Московский государственный технический университет им. Н.Э. Баумана

Факультет «Радиотехнический»
Кафедра «Системы обработки информации и управления»

Курс «Парадигмы и конструкции языков программирования »

Отчет по лабораторной работе №5
«Разработка простого бота для Telegram с использованием языка Python.»

Выполнил:

студент
группы
РТ5-31Б

Ермаков И.А.

Москва, 2024 г

Описание задания:

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

Текст программы:
Telegram_bot.py

```
from telegram import Update, InlineKeyboardButton, InlineKeyboardMarkup
from telegram.ext import Application, CommandHandler, CallbackQueryHandler,
ContextTypes

TOKEN = '8138558041:AAGsmHE8_7xCeOiueBM2RszHz9qQeofFzJs'

# Функция для обработки команды /start
async def start(update: Update, context: ContextTypes.DEFAULT_TYPE) ->
None:
    # Создаем клавиатуру с кнопками
    keyboard = [
        [InlineKeyboardButton("Пойти на лабы по ПиКЯП", callback_data='1')],
        [InlineKeyboardButton("Пойти на лабы по ПКШ", callback_data='2')],
        [InlineKeyboardButton("Пойти на лабы по ООП", callback_data='3')]
    ]
    # Оборачиваем кнопки в разметку для отправки в чат
    reply_markup = InlineKeyboardMarkup(keyboard)

    # Отправляем сообщение с кнопками
    await update.message.reply_text('Пожалуйста, выберите:',
reply_markup=reply_markup)

# Функция для обработки нажатий на кнопки
async def button(update: Update, context: ContextTypes.DEFAULT_TYPE) ->
None:
    query = update.callback_query  # Получаем информацию о нажатой кнопке
    await query.answer()  # Подтверждаем нажатие кнопки

    # Определяем действие в зависимости от нажатой кнопки
    if query.data == '1':
        await query.edit_message_text(text="Вы пошли на лабы по ПиКЯП")
    elif query.data == '2':
        await query.edit_message_text(text="Вы пошли на лабы по ПКШ")
    elif query.data == '3':
        await query.edit_message_text(text="Вы пошли на лабы по ООП")
```

```
# Функция для обработки команды /help
async def help_command(update: Update, context:
ContextTypes.DEFAULT_TYPE) -> None:
    # Отправляем подсказку с инструкцией по использованию бота
    await update.message.reply_text("Используйте /start для начала работы с
ботом.")

def main() -> None:
    # Создаем объект Application с токеном для инициализации бота
    application = Application.builder().token(TOKEN).build()

    # Добавляем обработчики для команд и нажатий на кнопки
    application.add_handler(CommandHandler("start", start))  # обработчик
команды /start
    application.add_handler(CallbackQueryHandler(button))  # обработчик
нажатий на кнопки
    application.add_handler(CommandHandler("help", help_command))  #
обработчик команды /help

    # Запускаем бота в режиме polling для отслеживания обновлений
    application.run_polling()

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

Пример выполнения программы