

Муниципальное общеобразовательное
учреждение
«Стрелецкая средняя общеобразовательная
школа
Белгородского района Белгородской области
имени Героя Советского Союза А.Е.
Черникова»

«ТЕЛЕГРАММ-БОТ ДОСТОПРИМЕЧАТЕЛЬНОСТЕЙ БЕЛГОРОДСКОЙ ОБЛАСТИ »

Выполнила: Смирнова Ксения
учащаяся 10 класса
Руководитель: Клименко
Татьяна Петровна,
учитель английского языка.

Цели и задачи:

- ▣ **Цель проекта:** создание телеграмм-бота достопримечательностей Белгородской области.

- ▣ **Задачи:**
 1. Определить понятие телеграмм-бота
 2. Провести анкетирование среди одноклассников
 3. Разработать код бота
 4. Обеспечить постоянную работу бота по средствам хостингов.
 5. Систематизировать изученный материал и сделать выводы

Какие бывают телеграмм-боты.

Классические чат-боты. Программы, которые ведут диалог с пользователем. Виртуальный собеседник обычно реагирует на заранее прописанные фразы, темы и команды. Так можно, к примеру, оптимизировать штат людей-операторов. Другие чат-боты пытаются имитировать человеческую беседу, но пользы от них не так много.

Помощники в администрировании. Пригодятся владельцам каналов и групповых чатов. Умеют собирать статистику, добавлять к постам реакции, URL-кнопки, файлы, создавать отложенные публикации, как [ControllerBot](#). Или отслеживать активность в чате, выдавать предупреждения и настраивать антиспам-фильтры — это умеет [Combot](#).

Магазины. Компании переносят свои каталоги в мессенджер — для тех, кому проще выбрать товар и оплатить его прямо в «Телеграме». Например, [через Baggins](#) можно купить чашку кофе, а [через G Shop](#) — женскую одежду. Главный минус таких магазинов — запутанный интерфейс: сложно просмотреть все товары и собрать корзину.

Инструменты. Разнообразные боты, которые добавляют мессенджеру новые функции. Они позволяют читать электронные письма, как [Gmail Bot](#), отслеживать цены на билеты, как [AviasalesBot](#), или проверить свой заказ [в боте СДЭКа](#). Еще они [напоминают о штрафах, скачивают файлы по URL и распознают музыку по аудиосообщению](#).
[47 полезных и необычных ботов в Telegram](#)

Поисковики. В них можно бесплатно найти книги, музыку, фильмы и прочие файлы. Зачастую эти боты пиратские — и «Телеграм» их периодически блокирует.

Развлечения. Все, что поможет скрасить досуг: [мемы](#), [частушки](#), игры и не только.

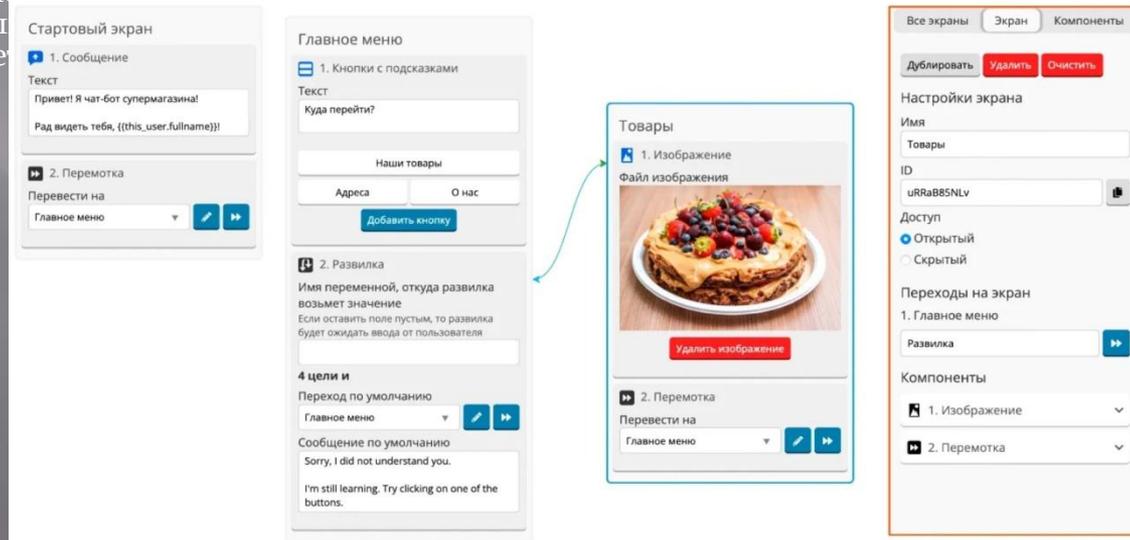
Какие навыки нужны, чтобы написать своего бота.

Есть два способа создать бота в «Телеграме»: с помощью кода и ноукод-платформ. В первом случае вы пишете бота самостоятельно, во втором — пользуетесь конструктором и собираете бота из уже готовых элементов.

Выбор зависит от того, насколько продвинутый бот вам нужен, как много у вас времени и ресурсов, владеете ли вы навыками программирования. Ноукод-платформы серьезно облегчают процесс, но ограничивают в возможностях. Такой способ не подойдет сложному приложению, особенно если оно должно решать нестандартные задачи.

Создать бота с помощью ноукод-платформы. Для создания ботов в конструкторе достаточно базовых знаний веб-приложений. Есть десятки платформ, в которых можно разработать ботов без знания кода. Если вы владеете Figma, Miro, Asana или перетаскивали карточки в Trello, то легко сможете составить бота из наглядной цепочки блоков в конструкторе. Отличаются они функциями, набором сценариев и стоимостью. Некоторые доступны прямо в «Телеграме» — к таким относится Manybot. Фактически это бот для создания ботов, но он сильно ограничен в возможностях. Веб-платформы же берут разнообразием: если вы не нашли подходящей функции на одном сайте, то она вполне может оказаться на другом.

Обычно в бесплатной версии ноукод-создатели позволяют создать одного бота. Но будет ограничение либо по количеству доступных экранов, либо по количеству компонентов. Если вы хотите создать классного бота не выйдете



Практическая часть.

Бота можно написать на любом языке программирования — от PHP и Node.js до Java и Go. Самым популярным считается Python, потому что в его экосистеме много открытых библиотек и готовых решений. К тому же он относительно легок в изучении. Особенно подойдет новичкам: достаточно понять логику синтаксиса и отступов. Начать изучение языка можно самостоятельно — например, [с книги Марка Вудса «Изучаем Python»](#), посмотреть [бесплатные курсы на «Ютубе»](#) или подобрать [один из курсов онлайн-школ](#).

Далее нужно установить [пакет python-telegram-bot](#) — это обертка для API от «Телеграма». В документации этого фреймворка описана вся необходимая информация для разработки бота. Также понадобится любой текстовый редактор для написания кода.

Для создания телеграмм бота необходимы базовые знания языка программирования Python. Первым делом, необходимо установить сам Python на ПК. Делается это через официальный сайт Python.

Далее необходимо зайти в командную строку. Необходимо иметь интернет подключение для корректной работы. Вводим: `python3 -m pip install pyTelegramBotAPI`. Это действие приведет к загрузке библиотеки telebot, с помощью которой и пишется код бота.

После завершения открываем IDLE(специальный встроенный IDE для написания программного бота на Python), создаем файл, обязательно чтобы формат файла был py(например, `my_first_bot.py`), и прописываем библиотеки: `random`(позволяет генерировать случайности), `telebot`(для создания простых ботов, позволяющая работать с асинхронным и синхронным подходом на выбор), `requests`(для составления

Дальше прописываем класс, чтобы было меньше строчек кода:

```
1 class Dost:
2     def __init__(self, title, text, sticker, link_one,
3         link_two, w, h, svbs ):
4         self.title = title
5         self.text = text
6         self.stike = sticker
7         self.link1 = link_one
8         self.link2 = link_two
9         self.w = w
10        self.h = h
11        self.class2 = svbs
12
13    def star(self, msg):
14        text = self.text
15        tittle = self.title
16        o = emojis.encode(":sparkles:")
17        bot.send_message(msg.from_user.id, o +tittle + o, parse_mode='Markdown')
18        bot.send_message(msg.from_user.id, emojis.encode(self.stike) + text)
19        bot.send_location(msg.from_user.id, self.w, self.h)
20        keyboard = telebot.types.InlineKeyboardMarkup(row_width=1)
21        but_url1 = telebot.types.InlineKeyboardButton("перейти на сайт" +
22            emojis.encode(":point_right:"), url=self.link2)
23        keyboard.add(but_url1)
24        bot.send_message(msg.from_user.id, "Больше информации" +
25            emojis.encode(":point_down:"), reply_markup=keyboard)
26
27    def start(self, msg):
28        response1 = requests.get(self.link1)
29        a1 = response1.content
30        html1 = BeautifulSoup(a1, "lxml")
31        b1 = html1.find(class_=self.class2)
32        z1 = b1.text
33        bot.send_message(msg.from_user.id, emojis.encode(":zap:") + "Отзывы" +
34            emojis.encode(":zap:") + z1)
35
36    def photos(self, msg, url_1):
37        bot.send_photo(msg.from_user.id, url_1)
38
```

Создаем клавиатуру, на которой будут кнопки самих районов:

```
1 @bot.message_handler(commands=["start"])
2 def start(msg):
3     p = emojis.encode(":four_leaf_clover:")
4     x = emojis.encode(":sunny:")
5     welcome = "Выберете или напишите район Белгородской области"
6     keyboard = telebot.types.ReplyKeyboardMarkup(resize_keyboard=True,
7         one_time_keyboard=False, row_width=3)
8     but_1 = telebot.types.KeyboardButton(p + "Алексеевский" + p)
9     but_2 = telebot.types.KeyboardButton(p + "Белгородский" + p)
10    but_3 = telebot.types.KeyboardButton(p + "Борисовский" + p)
11    but_4 = telebot.types.KeyboardButton(p + "Валуйский" + p)
12    but_5 = telebot.types.KeyboardButton(p + "Вейделевский" + p)
13    but_6 = telebot.types.KeyboardButton(p + "Волоконовский" + p)
14    but_7 = telebot.types.KeyboardButton(p + "Грайворонский" + p)
15    but_8 = telebot.types.KeyboardButton(p + "Губкинский" + p)
16    but_9 = telebot.types.KeyboardButton(p + "Ивнянский" + p)
17    but_11 = telebot.types.KeyboardButton(p + "Красненский" + p)
18    but_12 = telebot.types.KeyboardButton(p + "Красногвардейский" + p)
19    but_14 = telebot.types.KeyboardButton(p + "Новооскольский" + p)
20    but_15 = telebot.types.KeyboardButton(p + "Прохоровский" + p)
21    but_16 = telebot.types.KeyboardButton(p + "Ракитянский" + p)
22    but_18 = telebot.types.KeyboardButton(p + "Старооскольский" + p)
23    but_19 = telebot.types.KeyboardButton(p + "Чернянский" + p)
24    but_20 = telebot.types.KeyboardButton(p + "Шебекинский" + p)
25    but_21 = telebot.types.KeyboardButton(p + "Яковлевский" + p)
26    but_22 = telebot.types.KeyboardButton(x + "Мероприятия" + x)
27
28
29
30    keyboard.add( but_22, but_1, but_2, but_3, but_4, but_5, but_6, but_7,
31                but_8, but_9, but_11, but_12, but_14, but_15,
32                but_16, but_18, but_19, but_20, but_21)
33    bot.send_message(msg.from_user.id, welcome, reply_markup=keyboard)
```

Дальше прописываем каждый район, его достопримечательность, координаты достопримечательности, отзывы с сайта и фото.

```
1 @bot.message_handler(commands=["Алексеевский"])
2 def alek(msg):
3     list1 = [1, 0]
4     q = random.choice(list1)
5     if q == 1:
6         aaa = Dost("Историко-литературный музей Н.В. Станкевича*",
7             "Музей открыт в 1990 году, экспозиция посвящена "
8             "местному уроженцу - поэту, философу-просветителю, "
9             "организатору "
10            "«кружка Станкевича» Н.В. Станкевичу ", ":ticket:",
11            "https://autotravel.ru/otklik.php/27002",
12            "https://alemuz.ru/istoriko-literaturnyj-muzej-nikolaya/",
13            "50.676065", "38.867389",
14            "o_list")
15        aaa.photos(msg,
16            "https://cdn.culture.ru/images/8d80c759-42c3-5107-9f85-1a4dd1ab4fbc" )
17        aaa.start(msg)
18        aaa.start(msg)
19    else:
20        ss = Dost("Храм Святого благоверного князя Александра Невского*",
21            "Построен в 1888 году в новорусском стиле."
22            " Работа храма не прекращалась, действующий храм. "
23            "Памятник архитектуры регионального значения.", ":church:",
24            "https://autotravel.ru/otklik.php/27003",
25            "https://azbyka.ru/palomnik/Храм_Александра_Невского_(Алексеевка)",
26            "50.641555", "38.693938", "o_list")
27        ss.photos(msg,
28            "http://www.palomnik31.ru/images/hram/alekseevka/khram-svyatogo-bлаго"
29            "vernogo-knyazya-aleksandra-nevskogo.jpg")
30        ss.start(msg)
31        ss.start(msg)
```

Теперь прописываем обрабатывающий текст:

```
1 @bot.message_handler(content_types=["text"])
2 def user_start(msg):
3     p = emojis.encode(":four_leaf_clover:")
4     x = emojis.encode(":sunny:")
5     if msg.text == p + "Краснояржский" + p or msg.text == "Краснояржский":
6         red_yar(msg)
7     elif msg.text == p + "Грайворонский" + p or msg.text == "Грайворонский":
8         grai(msg)
9     elif msg.text == p + "Борисовский" + p or msg.text == "Борисовский":
10        bor(msg)
11    elif msg.text == p + "Ракитянский" + p or msg.text == "Ракитянский":
12        rack(msg)
13    ...
14    elif msg.text == p + "Ровеньский" + p or msg.text == "Ровеньский":
15        rove(msg)
16    elif msg.text == p + "Старооскольский" + p or msg.text == "Старооскольский":
17        staos(msg)
18    elif msg.text == p + "Чернянский" + p or msg.text == "Чернянский":
19        cher(msg)
20    elif msg.text == p + "Шебекинский" + p or msg.text == "Шебекинский":
21        sheb(msg)
22    elif msg.text == p + "Яковлевский" + p or msg.text == "Яковлевский":
23        ikov(msg)
24    elif msg.text == x + "Мероприятия" + x or msg.text == "Мероприятия":
25        nevs(msg)
26    else:
27        bot.send_message(msg.from_user.id, emojis.encode(":sweat:") +
28            "Неправильная команда" + emojis.encode(":sweat:"))
29
```

И в самом конце прописываем функцию, после вызова TeleBot начинает опрашивать серверы Telegram на предмет новых сообщений:

```
bot.polling()
```

Вот и все, Телеграмм-бот готов, и в нем вышло 653 строчки кода.

Результаты реализации проекта.

Первоочередной задачей в реализации нашей идеи было проведение социологического опроса учащихся школы на предмет личного отношения к благоустройству территории вокруг школы. Были опрошены учащиеся 10 класса (всего 22 человека). При анализе анкет выяснилось, что 94% опрошенных считают, что такой бот нужен каждому, и что они будут им пользоваться. И только 6% опрошенных показали, что им не нравится идея телеграмм бота.

Важное значение мы придавали изучению информации по программированию на Python. Из всего многообразия вариантов видов ботов, мы становились на тех, которые по нашим мнениям легче всего в написании

Литература:

- ▣ Журнал «Colloquium-journal» #7(59).2020 / Technical science
- ▣ <https://journal.tinkoff.ru/guide/howto-telegram-bot/#one>