Министерство науки и высшего образования Российской Федерации  
ФГБОУ ВО АЛТАЙСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ  
Институт цифровых технологий, электроники и физики  
Кафедра вычислительной техники и электроники (ВТиЭ)

Отчёт по производственной практике: технологическая (проектно-технологическая) практика на тему:  
Telegram-бот

Выполнил студент 5.205-1 группы:  
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Шамне В.Е.  
«\_\_»\_\_\_\_\_\_\_\_\_\_\_\_2024 г.  
Проверил: Старший преподаватель  
\_\_\_\_\_\_\_\_\_\_\_\_\_\_И.А. Шмаков  
«\_\_» \_\_\_\_\_\_\_\_\_\_\_2024 г.

Барнаул 2024 г.

**РЕФЕРАТ**

Полный объем работы составляет 23 страницы, включая 3 рисунка. Данный отчёт по производственной практике: технологическая (проектно-технологическая) практика на тему: Telegram – бот включает в себя актуальность, цели, теоретическую и практическую часть создания телеграмм бота, задачи написания на языке программирования python с использованием библиотеки telebot telegram-бота, который может помочь с выбором квартиры посуточно/ почасово, используя предпочтения клиента, теоретическую часть создания telegram-бота, код программы. В работу так же включена пошаговая практическая часть создания Бота.

**ABSTRACT**

The full volume of the work is 23 pages, including 3 drawings. This report on production practice: technological (design and technological) practice on the topic: Telegram bot includes the relevance, goals, theoretical and practical part of creating telegram bots, tasks of writing in the python programming language using the telebot library telegram bot, which can help with choosing an apartment daily/ hourly, using preferences the client, the theoretical part of creating a telegram bot, the program code. The work also includes a step-by-step practical part of creating a Bot.

**СОДЕРЖАНИЕ**

[**ВВЕДЕНИЕ** 4](#_Toc164876061)

[1. ОБЗОР ТЕЛЕГРАММ ПЛАТФОРМЫ И ТЕОРЕТИЧЕСКАЯ ЧАСТЬ РАЗРАБОТКИ ТЕЛЕГРАММ БОТА 6](#_Toc164876062)

[1.1. Телеграмм платформа. 6](#_Toc164876063)

[1.2. Теоретическая часть разработки телеграмм бота 6](#_Toc164876064)

[**2.** **ПРАКТИЧЕСКАЯ ЧАСТЬ СОЗДАНИЯ ТЕЛЕГРАММ БОТА** 9](#_Toc164876065)

[2.1. Функционал телеграмм - бота 9](#_Toc164876066)

[2.2. Настройка бота 9](#_Toc164876067)

[2.3. Блок схема телеграмм бота 11](#_Toc164876068)

[2.4. Проверка телеграмм бота 12](#_Toc164876069)

[**ЗАКЛЮЧЕНИЕ** 15](#_Toc164876070)

[**СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ** 16](#_Toc164876071)

[**ПРИЛОЖЕНИЕ** 17](#_Toc164876072)

# **ВВЕДЕНИЕ**

**Актуальность практики и решение типичных задач:**

Практика и решение типичных задач являются актуальными в любой области, так как они помогают развивать навыки, улучшать результаты и повышать эффективность работы.

В бизнесе практика и решение задач помогают находить оптимальные способы развития компании, управлять ресурсами, принимать стратегические решения и достигать поставленных целей.

В науке практика и решение задач позволяют проверять гипотезы, исследовать новые технологии, разрабатывать новые методики и выявлять закономерности.

В образовании практика и решение задач способствуют углублению знаний, формированию навыков самостоятельной работы, развитию критического мышления и умения применять теоретические знания на практике.

Таким образом, практика и решение типичных задач являются ключевыми компонентами успешного прогресса в любой области, и их актуальность неизменна.

Актуальность Telegram-бота:

* Telegram-боты предоставляют удобный и быстрый способ взаимодействия между пользователями и сервисами.
* Telegram-боты могут использоваться для автоматизации различных процессов.
* Telegram-боты могут быть интегрированы с другими сервисами и платформами.
* Telegram-боты позволяют пользователям экономить время и усилия, предоставляя быстрые и удобные решения для различных задач.

Цель создания Telegram-бота, подбирающий квартиры - предоставить пользователям удобный и быстрый доступ к актуальной информации о наличии квартир в конкретном районе и на определенные даты

**Задачи:**

1. Перейти в @BotFather и получить индивидуальный токен для telegram-бота.
2. Установить в среду разработки библиотеку telebot, написать код для Бота и инициализировать его токеном, командами и функциями для обработки событий.
3. Запустить бота на сервере PythonAnyWhere для его работы после завершения сеанса в среде разработки.
4. Протестировать бота.

# ОБЗОР ТЕЛЕГРАММ ПЛАТФОРМЫ И ТЕОРЕТИЧЕСКАЯ ЧАСТЬ РАЗРАБОТКИ ТЕЛЕГРАММ БОТА

## Телеграмм платформа.

**Telegram** — это кроссплатформенная система мгновенного обмена сообщениями (мессенджер) с функциями обмена текстовыми, голосовыми и видеосообщениями, а также стикерами, фотографиями и файлами многих форматов.

**Telegram позволяет:**

* Совершать аудио и видеозвонки;
* Устраивать прямые эфиры в каналах и группах;
* Организовывать конференции, многопользовательские группы и каналы.

Также функциональность приложения можно значительно расширить при помощи ботов.

Клиентские приложения Telegram доступны для Android, iOS, Windows, macOS и GNU/Linux.

В июне 2022 года Telegram вошёл в пятёрку самых загружаемых приложений, а число его постоянных пользователей превысило 700 миллионов.

## Теоретическая часть разработки телеграмм бота

**Telegram-боты** — это мини-программы внутри мессенджера, которые управляются текстовыми командами в чате по принципу «вопрос — ответ». С 2015 года любой пользователь может сделать собственного бота в «Телеграме».

**Python -** это высокоуровневый язык программирования общего назначения с динамической строгой типизацией и автоматическим управлением памятью. Он ориентирован на повышение производительности разработчика, читаемости кода и его качества, а также на обеспечение переносимости написанных на нём программ.

В совокупности с тем, что язык программирования Python предоставляет разработчикам широкий функционал для создания ботов, а сам язык достаточно прост в изучении, то этот ЯП идеально подходит под задачи проекта.

**API ключ телеграмм бота** — это уникальный идентификационный код, который используется для доступа к API (Application Programming Interface) телеграмм бота. Этот ключ предоставляется при создании бота в Telegram и используется для аутентификации при отправке запросов к API для работы с ботом. API ключ позволяет боту взаимодействовать с пользователем и выполнять задачи, установленные разработчиком.

**Виды API ключей**:

1. Bot API key - ключ, который выдаётся при создании бота в Telegram BotFather и используется для взаимодействия с ботом через Bot API.
2. API ключ доступа - ключ, который необходим для доступа к API Telegram и выполнения различных операций, таких как отправка сообщений, управление ботом и т.д.
3. Webhook ключ - ключ, который используется для установки вебхука на ваш бот, чтобы получать обновления от Telegram в реальном времени.
4. Личный ключ аутентификации - ключ, который может быть использован для управления доступом к API вашего бота.
5. API ключ разработчика - ключ, который предоставляется разработчикам для доступа к дополнительной функциональности и инструментам Telegram API.

Для создание бота будет использоваться API ключ BotFather

1. Облачный хостинг - позволяет запускать ботов на удалённых серверах, которые принадлежат облачному провайдеру. Виртуальный хостинг предоставляет боту ресурсы виртуального сервера. Выделенный хостинг обеспечивает доступ к физическому серверу, который используется только одним ботом. В проекте используется PythonAnywhere из за его доступности

Для непосредственной разработки телеграмм бота потребуется:

1. Библиотека python-telegram-bot - официальная библиотека Python для работы с Telegram API.
2. Учетная запись в Telegram - чтобы создать бота, вам нужно будет зарегистрироваться в Telegram и получить API ключ для вашего бота.
3. Редактор кода - например, PyCharm, Visual Studio Code (для создание бота будет использоваться Pycharm, так как он имеет понятный интерфейс, предлагает автозаполнение кода и имеет более продвинутые инструменты откладки, что позитивно сказывается на результате новичков)
4. Хостинг для Telegram-ботов нужен, чтобы обеспечить постоянную доступность бота в сети. Без хостинга бот не сможет работать должным образом или вообще не будет функционировать. Хостинг предоставляет боту сервер, на котором он может обрабатывать запросы и отправлять ответы пользователям.

# **ПРАКТИЧЕСКАЯ ЧАСТЬ СОЗДАНИЯ ТЕЛЕГРАММ БОТА**

# Функционал телеграмм - бота

Для комфортного и интуитивно - понятного функционала требуется встроить в бота параметры для подбора квартир, такие как:

1. Район квартиры (индустриальный, центральный, октябрьский)
2. Количество комнат для проживания (1, 2, 3 комнаты)
3. Количество гостей, которые будут проживать в квартире (до 2-х, до 3-х, до 4-х, до 6- ти)

## Настройка бота

Создание бота через BotFather:

1. Открываем телеграмм и переходим в BotFather
2. Отправляем сообщение /newbot
3. Указываем название бота (World capital bot)
4. Указываем программное название бота (@World\_Capital2\_bot)
5. Получаем сообщение с API токеном бота ()

Программное создание бота:

1. Открываем PyCharm и скачиваем библиокеку pyTelegramBotApi, для этого в консоле вводим команду - pip install pyTelegramBotApi для того, чтобы среда разработки получила возможность взаимодействовать с API токеном, которые мы получили ранее. Ждем загрузки.
2. Подключаем библиотеку в самой среде разработки - пишем команду import telebot и from telebot import types
3. Создадим переменную Bot, записываем обращение к библиотеки telebot и обращение к классу Telebot, в этот класс помещается API токен (“”), результат: bot = telebot.TeleBot(‘ТОКЕН’)
4. Написание кода для бота
   1. Записываем декоратор для функции @bot
   2. Обращаемся к боту через команду message\_handlers
   3. Создается функция main, которая срабатывает при команде /start и отправляет сообщение с кнопками для выбора параметров.
   4. Создается обработчик для нажатия на кнопки (callback\_query\_handler) с вызовом функции callback\_message.
   5. В функции callback\_message создается разметка с кнопками "Район", "Количество гостей", "Количество комнат".
   6. Если пользователь выбирает "Район", выводится новая разметка с кнопками выбора района.
   7. Если пользователь выбирает "Количество гостей", выводится новая разметка с кнопками выбора количества гостей.
   8. Если пользователь выбирает "Количество комнат", выводится новая разметка с кнопками выбора количества комнат.
   9. В зависимости от выбранного параметра, создается соответствующее меню с вариантами выбора.
   10. Для каждого варианта выбора (район, количество гостей, количество комнат) предоставляется список ссылок на квартиры, удовлетворяющие выбранным критериям.
   11. Для каждого варианта выбора также предоставляется возможность вернуться к выбору других критериев.

## 2.3. Блок схема телеграмм бота

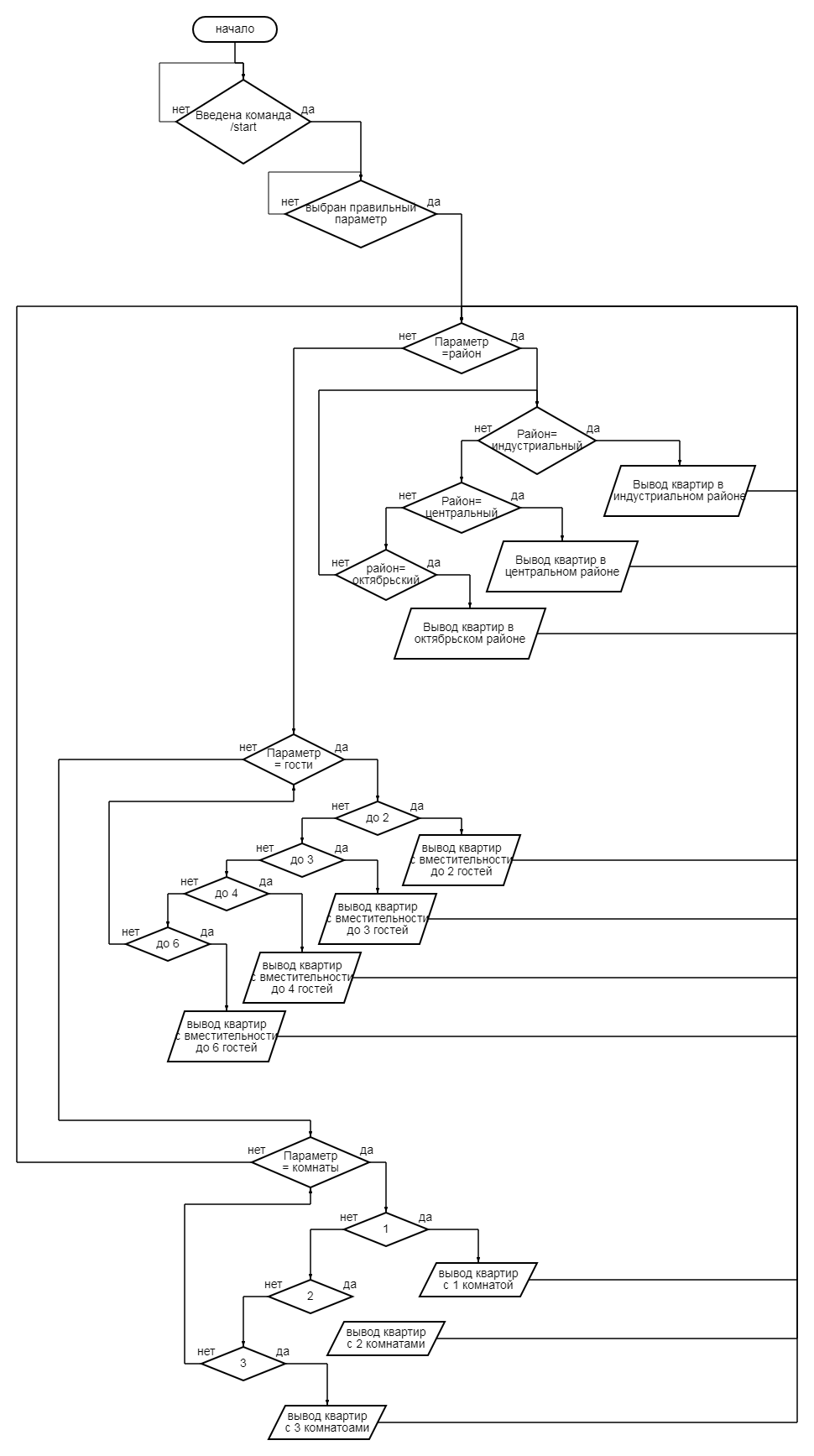
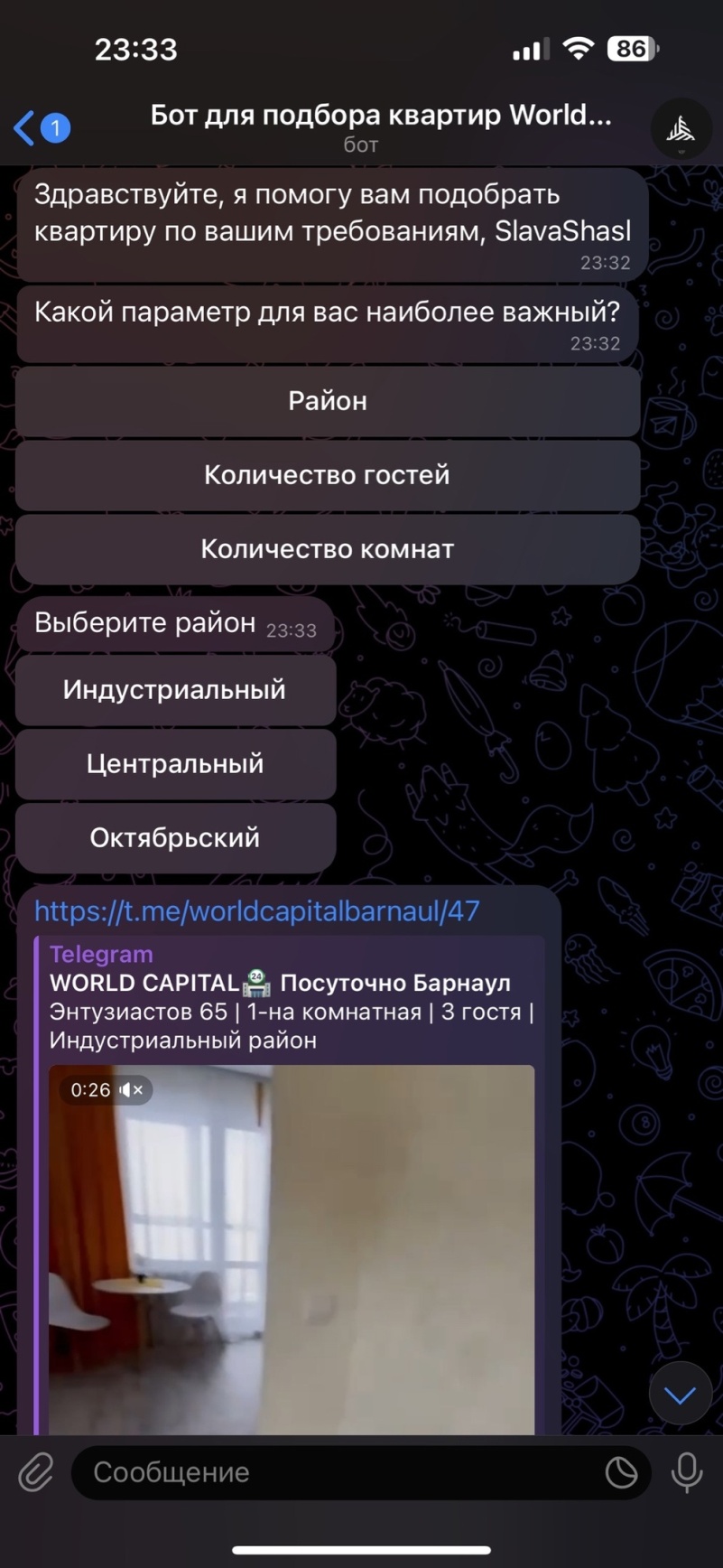
****

Рис. 1

## 2.4. Проверка телеграмм бота

**Тест 1**

1. Заходим в бота в телеграмме по имени @World\_Capital2\_bot и отпраляем /start
2. Выбираем критерий “Район”
3. Выбираем район “Индустриальный”
4. Выдается список из всех квартир в индустриальном районе с информацией о количестве комнат и допустимых гостей. Результат представлен на рисунке 2

****

**Рис.2. Результат выполнения 1 теста**

**Тест 2**

1. Заходим в бота в телеграмме по имени @World\_Capital2\_bot и отпраляем /start
2. Выбираем критерий “количество гостей”
3. Выбираем количество “до 6- ти”
4. Выдается список всех квартир, в которые может заселиться до 6- ти человек с информацией о районе и количестве комнат. Результат представлен на рисунке 3

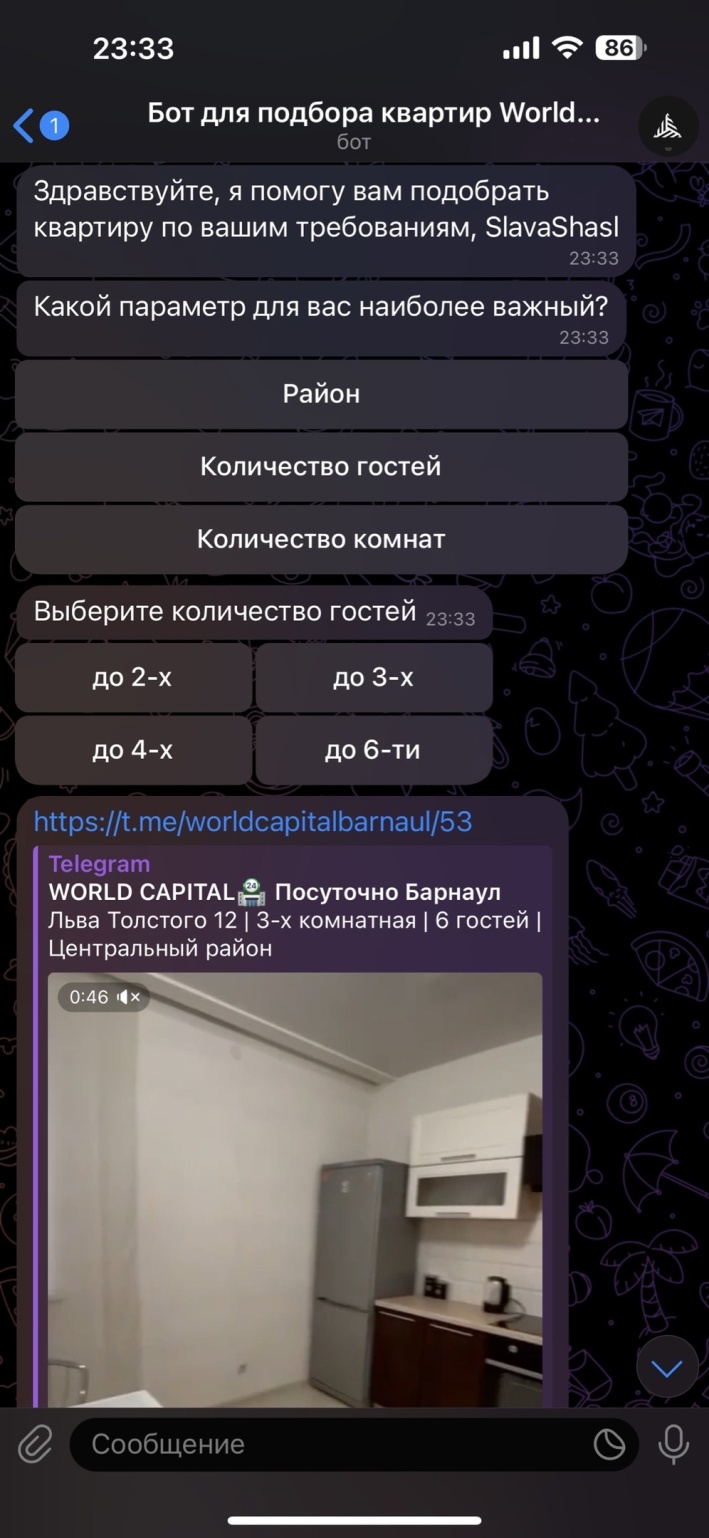


Рис.3. Результат выполнения 2 теста

**Тест 3**

1. После завершения 2 теста не закрываем бота, выбираем другой параметр “Количество комнат”
2. Выбираем количество “2”
3. Выдается список из всех квартир с двумя комнатами с информацией о количестве допустимых гостей и районе проживания. Результат представлен на рисунке 4

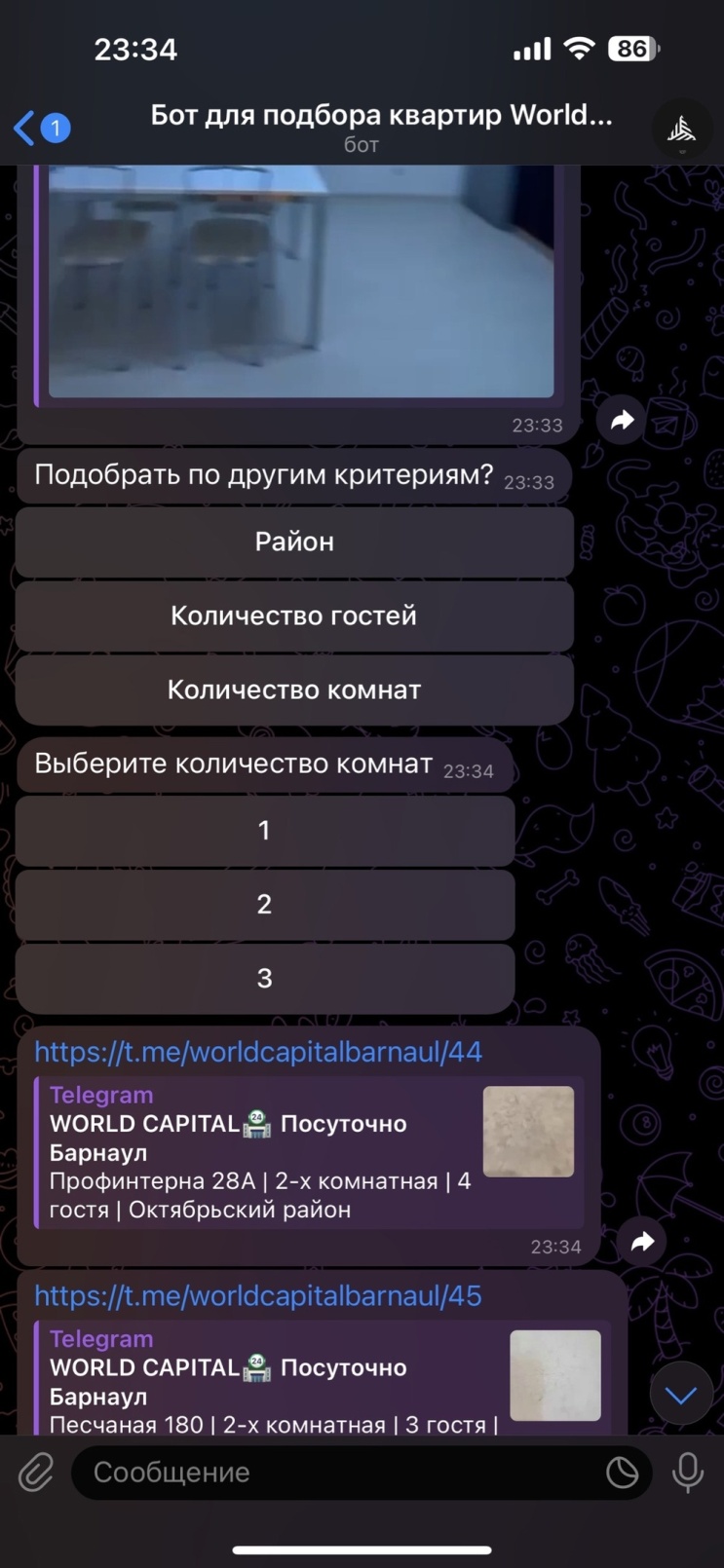


Рис.4. Результат выполнения 3 теста

# **ЗАКЛЮЧЕНИЕ**

Был создан бот, который имеет следующие преимущества по сравнению с личным подбором квартир на авито

* Удобство и доступность информации о наличии квартир (владелец, количество комнат, количество спальных мест, местоположение квартиры) в мессенджере.
* Экономия времени и оперативная помощь от арендодателя
* Проверенность арендодателя и без комиссионная оплата

Все это делает Telegram-бота, подбирающего квартиры, актуальным и полезным инструментом для пользователей.

# **СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ**

1. Все о телеграмм бота / [Электронный ресурс] // Core : [сайт]. — URL: https://core.telegram.org/bots/samples (дата обращения: 19.04.2024).
2. Как создать телеграмм бота / [Электронный ресурс] // habr : [сайт]. — URL: https://habr.com/ru/articles/442800/ (дата обращения: 19.04.2024).
3. telegram bot python / [Электронный ресурс] // Pytba : [сайт]. — URL: https://pytba.readthedocs.io/en/latest/index.html (дата обращения: 19.04.2024).
4. Создание бота с telegram API / [Электронный ресурс] // directprobi.ru : [сайт]. — URL: http://directprobi.ru/ (дата обращения: 19.04.2024).
5. Запускаем телеграмм бота на хостинке PythonAnywhere / [Электронный ресурс] // xakep : [сайт]. — URL: https://xakep.ru/2021/11/28/python-telegram-bots/ (дата обращения: 19.04.2

# ПРИЛОЖЕНИЕ

**Текст программы**

import telebot

from telebot import types

bot = telebot.TeleBot(‘токен’)

@bot.message\_handler(commands=['start'])

def main(message):

markup = types.InlineKeyboardMarkup()

markup.add(types.InlineKeyboardButton('Район', callback\_data='rayon'))

markup.add(types.InlineKeyboardButton('Количество гостей', callback\_data='gosti'))

markup.add(types.InlineKeyboardButton('Количество комнат', callback\_data='room'))

bot.send\_message(message.chat.id, f'Здравствуйте, я помогу вам подобрать квартиру по вашим требованиям, {message.from\_user.first\_name}{message.from\_user.last\_name}')

bot.send\_message(message.chat.id,f'Какой параметр для вас наиболее важный?',reply\_markup=markup)

@bot.callback\_query\_handler(func = lambda callback: True)

def callback\_message(callback):

markup = types.InlineKeyboardMarkup()

markup.add(types.InlineKeyboardButton('Район', callback\_data='rayon'))

markup.add(types.InlineKeyboardButton('Количество гостей', callback\_data='gosti'))

markup.add(types.InlineKeyboardButton('Количество комнат', callback\_data='room'))

if callback.data == 'rayon':

markup1 = types.InlineKeyboardMarkup()

markup1.add(types.InlineKeyboardButton('Индустриальный', callback\_data='industrial'))

markup1.add(types.InlineKeyboardButton('Центральный', callback\_data='centr'))

markup1.add(types.InlineKeyboardButton('Октябрьский', callback\_data='october'))

bot.send\_message(callback.message.chat.id,f'Выберите район',reply\_markup=markup1)

elif callback.data == 'gosti':

markup2 = types.InlineKeyboardMarkup()

btn1 = types.InlineKeyboardButton('до 2-х', callback\_data='2-h')

btn2 = types.InlineKeyboardButton('до 3-х', callback\_data='3-h')

markup2.row(btn1,btn2)

btn3 = types.InlineKeyboardButton('до 4-х', callback\_data='4-h')

btn4 = types.InlineKeyboardButton('до 6-ти', callback\_data='6-h')

markup2.row(btn3, btn4)

bot.send\_message(callback.message.chat.id, f'Выберите количество гостей', reply\_markup=markup2)

elif callback.data == 'room':

markup3 = types.InlineKeyboardMarkup()

markup3.add(types.InlineKeyboardButton('1', callback\_data='one'))

markup3.add(types.InlineKeyboardButton('2', callback\_data='two'))

markup3.add(types.InlineKeyboardButton('3', callback\_data='three'))

bot.send\_message(callback.message.chat.id, f'Выберите количество комнат', reply\_markup=markup3)

elif callback.data == 'industrial':

bot.send\_message(callback.message.chat.id, f'https://t.me/worldcapitalbarnaul/47')

bot.send\_message(callback.message.chat.id, f'https://t.me/worldcapitalbarnaul/48')

bot.send\_message(callback.message.chat.id, f'https://t.me/worldcapitalbarnaul/51')

bot.send\_message(callback.message.chat.id, f'https://t.me/worldcapitalbarnaul/52')

bot.send\_message(callback.message.chat.id, f'https://t.me/worldcapitalbarnaul/54')

bot.send\_message(callback.message.chat.id, f'https://t.me/worldcapitalbarnaul/49')

bot.send\_message(callback.message.chat.id, f'Подобрать по другим критериям?', reply\_markup=markup)

elif callback.data == 'october':

bot.send\_message(callback.message.chat.id, f'https://t.me/worldcapitalbarnaul/44')

bot.send\_message(callback.message.chat.id, f'Подобрать по другим критериям?', reply\_markup=markup)

elif callback.data == 'centr':

bot.send\_message(callback.message.chat.id, f'https://t.me/worldcapitalbarnaul/45')

bot.send\_message(callback.message.chat.id, f'https://t.me/worldcapitalbarnaul/46')

bot.send\_message(callback.message.chat.id, f'https://t.me/worldcapitalbarnaul/53')

bot.send\_message(callback.message.chat.id, f'https://t.me/worldcapitalbarnaul/55')

bot.send\_message(callback.message.chat.id, f'https://t.me/worldcapitalbarnaul/56')

bot.send\_message(callback.message.chat.id, f'https://t.me/worldcapitalbarnaul/57')

bot.send\_message(callback.message.chat.id, f'https://t.me/worldcapitalbarnaul/58')

bot.send\_message(callback.message.chat.id, f'Подобрать по другим критериям?',reply\_markup=markup)

elif callback.data == '2-h':

bot.send\_message(callback.message.chat.id, f'https://t.me/worldcapitalbarnaul/52')

bot.send\_message(callback.message.chat.id, f'Подобрать по другим критериям?', reply\_markup=markup)

elif callback.data == '3-h':

bot.send\_message(callback.message.chat.id, f'https://t.me/worldcapitalbarnaul/45')

bot.send\_message(callback.message.chat.id, f'https://t.me/worldcapitalbarnaul/47')

bot.send\_message(callback.message.chat.id, f'https://t.me/worldcapitalbarnaul/49')

bot.send\_message(callback.message.chat.id, f'https://t.me/worldcapitalbarnaul/56')

bot.send\_message(callback.message.chat.id, f'https://t.me/worldcapitalbarnaul/58')

bot.send\_message(callback.message.chat.id, f'Подобрать по другим критериям?',reply\_markup = markup)

elif callback.data == '4-h':

bot.send\_message(callback.message.chat.id, f'https://t.me/worldcapitalbarnaul/44')

bot.send\_message(callback.message.chat.id, f'https://t.me/worldcapitalbarnaul/46')

bot.send\_message(callback.message.chat.id, f'https://t.me/worldcapitalbarnaul/48')

bot.send\_message(callback.message.chat.id, f'https://t.me/worldcapitalbarnaul/51')

bot.send\_message(callback.message.chat.id, f'https://t.me/worldcapitalbarnaul/54')

bot.send\_message(callback.message.chat.id, f'https://t.me/worldcapitalbarnaul/55')

bot.send\_message(callback.message.chat.id, f'https://t.me/worldcapitalbarnaul/57')

bot.send\_message(callback.message.chat.id, f'Подобрать по другим критериям?', reply\_markup=markup)

elif callback.data == '6-h':

bot.send\_message(callback.message.chat.id, f'https://t.me/worldcapitalbarnaul/53')

bot.send\_message(callback.message.chat.id, f'Подобрать по другим критериям?', reply\_markup=markup)

elif callback.data == 'one':

bot.send\_message(callback.message.chat.id, f'https://t.me/worldcapitalbarnaul/47')

bot.send\_message(callback.message.chat.id, f'https://t.me/worldcapitalbarnaul/48')

bot.send\_message(callback.message.chat.id, f'https://t.me/worldcapitalbarnaul/49')

bot.send\_message(callback.message.chat.id, f'https://t.me/worldcapitalbarnaul/51')

bot.send\_message(callback.message.chat.id, f'https://t.me/worldcapitalbarnaul/52')

bot.send\_message(callback.message.chat.id, f'https://t.me/worldcapitalbarnaul/55')

bot.send\_message(callback.message.chat.id, f'https://t.me/worldcapitalbarnaul/56')

bot.send\_message(callback.message.chat.id, f'https://t.me/worldcapitalbarnaul/57')

bot.send\_message(callback.message.chat.id, f'https://t.me/worldcapitalbarnaul/58')

bot.send\_message(callback.message.chat.id, f'Подобрать по другим критериям?', reply\_markup=markup)

elif callback.data == 'two':

bot.send\_message(callback.message.chat.id, f'https://t.me/worldcapitalbarnaul/44')

bot.send\_message(callback.message.chat.id, f'https://t.me/worldcapitalbarnaul/45')

bot.send\_message(callback.message.chat.id, f'https://t.me/worldcapitalbarnaul/46')

bot.send\_message(callback.message.chat.id, f'https://t.me/worldcapitalbarnaul/54')

bot.send\_message(callback.message.chat.id, f'Подобрать по другим критериям?', reply\_markup=markup)

elif callback.data == 'three':

bot.send\_message(callback.message.chat.id, f'https://t.me/worldcapitalbarnaul/53')

bot.send\_message(callback.message.chat.id, f'Подобрать по другим критериям?', reply\_markup=markup)

@bot.message\_handler()

def chat(message):

if message.text.lower() == 'привет':

markup = types.InlineKeyboardMarkup()

markup.add(types.InlineKeyboardButton('Район', callback\_data='rayon'))

markup.add(types.InlineKeyboardButton('Количество гостей', callback\_data='gosti'))

markup.add(types.InlineKeyboardButton('Количество комнат', callback\_data='room'))

bot.send\_message(message.chat.id,f'Здравствуйте, я помогу вам подобрать квартиру по вашим требованиям, {message.from\_user.first\_name}{message.from\_user.last\_name}')

bot.send\_message(message.chat.id, f'Какой параметр для вас наиболее важный?',reply\_markup=markup)

if message.text.lower() == 'старт':

markup = types.InlineKeyboardMarkup()

markup.add(types.InlineKeyboardButton('Район', callback\_data='rayon'))

markup.add(types.InlineKeyboardButton('Количество гостей', callback\_data='gosti'))

markup.add(types.InlineKeyboardButton('Количество комнат', callback\_data='room'))

bot.send\_message(message.chat.id,f'Здравствуйте, я помогу вам подобрать квартиру по вашим требованиям, {message.from\_user.first\_name}{message.from\_user.last\_name}')

bot.send\_message(message.chat.id, f'Какой параметр для вас наиболее важный?',reply\_markup=markup)

bot.polling(none\_stop=True)