** Министерство науки и высшего образования Российской Федерации Федеральное государственное бюджетное образовательное учреждение высшего образования**

**«Московский государственный технический университет имени Н.Э. Баумана**

**(национальный исследовательский университет МГТУ им. Н.Э. Баумана)**

**Факультет «Информатика и системы управления» Кафедра «Системы обработки информации и управления»**

Лабораторная работа №6

**«Разработка бота для Telegram с использованием языка Python»**

по предмету

«Базовые компоненты интернет-технологий»

Выполнил:

студент группы № ИУ5-31Б

Михалёв Ярослав

Проверил:

Преподаватель кафедры ИУ-5

Гапанюк Юрий

2022 г.

Задание

Разработайте простого бота для Telegram. Бот должен использовать функциональность создания кнопок.

Текст программы

**config.py**

BOT\_TOKEN = ""  
ADMIN\_ID = ""

**loader.py**

*import* logging  
  
*from* aiogram *import* Bot, Dispatcher, types  
  
*from* config *import* BOT\_TOKEN  
  
  
bot = Bot(token=BOT\_TOKEN, parse\_mode=types.ParseMode.HTML)  
dp = Dispatcher(bot)  
  
logging.basicConfig(format=u'%(filename)s [LINE:%(lineno)d] #%(levelname)-8s [%(asctime)s] %(message)s', level=logging.INFO)

**app.py**

*from* aiogram *import* executor  
*from* handlers *import* dp  
*from* loader *import* bot  
  
*from* utils.notify\_admins *import* on\_startup\_notify  
  
  
*async def* on\_startup(*dp*):  
 *await* on\_startup\_notify(*dp*)  
  
*async def* on\_shutdown(dp):  
 *await* bot.close()  
  
  
*if* \_\_name\_\_ == '\_\_main\_\_':  
 executor.start\_polling(dp, on\_startup=on\_startup\_notify, on\_shutdown=on\_shutdown)

**keyboards/choice\_buttons.py**

*from* aiogram.types *import* InlineKeyboardButton, InlineKeyboardMarkup  
  
  
choice = InlineKeyboardMarkup(row\_width=2)  
  
get\_balance = InlineKeyboardButton(text="Узнать баланс", callback\_data="get\_balance")  
choice.insert(get\_balance)  
  
deposit\_money = InlineKeyboardButton(text="Пополнить счёт", callback\_data="deposit\_money")  
choice.insert(deposit\_money)  
  
derive\_money = InlineKeyboardButton(text="Снять наличные", callback\_data="derive\_money")  
choice.insert(derive\_money)  
  
cancel\_button = InlineKeyboardButton(text="Выход", callback\_data="cancel")  
choice.insert(cancel\_button)

**handlers/operations.py**

*import* logging  
  
*from* aiogram.dispatcher *import* FSMContext  
*from* aiogram.dispatcher.filters *import* Command  
*from* aiogram.types *import* Message, CallbackQuery  
  
*from* keyboards.choice\_buttons *import* choice  
*from* loader *import* dp  
  
*from* utils.balance *import* getBalance, changeBalance  
  
*from* config *import* ADMIN\_ID  
  
  
@dp.message\_handler(commands=["start", "help"])  
*async def* show\_menu(*message*: Message):  
 *await message*.answer(text="Сбербанк банкомат", reply\_markup=choice)  
  
  
@dp.callback\_query\_handler(text="get\_balance")  
*async def* get\_balance(*call*: CallbackQuery):  
 *await call*.message.answer(f"Ваш баланс: {getBalance()} рублей")  
  
  
@dp.callback\_query\_handler(text="deposit\_money")  
*async def* on\_deposit\_money\_pressed(*call*: CallbackQuery):  
 *await call*.answer(cache\_time=60)  
  
 newBalance = getBalance() + 100  
 changeBalance(newBalance)  
  
 logging.info(f"Баланс пользователя {ADMIN\_ID} увеличен на 100 рублей")  
 logging.info(f"Текущий баланс пользователя {ADMIN\_ID} составляет {getBalance()} рублей")  
 *await call*.message.answer("Баланс пополнен на 100 рублей", reply\_markup=choice)  
  
  
@dp.callback\_query\_handler(text="derive\_money")  
*async def* on\_derive\_money\_pressed(*call*: CallbackQuery):  
 *await call*.answer(cache\_time=60)  
  
 newBalance = getBalance() - 100  
 changeBalance(newBalance)  
  
 logging.info(f"Баланс пользователя {ADMIN\_ID} уменьшен на 100 рублей")  
 logging.info(f"Текущий баланс пользователя {ADMIN\_ID} составляет {getBalance()} рублей")  
 *await call*.message.answer("Вы вывели 100 рублей", reply\_markup=choice)  
  
  
@dp.callback\_query\_handler(text="cancel")  
*async def* cancel\_buying(*call*: CallbackQuery):  
 logging.info(f"Нажата кнопка выхода")  
  
 # Ответим в окошке с уведомлением!  
 *await call*.answer("Вы вышли", show\_alert=*True*)  
  
 # Отправляем пустую клавиатуру изменяя сообщение, для того, чтобы ее убрать из сообщения!  
 *await call*.message.edit\_reply\_markup(reply\_markup=*None*)

**utils/balance.py**

*import* random  
  
*from* config *import* ADMIN\_ID  
*from* .db *import* write\_data, load\_data  
  
  
*def* generateBalance():  
 balance = random.randint(0, 10000)  
 data = {  
 "user\_id": ADMIN\_ID,  
 "balance": balance  
 }  
  
 write\_data(data)  
  
  
*def* getBalance():  
 data = load\_data()  
 *return* data["balance"]  
  
  
*def* changeBalance(*newBalance*):  
 data = load\_data()  
 data["balance"] = *newBalance* write\_data(data)

**utils/db.py**

*import* json  
  
  
*def* write\_data(*data*, *title*="data/data"):  
 *with* open(f"{*title*}.json", "w", encoding="utf-8") *as* file:  
 json.dump(*data*, file, indent=2, ensure\_ascii=*False*)  
  
*def* load\_data(*title*="data/data"):  
 *with* open(f"{*title*}.json", "r") *as* file:  
 data = json.load(file)  
  
 *return* data

**utils/notify\_admins.py**

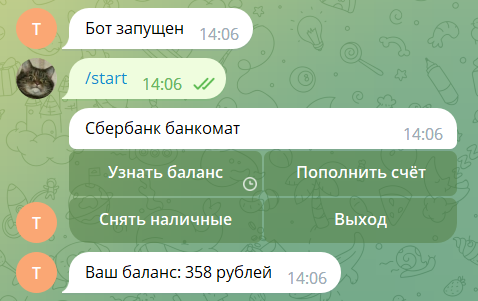
*import* logging  
  
*from* aiogram *import* Dispatcher  
  
*from* config *import* ADMIN\_ID  
  
  
*async def* on\_startup\_notify(*dp*: Dispatcher):  
 logging.info("Бот запущен")  
 *await dp*.bot.send\_message(chat\_id=ADMIN\_ID, text="Бот запущен")

**data/data.json**

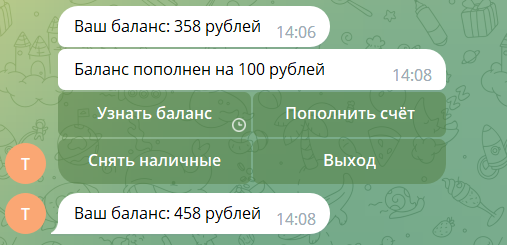
{  
 "user\_id": "",  
 "balance": 358  
}

Возможности бота

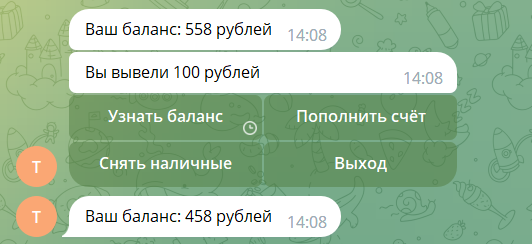
Вывести текущий баланс пользователя



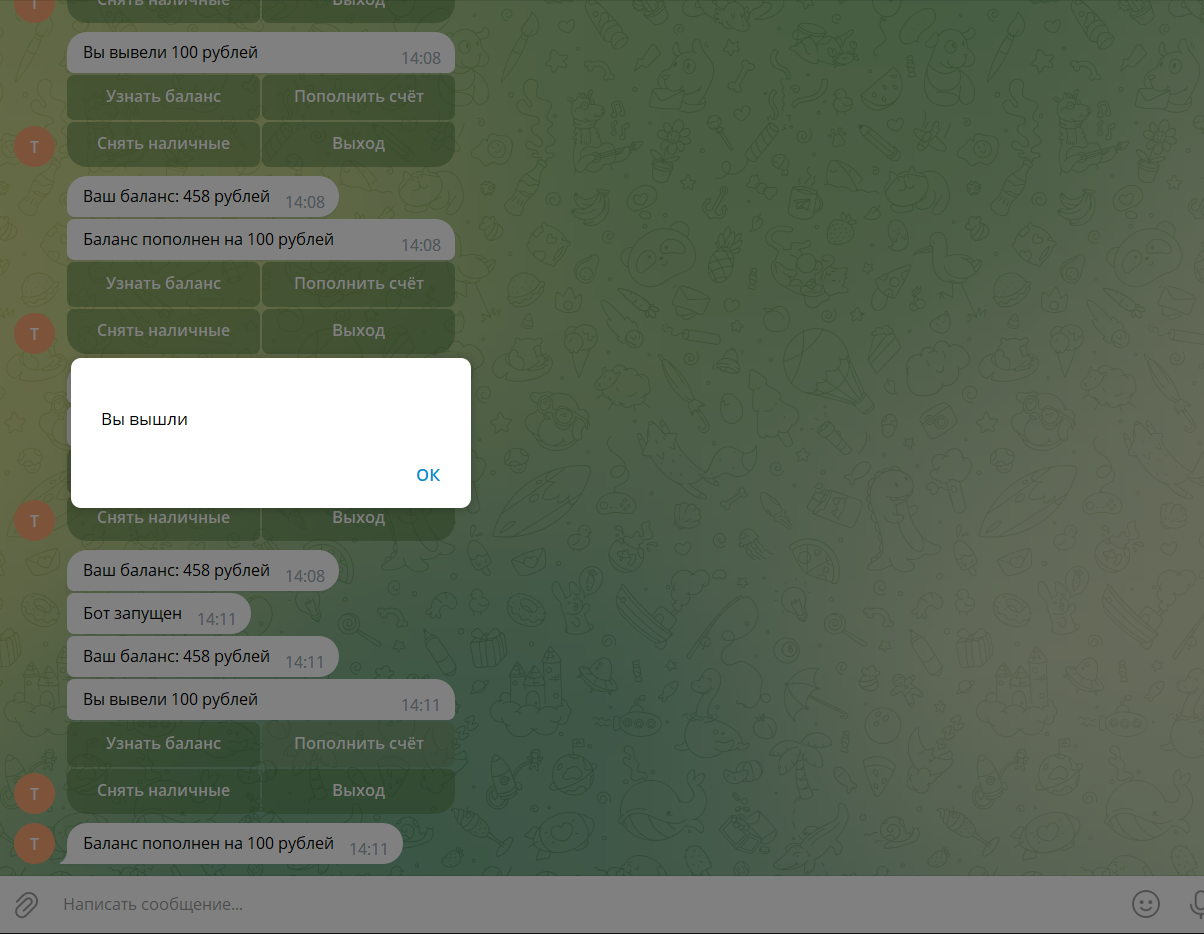
Пополнить счет на 100 рублей



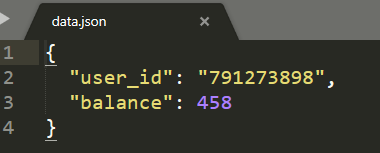
Вывести 100 рублей



Остановка переписки



Данные о счете хранятся в файле data / data.json



Все изменения логируются

