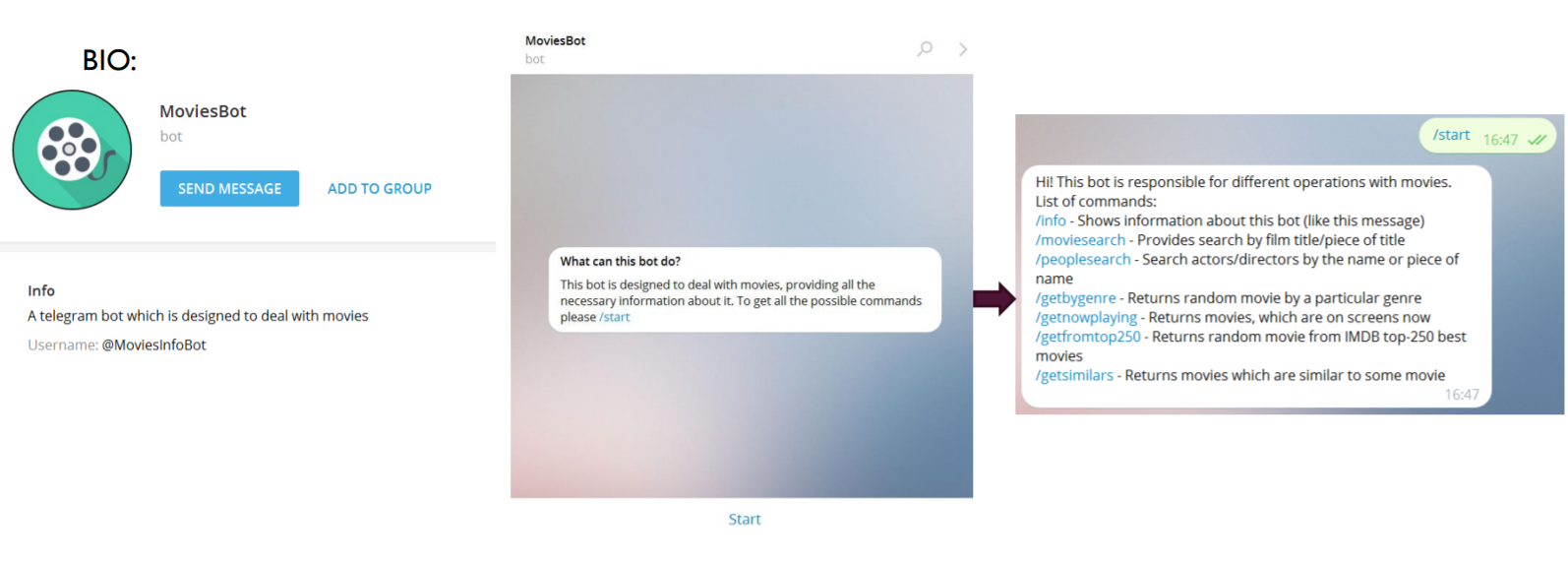
1. MovieService – класс, отвечающий за работу с веб-сервисом для поиска информации по фильмам
2. TelegramBotClient – класс, отвечающий за работу с API Telegram. (отправка/получение сообщений, фото и т.д)
3. BotManager – класс, отвечающий за обработку пришедших сообщений и выполнение соответствующих действий. Принимает IMovieService и ITelegramBotClient.
4. Interfaces – интерфейсы IMovieService и ITelegramBotClient
5. Program – класс, который запускает работу бота Telegram
   1. **MoviedBot.Data**
6. MovieData – папка классов, хранящих в себе данные по фильмам.
7. Model:

I – Movie – класс, описывающий фильм

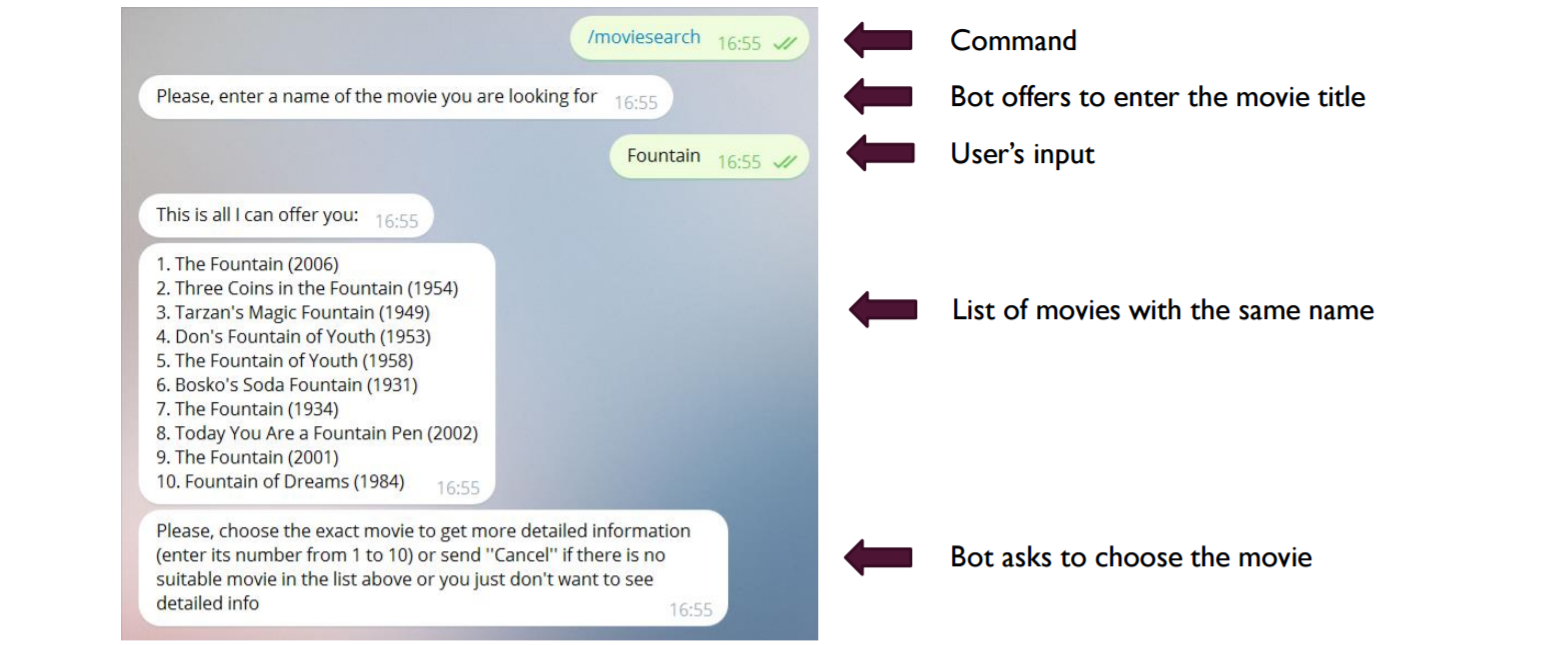
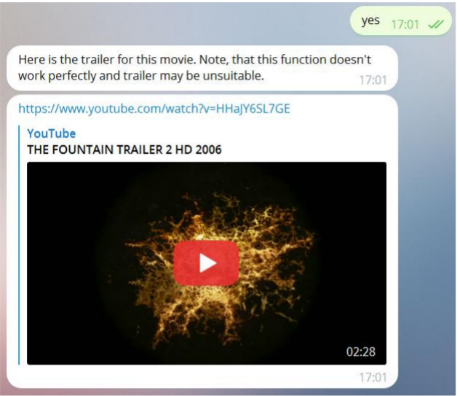
II – Actor – класс, описывающий актера/режиссера

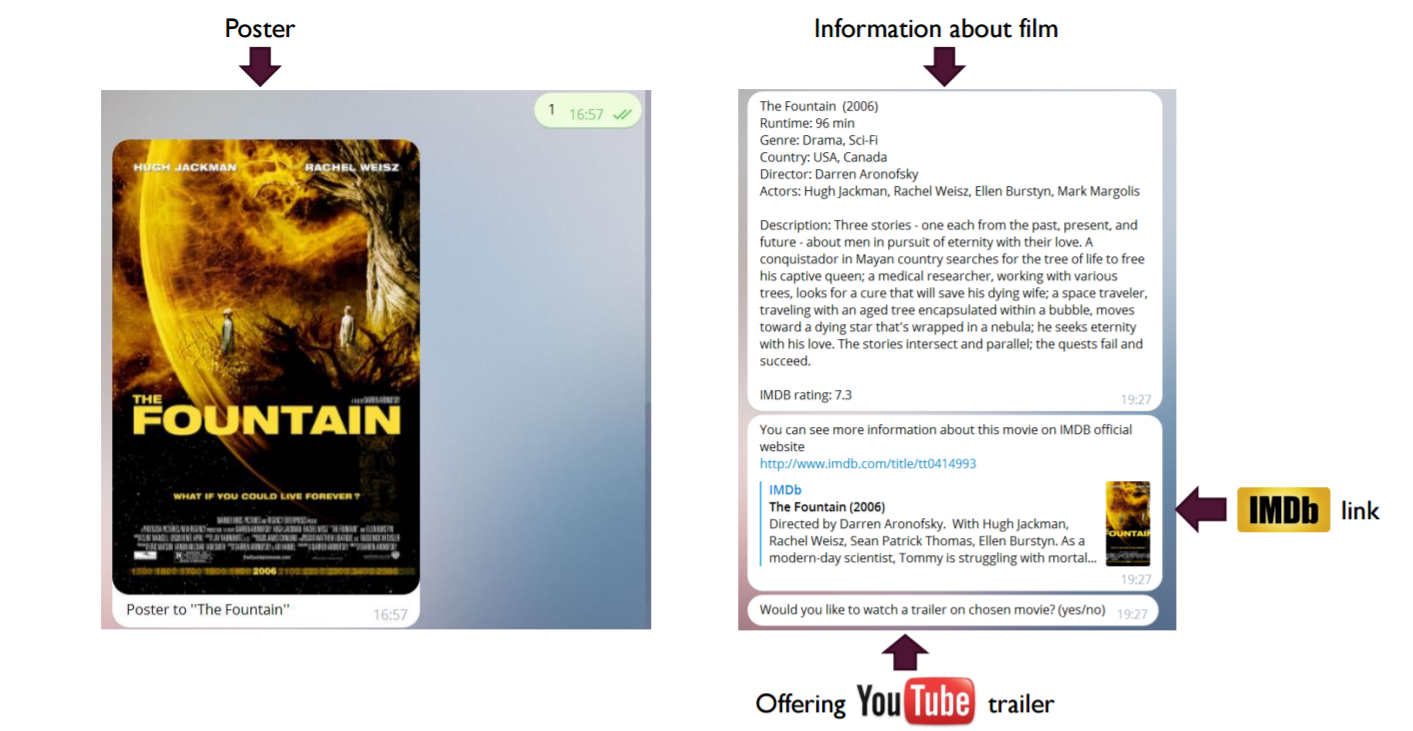
1. omdbDTO – классы, отвечающие за десериализацию json-ответов сервиса omdb.com
2. themoviedbDTO – классы, отвечающие за десерализацию json-ответов сервиса themoviedb.com
3. TegramBotData - папка классов, отвечающих за данные для работы с ботом
4. Enums
5. Types – классы, необходимые для сериализации и десериализации данных
6. BotAnswers – статический класс, формирующий сообщения бота. Все методы возвращают строку, которая и будет отправлена пользователю
7. **Интерфейс программы:**

Для работы с ботом необходимо приложение Telegram. Username нашего бота - @MoviesInfoBot. Его необходимо найти в поиске.

При старте работы с ботом он отправляет информационное сообщение. Это сообщение можно также вызвать, выполнив команду /info

Далее можно воспользоваться одной из 6 команд. Для примера была взята основная команда - /moviesearch.



* При получении этой команды бот отправляет приглашение пользователя к вводу названия фильма
* Затем пользователю возвращается список фильмов, которые возможно удовлетворяют его запрос. И отправляется предложение пользователю выбрать нужный ему фильм

- После того, как пользователь выбрал фильм, ему приходит о нем полная информация, включая постер и ссылку на IMDB. А задается вопрос о надобности трейлера:

