

ПРИМЕНЕНИЕ ТЕХНОЛОГИЙ ЧАТВОТ В ОБЕСПЕЧЕНИИ ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ

П.Ю. Филяк, А.Н. Дымов

Рассматривается обеспечение состояния защищенности в технологии чат-бот (ChatBot) приложений с двумя разными подходами реализации данных программ. Для раскрытия темы (ChatBot) и понимания её значимости в обеспечении информационной безопасности в статье рассмотрено применение чат-ботов с использованием программного обеспечения мессенджера «Telegram», позволяющего пользователям не только эффективно и массово обмениваться информацией, но и выполнять запросы пользователей. Также рассматривается возможность «связанного» функционала мессенджера «Telegram» и чат-бот приложения, несмотря на то, какой подход реализации был использован в построении самого бота. В статье представлены два варианта создания чат-ботов, первый из которых базируется на использовании конструктора ботов и соответствующей среды с интерфейсом, что существенно упрощает задачи для пользователя, особенно для тех, кто не владеет серьезным заданием и навыками программирования. Второй вариант – профессионализированный, основанный на написании скриптов на языках программирования высокого уровня.

Ключевые слова: ChatBot, Telegram, мессенджер, Aimylogic, Python, pyTelegramBotAPI, BotFather, API, токен, конструктор ботов, программирование, скрипты, коды, компиляция, настройка, отладка.

Для начала дадим понятие «чат-бот». Чат-бот (или чат-бот приложение) – это программа, которая способна обрабатывать запросы пользователя-собеседника. В основе чат-бот приложений лежат специальные алгоритмы, симулирующие поведения человека, и они же обеспечивают работоспособность чат-бота.

Существует два подхода к написанию (созданию) чат-бот приложения: первый – это проектирование поведения чат-бота через специальные сервисы-конструкторы, а второй – это программирование, т. е. через

скрипты. Создадим тестового чат-бота, для этого воспользуемся информацией из теоретической базы [1].

1. Посредством конструктора чат-ботов.

Итак, будем использовать сервис для создания чат-ботов – конструктор под названием Aimylogic и мессенджер «Telegram» для общения с ботом. Для начала необходимо пройти регистрацию на официальном сайте сервиса «Aimylogic»: «<https://aimylogic.com/ru/>» (рис. 1).

Регистрация аккаунта Just AI

ЗАРЕГИСТРИРОВАТЬСЯ — ПОЛУЧИТЬ ПОДАРОК — НАЧАТЬ РАБОТАТЬ

Ваше имя и фамилия

Email
dymovan@ssuedu.ru

Пароль
.....

Пароль должен быть не менее 8 символов

Я согласен(на) с [условиями использования](#)

ПРОДОЛЖИТЬ

Рис. 1. Регистрация на сервисе «Aimylogic»

Затем выбираем тарифный «Developer», конечно, функционал данного тарифного плана сильно ограничен и не подходит для коммерческих проектов, но так как мы создаём лишь тестового чат-бота этого тарифного плана будет достаточно, к тому же он является бесплатным (рис. 2).

Теперь необходимо создать «пустого» чат-бота через Telegram. Для этого проходим в Телеграм-канал под названием «BotFather». Здесь мы сможем создать чат-бота с

помощью простой команды: «/newbot» (рис.3).

Теперь у нас есть «пустой» чат-бот, его необходимо «наполнить» диалогами и командами. Для этого необходимо подключить чат-бота к сервису «Aimylogic», запрашиваем в Telegram-канале «BotFather» API токен только что созданного бота (рис. 4). Теперь полученный API токен чат-бота представляем нашему конструктору «Aimylogic» (рис. 5).

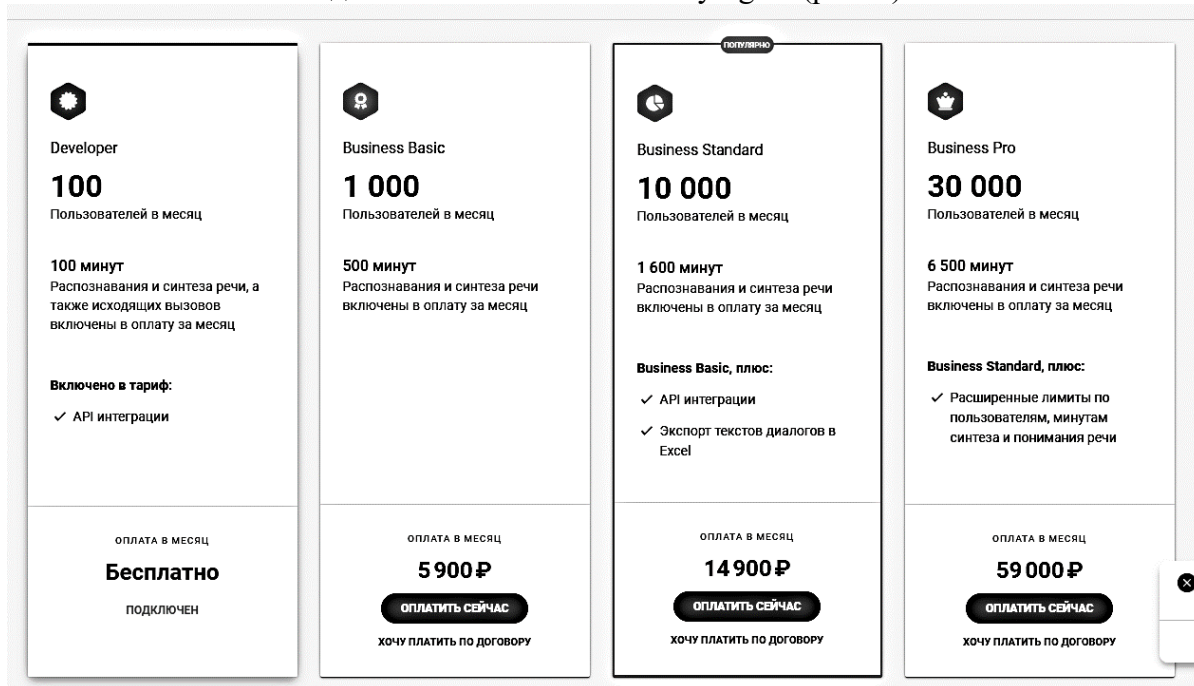


Рис. 2. Тарифные планы сервиса «Aimylogic»

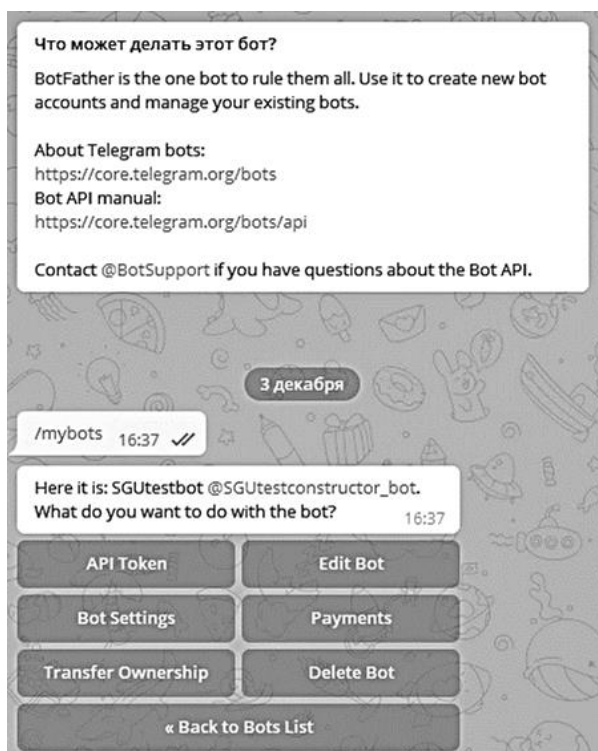


Рис. 3. Telegram-канал «BotFather»

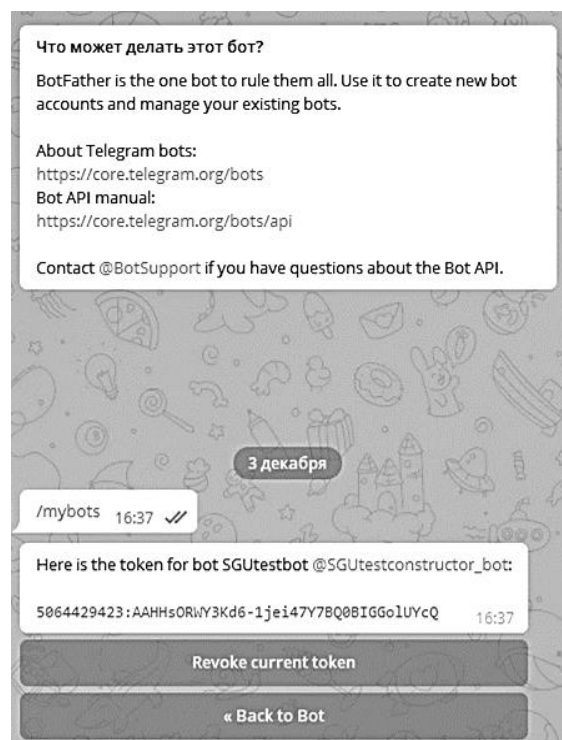


Рис. 4. Запрос API токена

На данный момент существует полноценный чат-бот для диалога с пользователями, но он не «наполнен» диалогами. Для того, чтобы у чат-бота появились диалоги необходимо в конструкторе перейти во вкладку «Сценарии» и создать необходимые сценарии поведения, данного чат-бот приложения.

Первым создадим сценарий поведения «Приветствия», который будет использоваться при запуске чат-бота (рис. 6). Далее создадим сценарий возможностей чат-бота. Здесь чат-бот будет ожидать ввода сообщений от пользователя и выполнять заложенные команды (рис. 7).



Рис. 5. Передача API токена

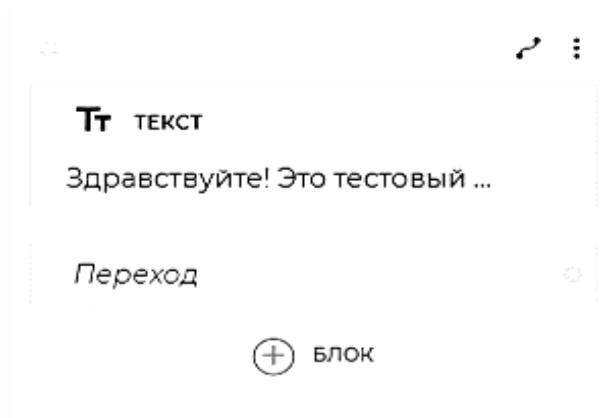


Рис. 6. Создание сценария приветствия

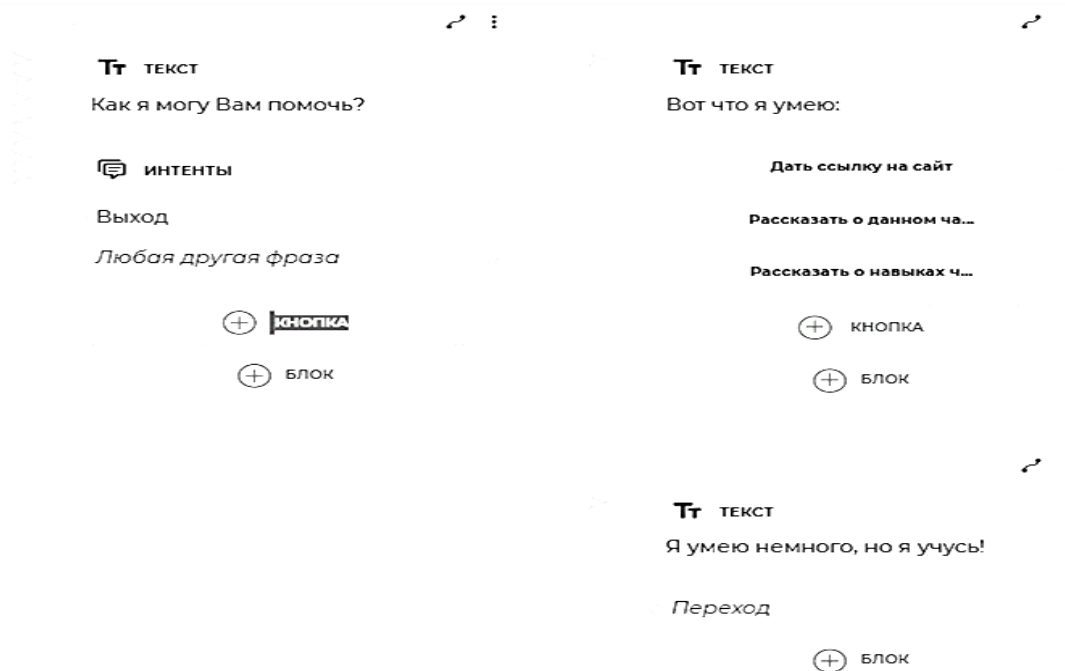


Рис. 7. Создание сценария возможностей

Функционал данного чат-бота не так широк, как у других чат-бот приложений. Но в рамках данного исследования этого достаточно, чтобы увидеть, как создаётся чат-бот через специальный конструктор. К тому же для создания чат-ботов с большим

функционалом в различных коммерческих проектах привлекаются целые команды специалистов. Для того, чтобы увидеть, как функционирует наш чат-бот приложение, необходимо провести с ним диалог (рис. 8, рис. 9).

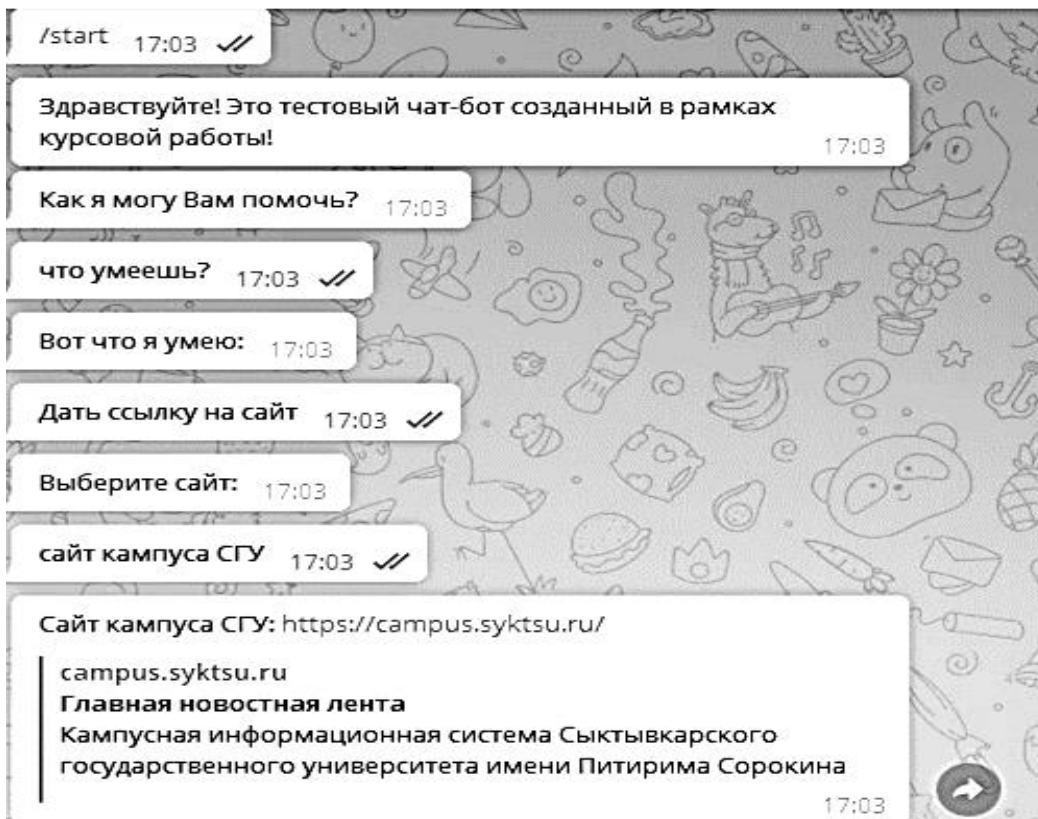


Рис. 8. Диалог с чат-ботом

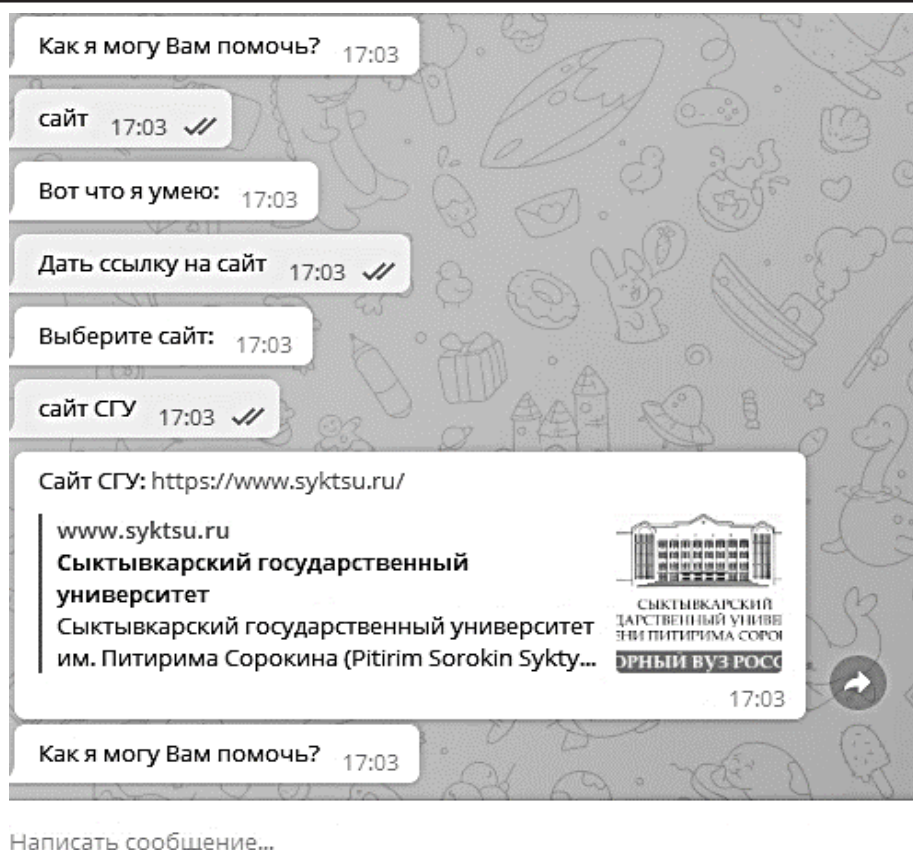


Рис. 9. Диалог с чат-ботом

Необходимо заметить, что сервис, который мы использовали для создания чат-бота, хранит API. Это достаточно большая проблема, так как сервис становится объектом с повышенным уровнем угрозы (как внешней, так и внутренней), так как лица, которые хотят (или хотели бы) получить полный контроль над ботом могут атаковать сервис.

2. С помощью программирования (скриптов)

Создав чат-бота с помощью первого способа, мы увидели, что это достаточно легко. Теперь воспользуемся вторым из способов, чтобы создать другого чат-бота. Для создания чат-бота воспользуемся языком программирования «Python», а точнее интегрированной средой разработки

«PyCharm» от компании JetBrains, и мессенджером «Telegram» для общения с ботом. Вся работа чат-бота строится на программном коде, но общение с пользователем происходит через мессенджер «Telegram». «Связь» между мессенджером и файлами - программами обеспечивается благодаря специальной библиотеке «pyTelegramBotAPI» для языка программирования «Python» [1-3]. Для начала необходимо создать «пустого» чат-бота в Телеграм-канале под названием «BotFather» (Рис. 10) и запросить API токен для только что созданного чат-бота (Рис. 11). Далее мы создаём проект «pythonbot» в среде разработки «PyCharm», после в этом же проекте создаём дополнительный файл «config.py» для сохранения в нём API токена (рис.10).

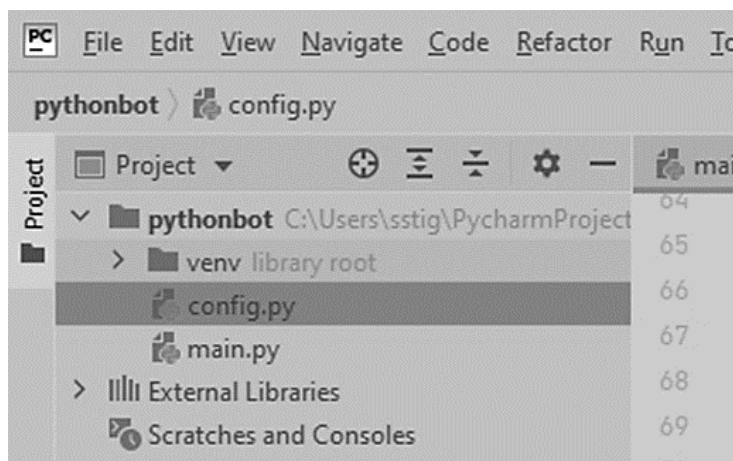


Рис. 10. Проект «pythonbot» в среде разработки «PyCharm»

Затем, в файле «main.py», пишется программный код для самого чат-бота. Скачиваем и импортируем «pyTelegramBotAPI» (рис. 11)

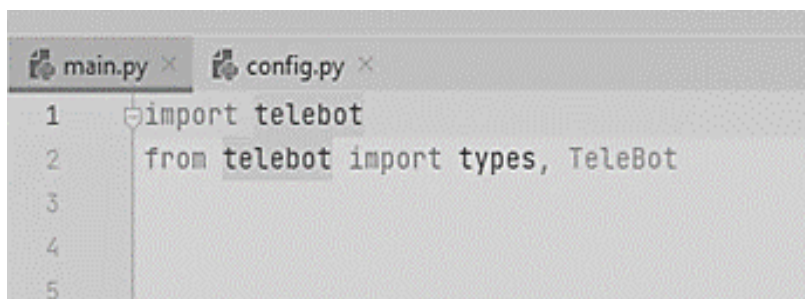


Рис. 11. Импорт библиотеки «pyTelegramBotAPI»

Перед основной функцией «def hello()» прописываем команду «@bot.message_handler(commands=['start'])» [3]. Чтобы программа ожидала ввода команды «/start» от пользователя и могла начать работу после её ввода (рис. 12).

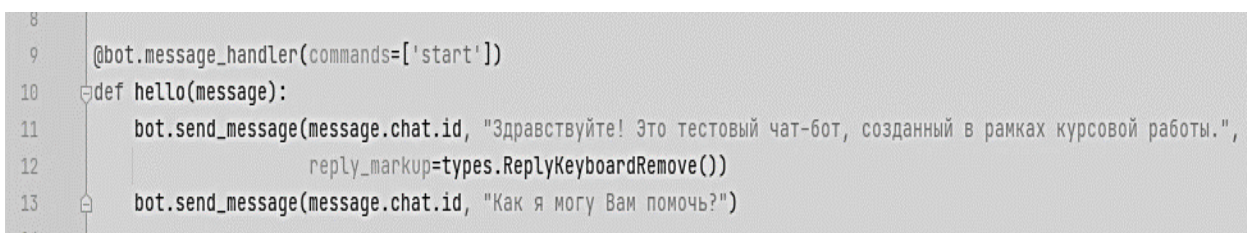


Рис. 12. Программный код чат-бота

Затем перед функцией «def first()» создаются интерактивные кнопки, которые будут всплывать в мессенджере, чтобы пользователь мог выбрать необходимое действие из списка (рис. 13). И также в самой функции прописываем команду «@bot.message_handler(content_types=['text'])», чтобы программа ожидала ввода текста от пользователя.

```
@bot.message_handler(content_types=['text'])
def first(message):
    if message.text != 'stopbot':
        markup = types.ReplyKeyboardMarkup(resize_keyboard=True)
        markup.add(types.KeyboardButton("Посчитать!"), types.KeyboardButton("Дать ссылку на сайт!"),
                  types.KeyboardButton("Очистить диалог!"))
        msg = bot.send_message(message.chat.id, "Я умею:", reply_markup=markup)
        bot.register_next_step_handler(msg, wait)
```

Рис. 13. Программный код чат-бота

Первая функция чат-бота – математические действия над двумя числами. Для выполнения данного действия создана «ожидающая» функция «def wait()» (рис. 14), в которую пользователь будет попадать для того, чтобы выбрать действие. После выбора пользователем функции «калькулятора» программа переходит в следующую функцию «def calculator1()», которая ожидает ввода первого числа, а затем – в «def calculator2()», которая ожидает ввода второго числа (рис. 15). После ввода чисел программа перенаправляет пользователя в функцию «def calculator()», которая создаёт интерактивные кнопки, которые будут всплывать в Telegram-чате нашего бота с пользователем. Также в нашем случае данные кнопки дают возможность пользователю выполнить определённое математическое действие.

Вторая функция чат-бота – выдача ссылки на сайт. Для этого из основной функции программа перенаправляет пользователя в функцию «def site()». Данная функция создаёт интерактивные кнопки и ожидает текст, чтобы дать ссылку на конкретный сайт.

Третья функция чат-бота не имеет практического применения. Она лишь даёт ответное сообщение пользователю. Для этого в самой функции «def wait()» программа ожидает ввода конкретного текста, если текст совпадает, то программа выводит ответное сообщение (рис. 16).

Пример работы чат-бота «@sgu_testpython_bot» в Telegram-канале (рис. 17)

```
def wait(message):
    if message.text == "Посчитать!":
        msg = bot.send_message(message.chat.id, "Посчитано!", reply_markup=types.ReplyKeyboardRemove())
        bot.send_message(message.chat.id, "Введите первое число:")
        bot.register_next_step_handler(msg, calculator1)
    elif message.text == "Дать ссылку на сайт!":
        msg = bot.send_message(message.chat.id, "Хорошо!", reply_markup=types.ReplyKeyboardRemove())
        markup = types.ReplyKeyboardMarkup(resize_keyboard=True)
        markup.add(types.KeyboardButton("Сайт СГУ"), types.KeyboardButton("Сайт Кампуса СГУ"),
                  types.KeyboardButton("Сайт Библиотеки СГУ"))
        msg = bot.send_message(message.chat.id, "Выберите сайт:", reply_markup=markup)
        bot.register_next_step_handler(msg, site)
    elif message.text == "Очистить диалог!":
        msg = bot.send_message(message.chat.id, "Прошу прощения, но я так не умею!",
                              reply_markup=types.ReplyKeyboardRemove())
        bot.register_next_step_handler(msg, hello)
```

Рис. 14. Программный код чат-бота

```

def calculator1(message):
    global x
    try:
        x = int(message.text) - 0
    except ValueError:
        msg = bot.send_message(message.chat.id, "Введите число!")
        bot.register_next_step_handler(msg, calculator1)
    else:
        msg = bot.send_message(message.chat.id, "Хорошо! Введите второе число:")
        bot.register_next_step_handler(msg, calculator2)

def calculator2(message):
    global y
    try:
        y = int(message.text) - 0
    except ValueError:
        msg = bot.send_message(message.chat.id, "Введите число!")
        bot.register_next_step_handler(msg, calculator2)
    else:
        markup = types.ReplyKeyboardMarkup(resize_keyboard=True)
        markup.add(types.KeyboardButton("Сложение(+)"), types.KeyboardButton("Вычитание(-)"),
                    types.KeyboardButton("Деление(÷)", types.KeyboardButton("Умножение(*)"))
        msg = bot.send_message(message.chat.id, "Выберите математическое действие:", reply_markup=markup)
        bot.register_next_step_handler(msg, calculator)
    
```

Рис. 15. Программный код чат-бота

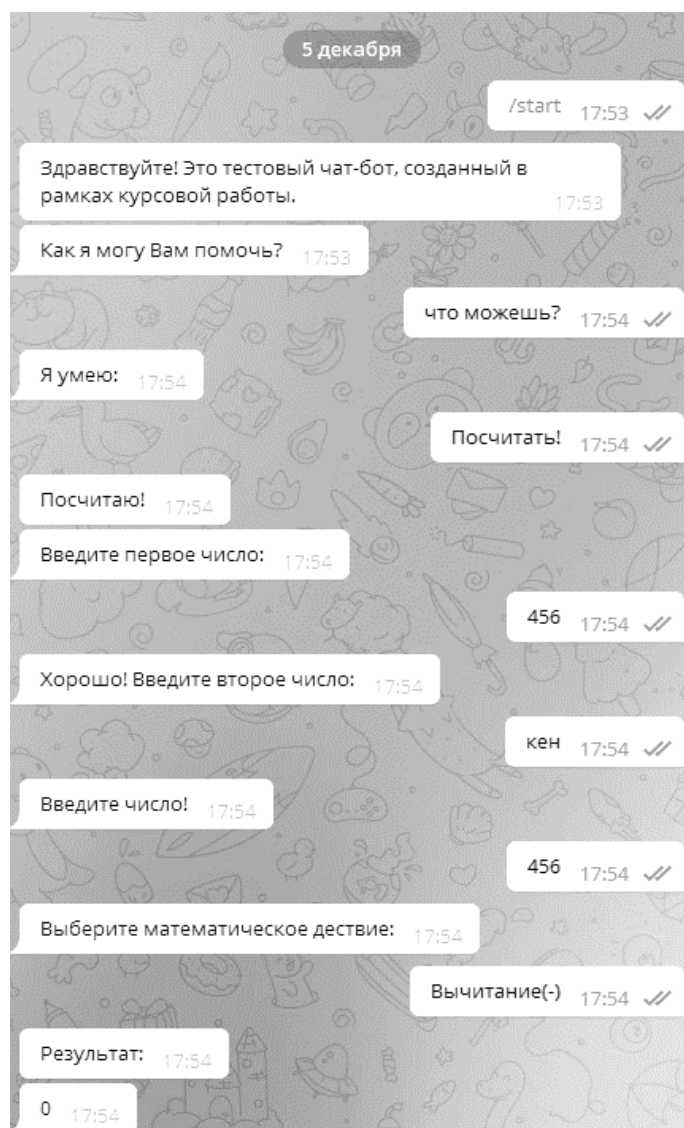


Рис. 16. Пример работы чат-бота

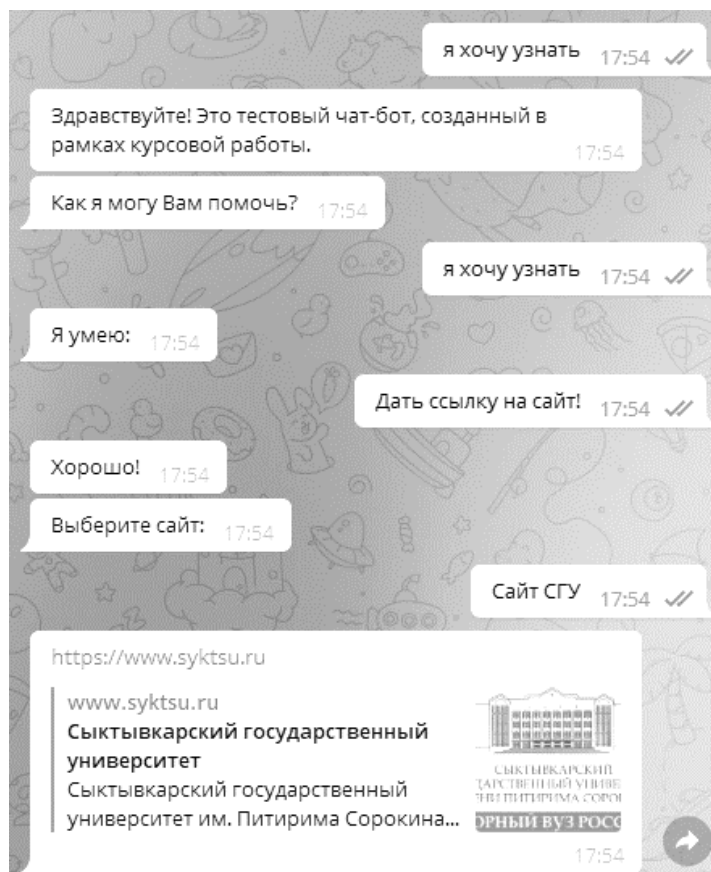


Рис. 17. Пример работы чат-бота

Также заметим, что на данный момент этот способ более безопасный, потому что потенциальные злоумышленники, не знают, где хранится API. Но после получения такой информации, злоумышленники могут подвергнуть угрозам то оборудование, на котором хранится файл с API чат-бота.

Список литературы

1. Слепцова Ю.Н. Автоматизация маркетинговых процессов при помощи чат-

бота // Научный журнал. 2020. №3 (48). URL: <https://cyberleninka.ru/article/n/avtomatizatsiya-marketingovyh-protsessov-pri-pomoschi-chat-bota> (дата обращения: (27.07.2022).

2. 13 онлайн-сервисов для создания чат-ботов. URL: <https://habr.com/ru/company/click/blog/567446/> (дата обращения: (27.07.2022).

3. Telegram Bot API. URL: <https://core.telegram.org/bots/api> (дата обращения: (27.07.2022).

Московский государственный лингвистический университет
Moscow State Linguistic University

Сыктывкарский государственный университет им. Питирима Сорокина
Syktyvkar State University after the Pitirim Sorokin (SSU)

Поступила в редакцию 04.08.2022

Информация об авторах

Филяк Петр Юрьевич – канд. техн. наук., директор Института информационных наук Московского государственного лингвистического университета, e-mail: p,filiak@linguanet.ru

Дымов Антон Николаевич – студент бакалавриата направления подготовки «Информационная безопасность», Сыктывкарский государственный университет, e-mail: dymovantonRUS@yandex.ru

APPLICATION OF CHATBOT TECHNOLOGIES IN ENSURING INFORMATION SECURITY

P.Yu. Filyak, A.N. Dymov

The article considers ensuring the state of security in the chatbot technology of applications with two different approaches to the implementation of these programs. To reveal the topic (ChatBot) and understand its importance in ensuring information security, the article discusses the use of chatbots using software, the Telegram messenger, which allows users not only to exchange information efficiently and massively, but also to fulfill user requests.. Also to understand the organization of the software (ChatBot), which allows you to process and execute user requests autonomously. It should be noted that the possibility of a "connected" functionality of the Telegram messenger and a chatbot application is also being considered, despite the implementation approach used in the construction of the bot itself. The article presents two options for creating chatbots, the first of which is based on the use of a bot constructor and an appropriate environment with an interface, which greatly simplifies the tasks for the user, especially for those who do not have a serious foundation and programming skills. The second option is professionalized – by writing scripts in a high-level programming language.

Keywords: ChatBot, Telegram, messenger, Aimylogic, Python, pyTelegramBotAPI, BotFather, API, token, bot constructor, programming, scripts, codes, compilation, configuration, debugging.

Submitted 04.08.2022

Information about the authors

Petr Yu. Filyak – Cand. Sc (Technical), Director of the Institute information Sciences of Moscow State Linguistic University, Associate Professor, e-mail: p.filiak@linguanet.ru

Anton N. Dymov – Bachelor of Information Security, Syktyvkar State University, e-mail: dymovantonRUS@yandex.ru