

Защищено:
Гапанюк Ю.Е.

"__" 2025 г.

Демонстрация:
Гапанюк Ю.Е.

"__" 2025 г.

**Отчет по лабораторной работе № 6 по курсу
Парадигмы и конструкции языков программирования
ГУИМЦ**

**Тема работы: "Разработка бота на основе конечного автомата для
Telegram с использованием языка Python."**

4
(количество листов)

ИСПОЛНИТЕЛЬ:

студент группы ИУ5Ц-54Б

Щетинин Д.С.

(подпись)

"__" 2025 г.

1. Тема и задание для выполнения лабораторной работы.

Тема работы: "Изучение разработки бота на основе конечного автомата для Telegram с использованием языка Python."

Задание:

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

2. Листинг программы

```
from telegram import Update
from telegram.ext import (
    Application,
    CommandHandler,
    MessageHandler,
    filters,
    ConversationHandler,
    ContextTypes,
)
# Определяем состояния
START, INPUT_NAME, INPUT_AGE = range(3)

# Обработчик команды /start
async def start(update: Update, context: ContextTypes.DEFAULT_TYPE) -> int:
    await update.message.reply_text(
        "Привет! Я бот с конечным автоматом. Напиши /begin чтобы начать."
    )
    return START

# Начало диалога
async def begin(update: Update, context: ContextTypes.DEFAULT_TYPE) -> int:
    await update.message.reply_text(
        "Введи своё имя:"
    )
    return INPUT_NAME

# Обработка введённого имени
async def input_name(update: Update, context: ContextTypes.DEFAULT_TYPE) -> int:
    user_name = update.message.text
    context.user_data['name'] = user_name
    await update.message.reply_text(
        f"Приятно познакомиться, {user_name}! Теперь введи свой возраст."
    )
    return INPUT_AGE

# Обработка введённого возраста
async def input_age(update: Update, context: ContextTypes.DEFAULT_TYPE) -> int:
    age = update.message.text
    if not age.isdigit():

```

```

await update.message.reply_text("Пожалуйста, введите число!")
return INPUT_AGE

context.user_data['age'] = int(age)
await update.message.reply_text(
    f"Спасибо! Ты {context.user_data['name']}, тебе {age} лет.\n"
    "Напиши /begin чтобы начать заново.\n"
    "Или /cancel, если хочешь прервать диалог."
)
return START

# Отмена диалога
async def cancel(update: Update, context: ContextTypes.DEFAULT_TYPE) -> int:
    await update.message.reply_text(
        "Диалог прерван. Напиши /begin если хочешь начать заново."
    )
    return START

def main() -> None:
    # Создаём приложение и передаём токен бота
    application = Application.builder().token("8230982855:AAEi8j5V5-
Lw8ADfoPKOrDEgHBZnOG09UEE").build()

    # Создаём обработчик диалога
    conv_handler = ConversationHandler(
        entry_points=[CommandHandler('start', start)],
        states={
            START: [
                CommandHandler('begin', begin),
            ],
            INPUT_NAME: [
                MessageHandler(filters.TEXT & ~filters.COMMAND, input_name),
            ],
            INPUT_AGE: [
                MessageHandler(filters.TEXT & ~filters.COMMAND, input_age),
            ],
        },
        fallbacks=[CommandHandler('cancel', cancel)],
    )

    # Добавляем обработчик в приложение
    application.add_handler(conv_handler)

    # Запускаем бота
    application.run_polling()

if __name__ == '__main__':
    main()

```

3. Результаты работы программы

