

## Лабораторная работа №6

### Разработка бота на основе конечного автомата для Telegram с использованием языка Python.

Цель лабораторной работы: изучение разработки ботов в Telegram.

#### Задание:

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

#### Текст программы

```
from telegram import Update
from telegram.ext import Application, CommandHandler, ContextTypes, ConversationHandler

# Состояния
STATE1, STATE2, STATE3 = range(3)

async def start(update: Update, context: ContextTypes.DEFAULT_TYPE) -> int:
    await update.message.reply_text('Добро пожаловать! Введите /hello для приветствия.')
    return STATE2

async def hello(update: Update, context: ContextTypes.DEFAULT_TYPE) -> int:
    await update.message.reply_text('Привет! Введите /bye для завершения.')
    return STATE3

async def bye(update: Update, context: ContextTypes.DEFAULT_TYPE) -> int:
    await update.message.reply_text('До свидания! Напишите /start для нового начала.')
    return ConversationHandler.END

async def unknown(update: Update, context: ContextTypes.DEFAULT_TYPE) -> None:
    await update.message.reply_text('Неизвестная команда.')

def main() -> None:
    # Замените <TOKEN> на ваш токен
    application = Application.builder().token("токен").build()

    # Определяем состояния и их обработчики
    conv_handler = ConversationHandler(
        entry_points=[CommandHandler('start', start)],
        states={
            STATE2: [CommandHandler('hello', hello)],
            STATE3: [CommandHandler('bye', bye)],
        },
        fallbacks=[CommandHandler("unknown", unknown)],
    )

    application.add_handler(conv_handler)

    application.run_polling()
```

```
if __name__ == '__main__':  
    main()
```

### Пример выполнения программы



