Министерство просвещения Российской Федерации

Муниципальное автономное общеобразовательное учреждение

«Средняя общеобразовательная школа № 76 имени Д.Е. Васильева»

II Международный конкурс исследовательских работ школьников

«Research start» 2024/2025

Проектная работа

**Телеграмм-бот «Подготовка к ОГЭ»**

Выполнил: Фролов Арсений Дмитриевич

ученик 11 класса

Руководитель: Ф.И.О.

учитель Иванова Елена Петровна

2024

**Оглавление**

[Введение 3](#_Toc181185570)

[Глава I. Инструменты разработчика 5](#_Toc181185571)

[Глава II. Разработка чат-бота «Подготовка к ОГЭ» 8](#_Toc181185572)

[Описание бота 8](#_Toc181185573)

[Алгоритм создания бота 8](#_Toc181185574)

[Аналоги бота 10](#_Toc181185575)

[Анкетирование учащихся 10](#_Toc181185576)

[Заключение (Выводы) 12](#_Toc181185577)

[Список литературы и Интернет-источников 13](#_Toc181185578)

[Приложение 1. Код чат-бота 14](#_Toc181185579)

[Приложение 2. Скриншоты чат-бота 23](#_Toc181185580)

# **Введение**

В век информационных технологий общество очень нуждается в грамотных специалистах в сфере IT. В нашей стране взрослыми активно поддерживается развитие инженерного мышления у детей. Предлагается множество различных мероприятий в этом направлении. Думаю, что кроме участия в различных мероприятиях, развивающих техническое творчество, инженерное мышление, молодым людям, кто в будущем хотел бы стать программистами, важно уметь выходить за рамки школьной программы по информатике, изучать самостоятельно актуальные языки программирования, создавая программные продукты.

***Актуальность***

Язык программирования Python достаточно востребован в обществе. На этом языке создаются приложения, чат-боты и многое другое. Боты Телеграм универсальны и выполняют множество задач на любой вкус. Найти информацию, скачать книгу, фильм, музыку, развлечь — это лишь маленький перечень возможностей ботов.

В прошлом учебном году я и мои друзья-одноклассники сдавали ОГЭ. Я решил создать бота, который поможет подготовиться к ОГЭ по информатике (тестовая часть), математике, физике и русскому языку. Времени на подготовку всегда не хватает, к компьютеру не всегда есть доступ, а телефон обычно под рукой, поэтому девятиклассник может использовать бот тогда, когда удобно ему (на перемене в школе, в дороге, в очереди и т.д.)

***Проблемы***

1. Общеизвестно, что изучать любой язык программирования трудно. В школьном курсе при изучении темы «Программирование» не всем детям понятно, для чего она изучается. А если обучаться программированию, параллельно создавая свой продукт, например, чат-бот, то процесс обучения программированию протекает легче и человеку сразу виден результат.
2. Различных чат-ботов много, но нет такого, который бы помог самостоятельно подготовиться к ОГЭ с большим количеством различных заданий, решением этих заданий и видеоразбором основных типов заданий.

***Новизна*** проекта заключается в том, что подобного бота не существует. При использовании бота необходимо отвечать на вопросы по десяти вопросам тестовой части ОГЭ по информатике, смотреть видеоразборы и решения.

***Гипотеза.*** Зная основные алгоритмические конструкции (следование, ветвление и цикл) и обладая навыками создания программ на учебном языке программирования (например, Pascal, Python), можно самостоятельно научиться программировать на более высоком уровне, создав телеграмм-бота.

***Цель проекта***

Создать телеграмм-бота на языке программирования Python для подготовки к ОГЭ по информатике (тестовая часть) математике, физике и русскому языку.

Для достижения поставленной цели необходимо решить следующие **задачи:**

1. изучить библиотеку telebot;
2. познакомиться с ботом @BotFather
3. применить полученные знания для моделирования бота;
4. подготовить задания с сайта «Решу ОГЭ» в необходимом формате;
5. протестировать бота;
6. показать продукт одноклассникам.

В проекте использованы такие ***методы*** как***:*** моделирование; метод визуализации данных.

# **Глава I. Инструменты разработчика**

Для разработки чат-бота необходимы были следующие 4 инструмента.

**1****. PyCharm**

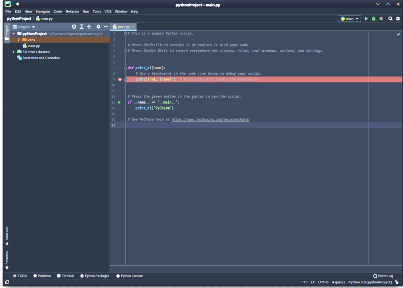
PyCharm - это интегрированная среда разработки для языка программирования Python, разработанная компанией JetBrains на основе IntelliJ IDEA. Предоставляет пользователю комплекс средств для написания кода и визуальный отладчик. [1]

Рисунок Скрин PyCharm

**2. BotFather**

BotFather – официальный сервис для настройки и управления созданными телеграм-ботами. «бот над всеми ботами» в Телеграмме.

BotFather представляет собой интерфейс доступа к API (Application Programming Interface) мессенджера Telegram. С помощью BotFather можно создавать ботов и управлять существующими. [2]

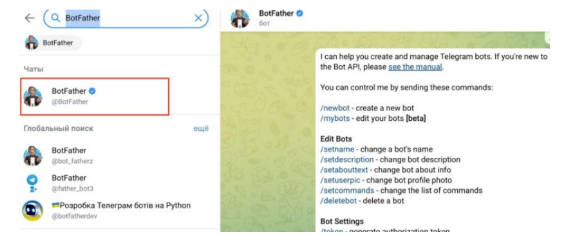


Рисунок Скрин BotFather

**3. Сайт «Решу ОГЭ»**

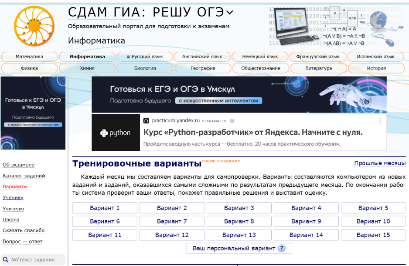


Рисунок Сайт "Решу ОГЭ"

Сайт «Решу ОГЭ» - это база заданий с решениями для подготовки к ОГЭ по всем предметам, система тестов для подготовки и самоподготовки к ОГЭ. [3]

**4. Telegram Messenger**

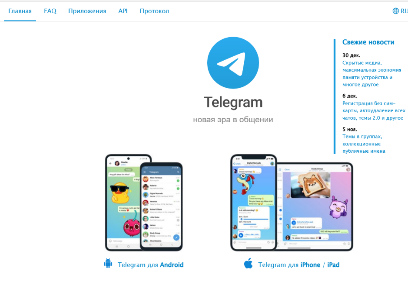


Рисунок Мессенжер Telegram

Telegram (от др.-греч. τῆλε «далеко» + др.-греч. γράμμα «запись») — кроссплатформенная система мгновенного обмена сообщениями (мессенджер) с функциями обмениваться текстовыми, голосовыми и видеосообщениями, стикерами и фотографиями, файлами многих форматов. С помощью ботов функционал приложения практически не ограничен. Клиентские приложения Telegram доступны для Android, iOS, Windows, macOS и GNU/Linux. В июне 2022 года вошёл в пятёрку самых загружаемых приложений, а число его постоянных пользователей превысило 700 миллионов. [4]

**5. Timeweb**

Timeweb – это российский хостинг, отличающийся стабильным функционированием, безопасностью, быстродействием и множеством предложений для размещений сайтов. [5]

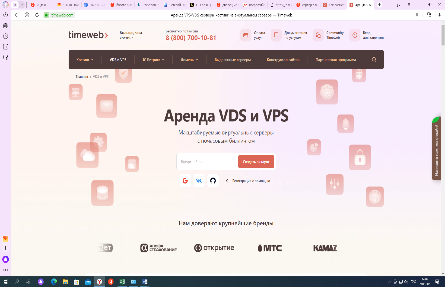


Рисунок Хостинг Timeweb

# **Глава II. Разработка чат-бота «Подготовка к ОГЭ»**

## **Описание бота**

Чат-бот предназначен для подростков 8-9 классов, которым нужно подготовиться к ОГЭ. Чат-бот можно использовать для самоконтроля знаний по тестовой части ОГЭ по информатике, физике, математике и русскому языку и другим предметам.

Рисунок QR-код бота "Подготовка к ОГЭ"

Задания взяты с сайта «Решу ОГЭ». Нарушения авторских прав нет, так как задания находятся в открытом доступе сети Internet и бот не является коммерческим продуктом, а используется в учебных целях. Также было написано письмо администратору сайта, и получен положительный ответ.

По каждому из типов заданий тестовой части имеется 35 вариантов.

Пользователь может решать выбранные задания из ОГЭ, знакомиться с ответами и решениями, а также просматривать обучающее видео.

## **Алгоритм создания бота**

В данном разделе приводится алгоритм создания программного кода бота, который можно разделить на 5 частей.

1) Создание бота в BotFather

2) Копирование токена и вставка его в код в PyCharm

3) Программирование бота

4) Тестирование и отладка бота

5) Формирование банка заданий с сайта «Решу ОГЭ» и вставка его в код

6) Подбор видео-материала, который расскажет подробно как решать определенный тип задания (использовался видеохостинг <https://www.youtube.com> )

7) Размещение бота на хостинге Timeweb

Если говорить о структуре, то каждая задача прописана в отдельном файле. Когда пользователь выбирает номер задания, то подключается блок кода, отвечающий за это задание. Это сделано для того, чтобы код работал быстрее. В таком случае нет необходимости обрабатывать весь код сразу, файлы подключаются по необходимости.

Все сообщения бота выведены в отдельные переменные, которые соответствуют случайному элементу списка реакций на ответ пользователя. Разделение на переменные помогло мне, как разработчику, оперативнее отладить код, если в нем обнаружена ошибка при тестировании.

В боте имеется база данных. В нее сохраняются данные пользователя: его ID, имя, количество баллов. База данных находится локально на сервере, для ее создания был использован язык SQL.

В основе кода бота лежат две функции. Опишу каждую функцию:

1. Функция проверки. Чат-бот получает от пользователя ответ на вопрос, проверяет его и выводит свой «вердикт» (правильно выполнено задание или нет). Если задание выполнено правильно, то чат-бот дает возможность выбирать пользователю следующие задание. Если же задание выполнено не верно, то чат-бот предлагает сделать выбор: решать задание еще раз, посмотреть видео-решение или посмотреть ответ.

2. Функция вывода решения. Чат-бот отправляет данные, которые получил по выполнению первой функции во вторую функцию.Эта функция выводит решение,следующие задание или видеоразбор.Перед созданием бота были просмотрены и выбраны видеоразборы с Youtube[6],[7],[8].

Адрес чат-бота в Telegram: @OGE\_TESTBot. Код бота представлен в Приложении 1. Некоторые скриншоты бота представлены в Приложении 2.

## **Аналоги бота**

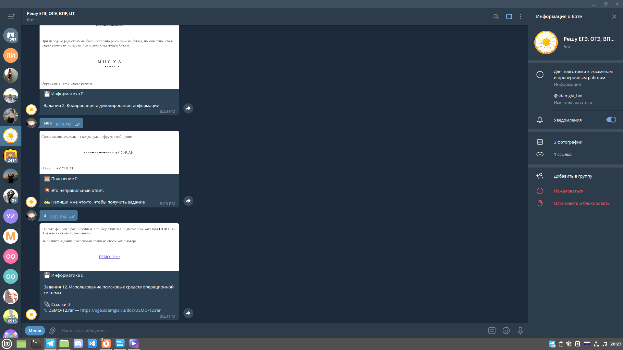
 Проанализировав боты по подготовке к ОГЭ, можно сделать следующие выводы: большинство ботов не работает (не поддерживает разработчик или неисправный код); во многих ботах есть такая проблема: чтобы начать выполнять задания нужно подписаться на множество различных групп. Выделю одного бота «Решу ЕГЭ, ОГЭ, ВПР». Этот бот подходит для подготовки к ОГЭ, но сначала его нужно настроить, а это займет немало времени. Пользователя «перебрасывает» на сайт, в котором нужно выбрать предмет, затем будут выводиться задания без возможности выбрать какое-то определенное (номер задания).

Рисунок Скрин бота "Решу ОГЭ,ЕГЭ, ВПР"

## **Анкетирование учащихся**

После того как бот был создан, его тестирование было предоставлено девятиклассникам школы, сдающим ОГЭ по информатике. После тестирования было выявлено несколько ошибок. Они были исправлены, рекомендации учтены. Для получения обратной связи было проведено анкетирование. Результаты анкетирования представлены в диаграммах ниже.

Рисунок Диаграмма 1

В нашей школе более ста девятиклассников. Около 80 процентов оценили удобный интерфейс, 78 процентов сказали, что 35 вариантов задний достаточно для подготовки к ОГЭ, почти все ребята отметили наличие видео-разбора в боте. Можно сделать вывод, что данный Telegram-бот актуален и полезен для девятиклассников, и больше половины учащихся собираются его использовать для подготовки к ОГЭ.

Рисунок Диаграмма 4

Рисунок Диаграмма 3

Рисунок Диаграмма 2

# **Заключение (Выводы)**

Задачи проекта выполнены, цель работы достигнута: создан чат-бот в Telegram «Подготовка к ОГЭ».

Чат-бота протестировали девятиклассники школы и выразили заинтересованность в его использовании.

Итак, зная основные алгоритмические конструкции (следование, ветвление и цикл) и обладая навыками создания программ на учебном языке программирования, можно самостоятельно научиться программировать на языке Python на более высоком уровне, создавая чат-бота. Гипотеза подтверждена.

Думаю, данный продукт будет полезен тем, кто собирается сдавать ОГЭ по информатике, математике, физике и русскому языку.

**Перспективы проекта:**

* Добавить новые предметы
* Увеличить количество заданий.
* Добавить различные бонусы за прохождение каждого задания
* Добавить модератора(бот) чата, который будет просматривать каждое сообщение и проверять его на наличие нецензурной лексики и удалять это сообщение.
* Разместить код бота на GitHab который будет доступен всем

# **Список литературы и Интернет-источников**

1. <https://ru.wikipedia.org/wiki/PyCharm> Статья «PyCharm» (Википедия). Свободная энциклопедия [Электронный ресурс]. Режим доступа: свободный (дата обращения 10.10.22).
2. <https://chatlabs.ru/botfather-instrukcziya-komandy-nastrojki/?ysclid=ldjuzkh8us379162125> Статья «[Полная инструкция по BotFather: токен, команды, платежи (chatlabs.ru)](https://chatlabs.ru/botfather-instrukcziya-komandy-nastrojki/?ysclid=ldjuzkh8us379162125)» (сайт «Создаем бота в BotFather») [Электронный ресурс]. Режим доступа: свободный (дата обращения 25.10.22).
3. <https://inf-oge.sdamgia.ru/> Сайт «Решу ОГЭ » [Электронный ресурс]. Режим доступа: свободный (дата обращения 29.11.22).
4. <https://ru.wikipedia.org/wiki/Telegram> Статья «Telegram» (Википедия). Свободная энциклопедия [Электронный ресурс]. Режим доступа: свободный (дата обращения 25.01.23).
5. <https://timeweb.com/ru/> Сайт “Timeweb“ [Электронный ресурс]. Режим доступа: свободный (дата обращения 29.01.23).
6. <https://habr.com/ru/post/580408/> Статья «модуль telebot» [Электронный ресурс]. Режим доступа: свободный (дата обращения 31.01.23).
7. <https://www.youtube.com/watch?v=HodO2eBEz_8&t=746s> Видео ресурс обучение как написать телеграмм бота используя модуль tekebot [Электронный ресурс]. Режим доступа: свободный (дата обращения 25.10.22).
8. <https://www.youtube.com/watch?v=M8fhrtvedHA&t=521s> Видео ресурс обучение как написать телеграмм бота используя модуль tekebot [Электронный ресурс]. Режим доступа: свободный (дата обращения 25.10.22).

# **Приложение 1.** **Код чат-бота**

**Первая функция. Проверяет правильность ответа:**

def reference\_6\_variable\_8(message):

'''задание 6 переменная 8'''

if message.text == '8':

markup = types.ReplyKeyboardMarkup(one\_time\_keyboard=True, resize\_keyboard=True, row\_width=1)

item\_solve\_a\_specific\_task = types.InlineKeyboardButton(text='Решить задачу')

item\_solve\_the\_demo\_test = types.InlineKeyboardButton(text='Решить ДЕМО тест')

item\_start\_training = types.InlineKeyboardButton(text='Начать обучение')

markup.add(item\_solve\_a\_specific\_task, item\_solve\_the\_demo\_test, item\_start\_training)

msg = bot.send\_message(message.chat.id,

'Крутяк, с победой!!! Вы решили правильно 6 задание. Можете выбрать что будете делать дальше', reply\_markup=markup)

bot.register\_next\_step\_handler(msg, gl\_buttons\_after\_an\_incorrect\_answer\_to\_a\_question)

else:

markup = types.ReplyKeyboardMarkup(one\_time\_keyboard=True, resize\_keyboard=True)

item\_want = types.InlineKeyboardButton(text='Хочу')

item\_not\_want = types.InlineKeyboardButton(text='Не хочу')

item\_view\_the\_solution = types.InlineKeyboardButton(text='Посмотреть решение')

item\_solve\_the\_problem = types.InlineKeyboardButton(text='Решить сначала')

markup.add(item\_want, item\_not\_want, item\_solve\_the\_problem, item\_view\_the\_solution)

msg = bot.send\_message(message.chat.id,

'Вы написали неправильный ответ 😢. Советую посмотреть ролик, для того что бы понять как решать подобные задачи. Посмотришь?', reply\_markup=markup)

bot.register\_next\_step\_handler(msg, checking\_wrong\_answer\_6\_variable\_8)

**Вторая функция. Выводи решение, видео материал или выводит следующие задание:**

def checking\_wrong\_answer\_6\_variable\_8(message):

''''проверка неправильного вопроса на 6 задание 8 переменная'''

if message.text == 'Посмотреть решение':

markup = types.ReplyKeyboardMarkup(one\_time\_keyboard=True, resize\_keyboard=True, row\_width=1)

item\_solve\_a\_specific\_task = types.InlineKeyboardButton(text='Решить задачу')

markup.add(item\_solve\_a\_specific\_task)

msg = bot.send\_message(message.chat.id, 'Заметим, что программа напечатает «YES», если хотя бы одна из введённых переменных s и t будет меньше 10. Значит, было 8 запусков, при которых программа напечатала «YES». В качестве значений переменных s и t в этих случаях вводились следующие пары чисел:\n(6, 4); (7, 8); (8, 5); (5, 6); (–5, 7); (–2, 2); (4, 5); (8, 6).\nОтвет: 8.', reply\_markup=markup)

bot.register\_next\_step\_handler(msg, gl\_buttons\_after\_an\_incorrect\_answer\_to\_a\_question)

if message.text == 'Хочу':

bot.send\_message(message.chat.id, 'Круто, держи ролик https://clck.ru/33RgBJ')

markup = types.ReplyKeyboardMarkup(one\_time\_keyboard=True, resize\_keyboard=True, row\_width=1)

item\_solve\_a\_specific\_task = types.InlineKeyboardButton(text='Решить задачу')

item\_solve\_the\_demo\_test = types.InlineKeyboardButton(text='Решить ДЕМО тест')

item\_start\_training = types.InlineKeyboardButton(text='Начать обучение')

markup.add(item\_solve\_a\_specific\_task, item\_solve\_the\_demo\_test, item\_start\_training)

msg = bot.send\_message(message.chat.id, 'После ролика выбирай что будешь делать дальше', reply\_markup=markup)

bot.register\_next\_step\_handler(msg, gl\_buttons\_after\_an\_incorrect\_answer\_to\_a\_question)

if message.text == 'Не хочу':

markup = types.ReplyKeyboardMarkup(one\_time\_keyboard=True, resize\_keyboard=True, row\_width=1)

item\_solve\_a\_specific\_task = types.InlineKeyboardButton(text='Решить задачу')

markup.add(item\_solve\_a\_specific\_task)

msg = bot.send\_message(message.chat.id, 'Жаль в этот раз не получилось, тогда можешь выполнить что-то из предложенного ниже', reply\_markup=markup)

bot.register\_next\_step\_handler(msg, gl\_buttons\_after\_an\_incorrect\_answer\_to\_a\_question)

if message.text == 'Решить сначала':

bot.send\_message(message.chat.id, 'На этот раз желаю удачи 😇, можете порешать еще раз')

with open('/home/arseniy/Applications/PythonProject/TG bot/Photo/6\_8.png', 'rb') as file:

msg = bot.send\_photo(message.chat.id, file)

bot.register\_next\_step\_handler(msg, reference\_6\_variable\_8)

***Вывод задания***

def checking\_the\_task\_number\_info(message):

'''проверка введенного номера задания'''

if message.text == 'Номер 6':

if gl\_variable == 1:

with open('/home/arseniy/Applications/PythonProject/TG bot/Photo/6\_1.png', 'rb') as file:

msg = bot.send\_photo(message.chat.id, file)

bot.register\_next\_step\_handler(msg, reference\_6\_variable\_1)

if gl\_variable == 2:

with open('/home/arseniy/Applications/PythonProject/TG bot/Photo/6\_2.png', 'rb') as file:

msg = bot.send\_photo(message.chat.id, file)

bot.register\_next\_step\_handler(msg, reference\_6\_variable\_2)

if gl\_variable == 3:

with open('/home/arseniy/Applications/PythonProject/TG bot/Photo/6\_3.png', 'rb') as file:

msg = bot.send\_photo(message.chat.id, file)

bot.register\_next\_step\_handler(msg, reference\_6\_variable\_3)

if gl\_variable == 4:

with open('/home/arseniy/Applications/PythonProject/TG bot/Photo/6\_4.png', 'rb') as file:

msg = bot.send\_photo(message.chat.id, file)

bot.register\_next\_step\_handler(msg, reference\_6\_variable\_4)

if gl\_variable == 5:

with open('/home/arseniy/Applications/PythonProject/TG bot/Photo/6\_5.png', 'rb') as file:

msg = bot.send\_photo(message.chat.id, file)

bot.register\_next\_step\_handler(msg, reference\_6\_variable\_5)

if gl\_variable == 6:

with open('/home/arseniy/Applications/PythonProject/TG bot/Photo/6\_6.png', 'rb') as file:

msg = bot.send\_photo(message.chat.id, file)

bot.register\_next\_step\_handler(msg, reference\_6\_variable\_6)

if gl\_variable == 7:

with open('/home/arseniy/Applications/PythonProject/TG bot/Photo/6\_7.png', 'rb') as file:

msg = bot.send\_photo(message.chat.id, file)

bot.register\_next\_step\_handler(msg, reference\_6\_variable\_7)

if gl\_variable == 8:

with open('/home/arseniy/Applications/PythonProject/TG bot/Photo/6\_8.png', 'rb') as file:

msg = bot.send\_photo(message.chat.id, file)

bot.register\_next\_step\_handler(msg, reference\_6\_variable\_8)

if gl\_variable == 9:

with open('/home/arseniy/Applications/PythonProject/TG bot/Photo/6\_9.png', 'rb') as file:

msg = bot.send\_photo(message.chat.id, file)

bot.register\_next\_step\_handler(msg, reference\_6\_variable\_9)

if gl\_variable == 10:

with open('/home/arseniy/Applications/PythonProject/TG bot/Photo/6\_10.png', 'rb')as file:

msg = bot.send\_photo(message.chat.id, file)

bot.register\_next\_step\_handler(msg, reference\_6\_variable\_10)

if gl\_variable == 11:

with open('/home/arseniy/Applications/PythonProject/TG bot/Photo/6\_11.png', 'rb')as file:

msg = bot.send\_photo(message.chat.id, file)

bot.register\_next\_step\_handler(msg, reference\_6\_variable\_11)

if gl\_variable == 12:

with open('/home/arseniy/Applications/PythonProject/TG bot/Photo/6\_12.png', 'rb')as file:

msg = bot.send\_photo(message.chat.id, file)

bot.register\_next\_step\_handler(msg, reference\_6\_variable\_12)

if gl\_variable == 13:

with open('/home/arseniy/Applications/PythonProject/TG bot/Photo/6\_13.png', 'rb')as file:

msg = bot.send\_photo(message.chat.id, file)

bot.register\_next\_step\_handler(msg, reference\_6\_variable\_13)

if gl\_variable == 14:

with open('/home/arseniy/Applications/PythonProject/TG bot/Photo/6\_14.png', 'rb')as file:

msg = bot.send\_photo(message.chat.id, file)

bot.register\_next\_step\_handler(msg, reference\_6\_variable\_14)

if gl\_variable == 15:

with open('/home/arseniy/Applications/PythonProject/TG bot/Photo/6\_15.png', 'rb') as file:

msg = bot.send\_photo(message.chat.id, file)

bot.register\_next\_step\_handler(msg, reference\_6\_variable\_15)

if gl\_variable == 16:

with open('/home/arseniy/Applications/PythonProject/TG bot/Photo/6\_16.png', 'rb') as file:

msg = bot.send\_photo(message.chat.id, file)

bot.register\_next\_step\_handler(msg, reference\_6\_variable\_16)

if gl\_variable == 17:

with open('/home/arseniy/Applications/PythonProject/TG bot/Photo/6\_17.png', 'rb') as file:

msg = bot.send\_photo(message.chat.id, file)

bot.register\_next\_step\_handler(msg, reference\_6\_variable\_17)

if gl\_variable == 18:

with open('/home/arseniy/Applications/PythonProject/TG bot/Photo/6\_18.png', 'rb') as file:

msg = bot.send\_photo(message.chat.id, file)

bot.register\_next\_step\_handler(msg, reference\_6\_variable\_18)

if gl\_variable == 19:

with open('/home/arseniy/Applications/PythonProject/TG bot/Photo/6\_19.png', 'rb') as file:

msg = bot.send\_photo(message.chat.id, file)

bot.register\_next\_step\_handler(msg, reference\_6\_variable\_19)

if gl\_variable == 20:

with open('/home/arseniy/Applications/PythonProject/TG bot/Photo/6\_20.png', 'rb') as file:

msg = bot.send\_photo(message.chat.id, file)

bot.register\_next\_step\_handler(msg, reference\_6\_variable\_20)

if gl\_variable == 21:

with open('/home/arseniy/Applications/PythonProject/TG bot/Photo/6\_21.png', 'rb') as file:

msg = bot.send\_photo(message.chat.id, file)

bot.register\_next\_step\_handler(msg, reference\_6\_variable\_21)

if gl\_variable == 22:

with open('/home/arseniy/Applications/PythonProject/TG bot/Photo/6\_22.png', 'rb') as file:

msg = bot.send\_photo(message.chat.id, file)

bot.register\_next\_step\_handler(msg, reference\_6\_variable\_22)

if gl\_variable == 23:

with open('/home/arseniy/Applications/PythonProject/TG bot/Photo/6\_23.png', 'rb') as file:

msg = bot.send\_photo(message.chat.id, file)

bot.register\_next\_step\_handler(msg, reference\_6\_variable\_23)

if gl\_variable == 24:

with open('/home/arseniy/Applications/PythonProject/TG bot/Photo/6\_24.png', 'rb') as file:

msg = bot.send\_photo(message.chat.id, file)

bot.register\_next\_step\_handler(msg, reference\_6\_variable\_24)

if gl\_variable == 25:

with open('/home/arseniy/Applications/PythonProject/TG bot/Photo/6\_25.png', 'rb') as file:

msg = bot.send\_photo(message.chat.id, file)

bot.register\_next\_step\_handler(msg, reference\_6\_variable\_25)

if gl\_variable == 26:

with open('/home/arseniy/Applications/PythonProject/TG bot/Photo/6\_26.png', 'rb') as file:

msg = bot.send\_photo(message.chat.id, file)

bot.register\_next\_step\_handler(msg, reference\_6\_variable\_26)

if gl\_variable == 27:

with open('/home/arseniy/Applications/PythonProject/TG bot/Photo/6\_27.png', 'rb') as file:

msg = bot.send\_photo(message.chat.id, file)

bot.register\_next\_step\_handler(msg, reference\_6\_variable\_27)

if gl\_variable == 28:

with open('/home/arseniy/Applications/PythonProject/TG bot/Photo/6\_28.png', 'rb') as file:

msg = bot.send\_photo(message.chat.id, file)

bot.register\_next\_step\_handler(msg, reference\_6\_variable\_28)

if gl\_variable == 29:

with open('/home/arseniy/Applications/PythonProject/TG bot/Photo/6\_29.png', 'rb') as file:

msg = bot.send\_photo(message.chat.id, file)

bot.register\_next\_step\_handler(msg, reference\_6\_variable\_29)

if gl\_variable == 30:

with open('/home/arseniy/Applications/PythonProject/TG bot/Photo/6\_30.png', 'rb') as file:

msg = bot.send\_photo(message.chat.id, file)

bot.register\_next\_step\_handler(msg, reference\_6\_variable\_30)

if gl\_variable == 31:

with open('/home/arseniy/Applications/PythonProject/TG bot/Photo/6\_31.png', 'rb') as file:

msg = bot.send\_photo(message.chat.id, file)

bot.register\_next\_step\_handler(msg, reference\_6\_variable\_31)

if gl\_variable == 32:

with open('/home/arseniy/Applications/PythonProject/TG bot/Photo/6\_32.png', 'rb') as file:

msg = bot.send\_photo(message.chat.id, file)

bot.register\_next\_step\_handler(msg, reference\_6\_variable\_32)

if gl\_variable == 33:

with open('/home/arseniy/Applications/PythonProject/TG bot/Photo/6\_33.png', 'rb') as file:

msg = bot.send\_photo(message.chat.id, file)

bot.register\_next\_step\_handler(msg, reference\_6\_variable\_33)

if gl\_variable == 34:

with open('/home/arseniy/Applications/PythonProject/TG bot/Photo/6\_34.png', 'rb') as file:

msg = bot.send\_photo(message.chat.id, file)

bot.register\_next\_step\_handler(msg, reference\_6\_variable\_34)

if gl\_variable == 35:

with open('/home/arseniy/Applications/PythonProject/TG bot/Photo/6\_35.png', 'rb') as file:

msg = bot.send\_photo(message.chat.id, file)

bot.register\_next\_step\_handler(msg, reference\_6\_variable\_35)

**Адрес чат-бота в Telegram: *@OGE\_TESTBot***

# **Приложение 2. Скриншоты чат-бота**

