**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**

