**Московский государственный технический университет им. Н.Э. Баумана**

Факультет «Информатика и системы управления»

Кафедра ИУ5 «Системы обработки информации и управления»

Курс «Парадигмы и конструкции языков программирования» Отчет по домашнему заданию

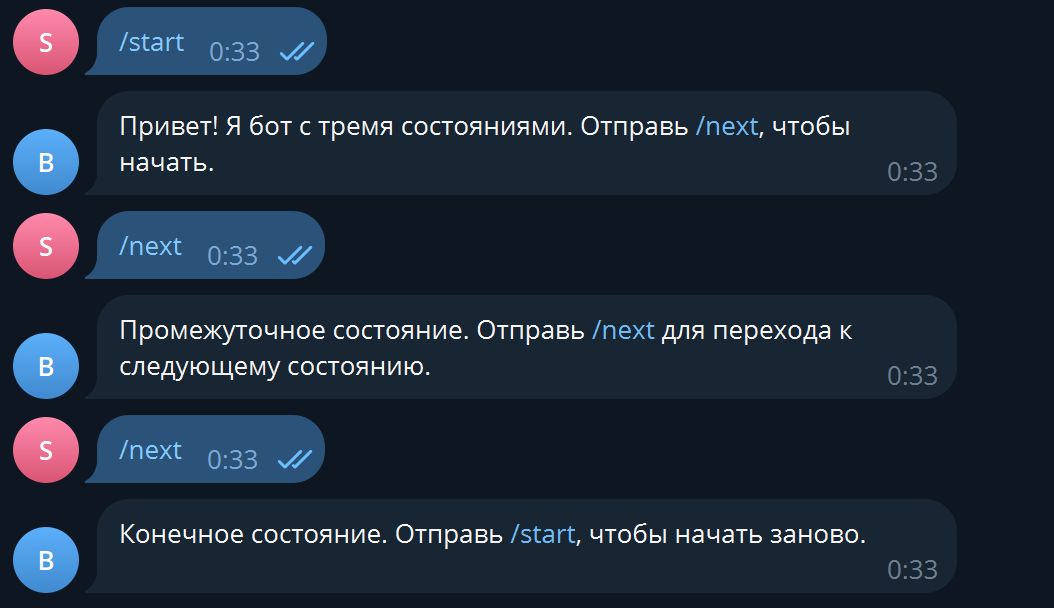
|  |  |
| --- | --- |
| Выполнил: | Проверил: |
| студент группы ИУ5-31Б | преподаватель каф. ИУ5 |
| Филенко Александр |  |
| Подпись и дата: | Подпись и дата: |

# Постановка задачи

**Задание:**

1. Разработайте бота для Telegram. Бот должен реализовывать конечный автомат из трех состояний.

# Результат

****

# Текст программы

from telegram import Update

from telegram.ext import Updater, CommandHandler, MessageHandler, Filters, CallbackContext, ConversationHandler

# Определение состояний

START, INTERMEDIATE, END = range(3)

def start(update: Update, context: CallbackContext) -> int:

    update.message.reply\_text("Привет! Я бот с тремя состояниями. Отправь /next, чтобы начать.")

    return START

def intermediate(update: Update, context: CallbackContext) -> int:

    update.message.reply\_text("Промежуточное состояние. Отправь /next для перехода к следующему состоянию.")

    return INTERMEDIATE

def end(update: Update, context: CallbackContext) -> int:

    update.message.reply\_text("Конечное состояние. Отправь /start, чтобы начать заново.")

    return END

def cancel(update: Update, context: CallbackContext) -> int:

    update.message.reply\_text("Операция отменена.")

    return ConversationHandler.END

def main():

    # Замените '6940234369:AAFE0CQjYCN3BIvX2Fq46BxfF83vW8ly35A' на реальный токен вашего бота

    updater = Updater(token='6940234369:AAFE0CQjYCN3BIvX2Fq46BxfF83vW8ly35A', use\_context=True)

    dp = updater.dispatcher

    conv\_handler = ConversationHandler(

        entry\_points=[CommandHandler('start', start)],

        states={

            START: [CommandHandler('next', intermediate)],

            INTERMEDIATE: [CommandHandler('next', end)],

            END: [CommandHandler('start', start)],

        },

        fallbacks=[CommandHandler('cancel', cancel)]

    )

    dp.add\_handler(conv\_handler)

    updater.start\_polling()

    updater.idle()

if \_\_name\_\_ == '\_\_main\_\_':

    main()