

Отчет по лабораторной работе №5

«Разработка простого бота для Telegram с использованием языка Python.»

Цель лабораторной работы: изучение разработки ботов в Telegram.

Задание:

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

Текст программы:

```
from typing import Final
from telegram import Update, KeyboardButton, ReplyKeyboardMarkup
from telegram.ext import Application, CommandHandler, MessageHandler,
filters, ContextTypes

TOKEN: Final = "*****"
BOT_USERNAME: Final = '*****'

async def start_command(update: Update, context: ContextTypes.DEFAULT_TYPE):
    keyboard = [
        [KeyboardButton("Узнать гороскоп на 2025")],
        [KeyboardButton("Помощь")]
    ]
    reply_markup = ReplyKeyboardMarkup(keyboard, resize_keyboard=True)

    await update.message.reply_text("Привет! Узнай свой гороскоп на 2025!",
    reply_markup=reply_markup)

async def help_command(update: Update, context: ContextTypes.DEFAULT_TYPE):
    await update.message.reply_text("Я HoroscopeBot! Напиши что-нибудь, я
отвечу!")

async def custom_command(update: Update, context: ContextTypes.DEFAULT_TYPE):
    await update.message.reply_text("Это custom команда!")

# ОТВЕТЫ
def handle_response(text: str) -> str:
    processed: str = text.lower()

    if "помощь" in processed:
        return "В будущем я смогу вам помочь."
    if "узнать гороскоп на 2025" in processed:
        return "В будущем я вам расскажу его."

    if "привет" in processed:
```

```

        return "Привет! Я вижу, вы зашли узнать свой гороскоп!"
    if "как дела?" in processed:
        return "Я отлично, могу помочь узнать, как будут ваши дела в следующем году!"

    return "Я не понял, что вы написали."

async def handle_message(update: Update, context: ContextTypes.DEFAULT_TYPE):
    message_type: str = update.message.chat.type
    text: str = update.message.text

    print(f'User ({update.message.chat.id}) in chat type "{message_type}": "{text}"')

    response: str = handle_response(text)
    print('Bot: ', response)
    await update.message.reply_text(response)

async def error(update: Update, context: ContextTypes.DEFAULT_TYPE):
    print(f'Update {update} caused error {context.error}')

if __name__ == '__main__':
    print("Starting...")
    app = Application.builder().token(TOKEN).build()

    app.add_handler(CommandHandler('start', start_command))
    app.add_handler(CommandHandler('help', help_command))
    app.add_handler(CommandHandler('custom', custom_command))

    app.add_handler(MessageHandler(filters.TEXT, handle_message))

    app.add_error_handler(error)

    print('Polling...')
    app.run_polling(poll_interval=3)

```

Экранные формы с примерами выполнения программы.



```

horoscope x
D:\УНИК\BOT_TG\Scripts\python.exe D:/Users/User/PycharmProjects/BOT_TG/horoscope.py
Starting....
Polling...

```





