**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ «МОСКОВСКИЙ АВИАЦИОННЫЙ ИНСТИТУТ (НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ)»**

**Журнал практики**

Студента *Лавринова Михаила Александровича*

##### Факультет Ступинский филиал МАИ

###### Кафедра *«Моделирование систем и информационные технологии»*

##### Учебная группа ТСО-105Б-22

Направление подготовки (специальность) *09.03.01*

*(шифр)*

*«Информатика и вычислительная техника»*

*(название направления, специальности)*

Вид практики *учебная*

*(учебной, производственной, преддипломной или другой вид практики)*

Руководитель практики от МАИ

*Мамонов Игорь Михайлович*  \_\_\_\_\_\_\_\_\_\_\_\_\_

*(фамилия, имя, отчество) (подпись)*

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/*Лавринов М.А.*/ “12” июля 2023 г.

*(подпись студента)* *(дата)*

Москва 2023

**1.Место и сроки проведения практики**

*Сроки проведения практики:*

*-дата начала практики 29.06.2023 г.*

*-дата окончания практики 12.07.2023 г.*

*Наименование предприятия Ступинский филиал МАИ*

*Название структурного подразделения (отдел, лаборатория) кафедра «Моделирование систем и информационные технологии»*

1. **Инструктаж по технике безопасности**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/Беспалов В.В./ “29” июня 2023 г.

*(подпись проводившего)* *(дата проведения)*

1. **Индивидуальное задание студенту**

Разработка чат бота интернет магазина с FAQ и услугами по продаже вещей в сети Telegram

**4.План выполнения индивидуального задания**

1. Изучить возможности данной социальной сети Telegram.

2. Выполнить установку и настройку чат-бота.

3. Подготовить реферат.

*Руководитель практики от МАИ*: */Мамонов И.М./*

*Руководитель от предприятия*:  */ /*

*/Лавринов М.А./* “29” июня 2023 г.

*(подпись студента)* *(дата)*

**5.Отзыв руководителя практики от предприятия**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

*Руководитель от предприятия*: / /

*(подпись) (фамилия, имя, отчество)*

«\_\_\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_2023 г.

М.П. (печать)

**6.Отчет студента о практике**

Целью учебной практики являлась разработка чат бота интернет магазина с FAQ и услугами по продаже вещей в сети Telegram.

Для начала нужно установить и настроить PyCharm

### Установка и настройка среды программирования:

Скачиваем и устанавливаем среду разработки PyCharm

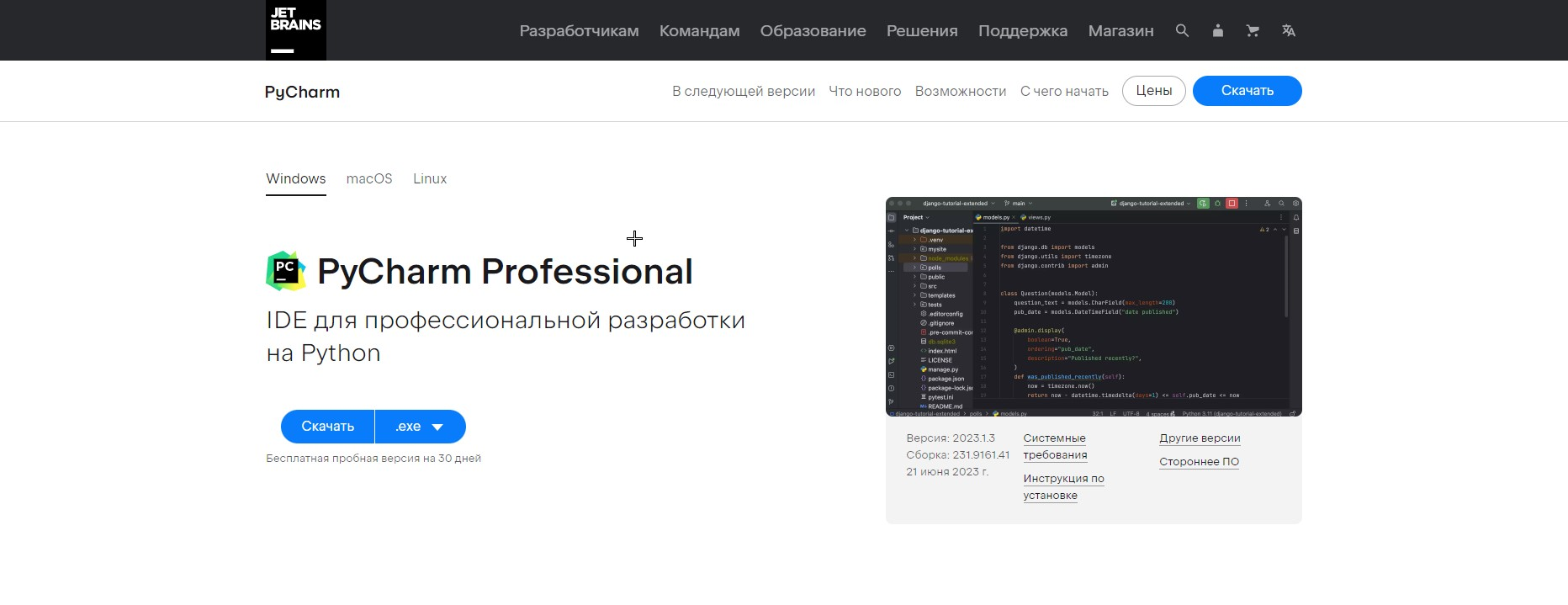


Рис. 1. Скачивание программы PyCharm

Установка библиотеки aiogram

В терминале PyCharm вводим команду pip install aiogram и устанавливаем библиотеку.

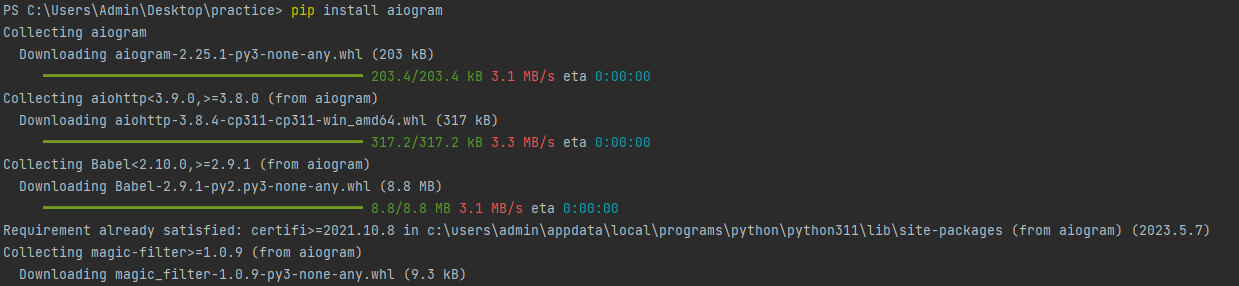


Рис. 2. Установка библиотеки aiogram

Создание основных файлов для работы с ботом.

Создаем файлы в самом проекте для удобства работы.

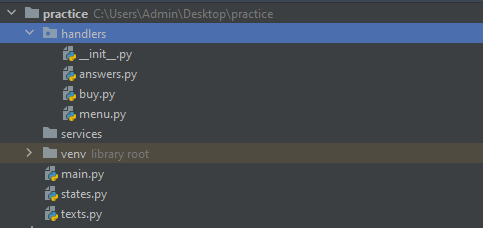


Рис. 3. Создание файлов для работы

Получение токена бота.

Обращаемся к боту botfather, пишем команду /newbot для создания личного бота, далее придумываем название и его URL. В конце нам будет выдан токен нашего бота.



Рис. 4. Получение токена бота

Функция запуска бота.

Для начала импортируем библиотеку aiogram. Далее прописываем функцию def main(): и в ней указываем токен для бота, полученный в botfather. А также создаем диспетчер.

**def** main():  
 bot = Bot(**"5761796030:AAH1vEewZ80\_EG4FQVFlCm3rBZx9m9TXtT0"**)  
 dp = Dispatcher(bot, storage=MemoryStorage())  
 start\_polling(dp, on\_startup=OnStartUp)

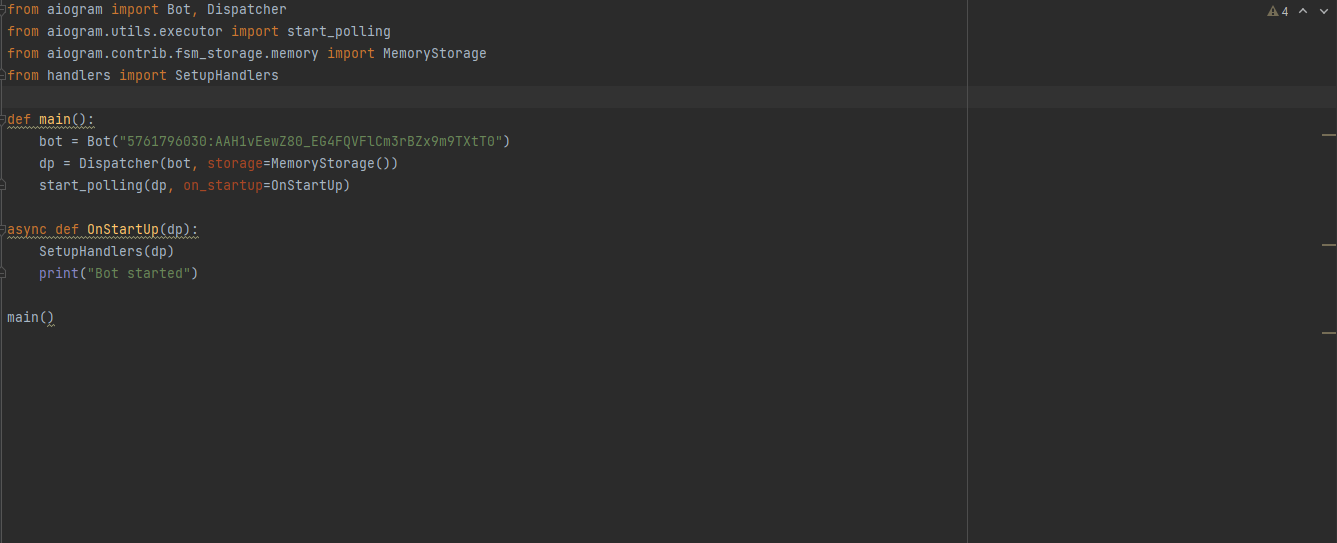


Рис. 5. Функция запуска бота и создание диспетчера

Создание кнопок бота.

Переходим в файл menu.py и создаем функцию async def menumessage в которой указываем названия наших кнопок. Так же пишем строку, где у нас будет присутствовать resize\_keyboard=True, которая отмасштабирует наши кнопки. Кнопки импортируем из texts.py при помощи from texts import btn1, … . Еще прописываем await m.answer, которая будет выводить сообщение от бота, чтобы мы выбрали вопрос.

**async def** menumessage(m:Message):  
 keyboard = ReplyKeyboardMarkup(resize\_keyboard=**True**)  
 keyboard.row(KeyboardButton(btn1), KeyboardButton(btn2))  
 keyboard.row(KeyboardButton(btn3), KeyboardButton(btn4))  
 keyboard.row(KeyboardButton(btn5), KeyboardButton(goods))  
 **await** m.answer(**"Выберите вопрос:"**, reply\_markup=keyboard)

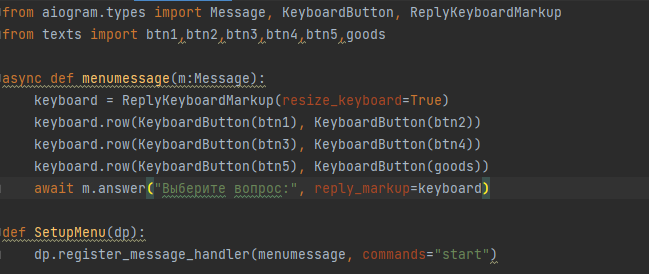


Рис. 6. Создание кнопок и вывод сообщения бота

Создаем текст, который будет в наших кнопках.

Заходим в файл texts.py и прописываем наши названия кнопок и через = в ковычках текст, который нам нужен.

btn1=**"Можно ли где-то оставить отзыв на товар?"**btn2=**"Как выбрать размер?"**btn3=**"Какие способы оплаты?"**btn4=**"В какие города есть доставка?"**btn5=**"Есть ли возможность возврата?"**goods=**"📦Товары"**

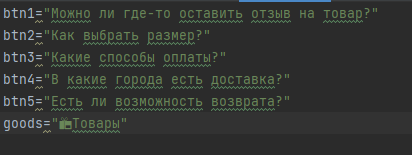


Рис. 7. Создание текста для кнопок

Создаем авто-ответы бота на наши вопросы.

Заходим в answers.py и создаем 6 функций async def Answer1-6, где прописываем через await m.answer наши ответы на вопросы.

**async def** Answer1(m:Message):  
 **await** m.answer(**"Да, на данный момент посмотреть отзывы можно во Вконтакте. В разделе обсуждений."**)  
  
**async def** Answer2(m:Message):  
 **await** m.answer(**"Для того, чтобы определить точный размер по нашей размерной сетке, измерьте обхват груди, талии и бедер с помощью мерной ленты. При выборе между двумя размерами — сделайте предпочтение в сторону большего. Так же можете использовать данные уже имеющихся удобных для вас вещей и сравнить их со значениями в таблице."**)  
  
**async def** Answer3(m:Message):  
 **await** m.answer(**"Онлайн оплата картами Visa, MasterCard, МИР."**)  
  
**async def** Answer4(m:Message):  
 **await** m.answer(**"По всей России через CDEK"**)  
  
**async def** Answer5(m:Message):  
 **await** m.answer(**"Вы вправе отказаться от товара надлежащего качества в течение 7 календарных дней после получения, в случае если произошла ошибка не с вашей стороны. При возврате товаров надлежащего качества стоимость доставки и комиссия за перевод платежа не возмещаются."**)  
  
**async def** Answer6(m:Message):  
 keyboard = InlineKeyboardMarkup()  
 keyboard.add(InlineKeyboardButton(**"Белая мужская рубашка"**, callback\_data=**"Товар 1"**))  
 keyboard.add(InlineKeyboardButton(**"Джинсы мужские"**, callback\_data=**"Товар 2"**))  
 **await** m.answer(**"Выберите товар:"**, reply\_markup=keyboard)

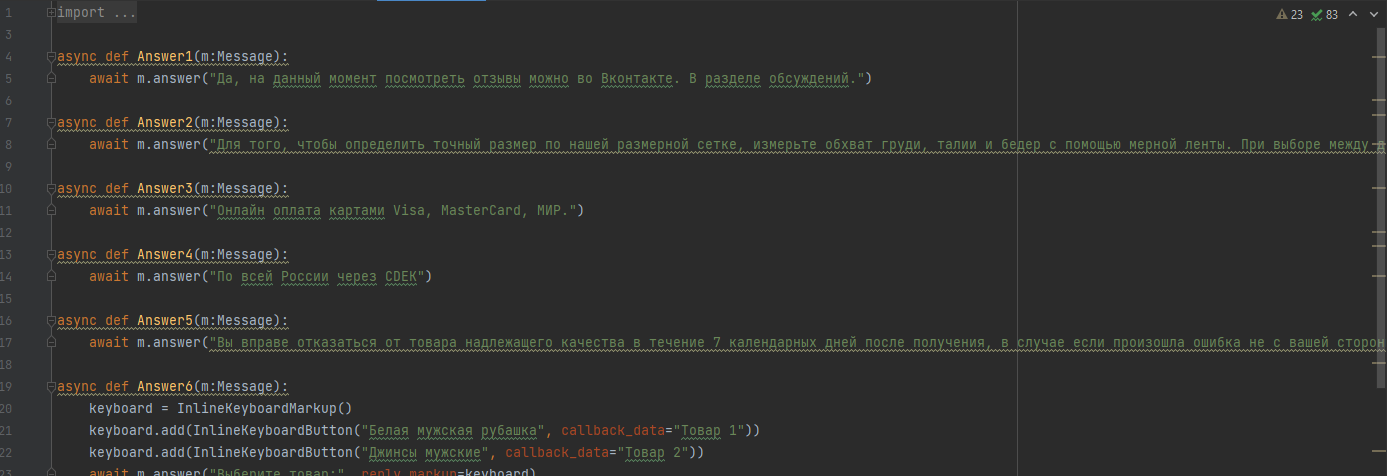


Рис. 8. Создание авто-ответов на вопросы

Регистрируем хендлеры через def SetupAnswers(dp):

**def** SetupAnswers(dp):  
 dp.register\_message\_handler(Answer1, text=btn1)  
 dp.register\_message\_handler(Answer2, text=btn2)  
 dp.register\_message\_handler(Answer3, text=btn3)  
 dp.register\_message\_handler(Answer4, text=btn4)  
 dp.register\_message\_handler(Answer5, text=btn5)  
 dp.register\_message\_handler(Answer6, text=goods)

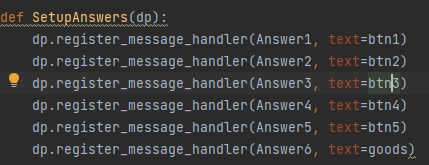


Рис. 9. Регистрация хендлеров

Создаем покупку товаров.

В функции Answer6 пишем добавление 2-х кнопок “Рубашка” и “Джинсы”. С помощью callback\_data делаем передачу данных.

**async def** Answer6(m:Message):  
 keyboard = InlineKeyboardMarkup()  
 keyboard.add(InlineKeyboardButton(**"Белая мужская рубашка"**, callback\_data=**"Товар 1"**))  
 keyboard.add(InlineKeyboardButton(**"Джинсы мужские"**, callback\_data=**"Товар 2"**))  
 **await** m.answer(**"Выберите товар:"**, reply\_markup=keyboard)

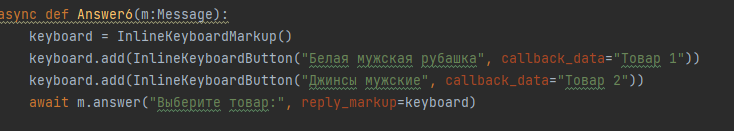


Рис. 10. Создание покупки товаров

Создаем карточку товара.

Переходим в файл buy.py. В ней пишем кнопку для покупки выбранного товара, а также через await c.message.answer\_photo создаем описание товара и его фото. Так же делаем и для второго товара.

**async def** Buygoods1(c:CallbackQuery):  
 keyboard = InlineKeyboardMarkup()  
 keyboard.add(InlineKeyboardButton(**"💲Купить💲"**, callback\_data=**"Купить 1"**))  
 **await** c.message.delete()  
 **await** c.message.answer\_photo(caption=**"Детали:\nАртикул: 280251\nМодель: Antony\nСтрана производства: Словения\nЦвет: Белый\nСостав: 100% хлопок\nРазмер: S-XL\nЦена: 9.999Р"**, reply\_markup=keyboard, photo=**r"https://yandex.ru/images/search?from=tabbar&img\_url=https%3A%2F%2Fae04.alicdn.com%2Fkf%2FS53e3bc89279f432183448ef1781abb71i.jpeg&lr=10756&pos=0&rpt=simage&text=%D1%80%D1%83%D0%B1%D0%B0%D1%88%D0%BA%D0%B0%20%D0%BC%D1%83%D0%B6%D1%81%D0%BA%D0%B0%D1%8F%20%D0%B1%D0%B5%D0%BB%D0%B0%D1%8F"**)

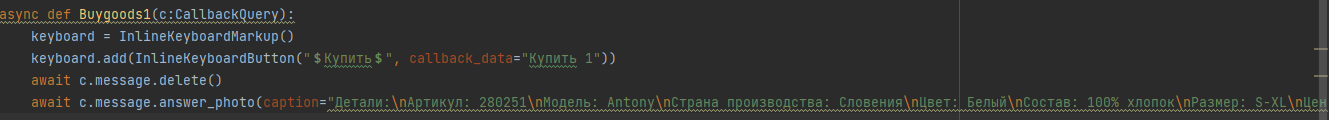


Рис. 11. Создание карточки товара

Создаем проверку фильтра callback\_data

**def** SetupBuy(dp):  
 dp.register\_callback\_query\_handler(Buygoods1, **lambda** c: c.data==**"Товар 1"**)  
 dp.register\_callback\_query\_handler(Buygoods2, **lambda** c: c.data == **"Товар 2"**)  
 dp.register\_callback\_query\_handler(AskAddress, **lambda** c: c.data==**"Купить 1"**)  
 dp.register\_callback\_query\_handler(AskAddress, **lambda** c: c.data == **"Купить 2"**)  
 dp.register\_message\_handler(SendToAdmin, state=UserStates.wait\_address)

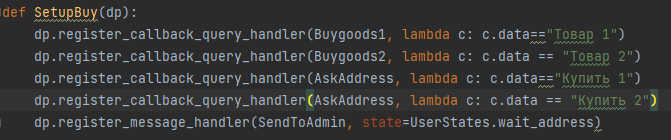


Рис. 12. Создание проверки фильтра callback\_data

Создаем процесс покупки.

Пишем функцию askaddress, где бот сможет нам после нажатия на кнопку “Купить” предложить написать наш адрес доставки.

**async def** AskAddress(c:CallbackQuery):  
 **await** UserStates.wait\_address.set()  
 **await** c.message.delete()  
 **await** c.message.answer(**"Введите ваш адрес:\nГород:\nулица:\nдом:\nквартира:"**)

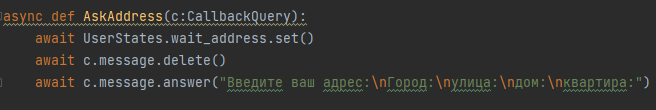


Рис. 13. Создание процесса покупки

Далее создаем отчет о покупке и инфо про покупку и покупателя для админа.

В функцию SendToAdmin прописываем админские права через m.bot.send\_message, где вводим наш id телеграмма. Так же создаем ответ бота покупателю через await m.answer.

**async def** SendToAdmin(m:Message, state:FSMContext):  
 **await** state.finish()  
 **await** m.bot.send\_message(968876784, **f"Новая покупка:\n{**m.text**}\n@{**m.from\_user.username**}"**)  
 **await** m.answer(**"Ожидайте ответа продавца, он вскоре свяжется с вами"**)

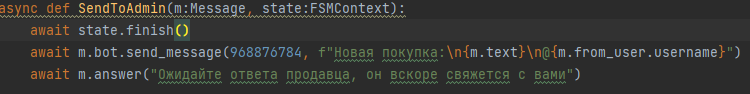


Рис. 14. Создание отчета о покупке и инфо покупателя для админа

Переходим в states.py

Создаем класс в котором содержится состояние для машины состояний(FSM).

**class** UserStates(StatesGroup):  
 wait\_address=State()



Рис. 15. Создание класса для состояния FSM

Проверяем бота.

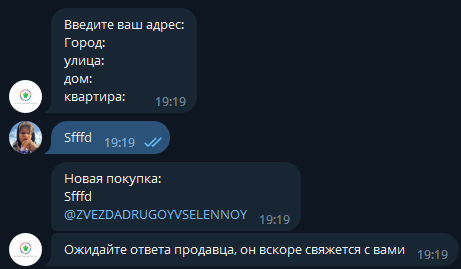
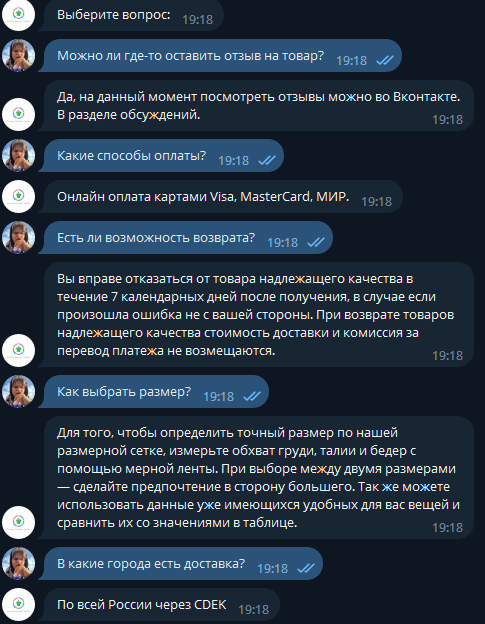


Рис. 16-18. Проверка работы бота

Как мы видим, бот работает.

В итоге, был запрограммирован и настроен чат-бот, в котором можно получить ответы на часто задаваемые вопросы (FAQ) и купить товары, которые предоставлены.

В результате прохождения учебной практики были изучены и выполнены следующие задачи:

* изучена технология создания ботов на основе библиотеки Aiogram;
* создан чат-бот в Telegram для получения FAQ и покупки товаров;
* выполнено тестирование;
* подготовлен реферат.