

**ЧАСТЬ 2**

**ХІІІ МЕЖДУНАРОДНАЯ НАУЧНО-  
ПРАКТИЧЕСКАЯ КОНФЕРЕНЦИЯ  
СТУДЕНТОВ И ШКОЛЬНИКОВ  
«МОЛОДЕЖЬ И ИННОВАТИКА»**

г.о.Серпухов  
**2020г.**

@Коллектив авторов

<b>131.</b>	<i>Автор: Билера Андрей Александрович, обучающийся 4 курса ВА РВСН имени Петра Великого (филиал в г. Серпухове) Научный руководитель: к.т.н., доцент Руденко Эдуард Михайлович.</i>	<i>ИССЛЕДОВАНИЕ МАРШРУТОВ ГРУППЫ БЕСПИЛОТНЫХ ЛЕТАТЕЛЬНЫХ АППАРАТОВ</i>	<b>465</b>
<b>132.</b>	<i>Автор: Воробьев Евгений Павлович, курсант филиала военной академии РВСН имени Петра Великого г. Серпухова Московской области Научный руководитель: Семикина Елена Викторовна, преподаватель информатики</i>	<i>ПРИМЕНЕНИЕ ГЕНЕТИЧЕСКОГО АЛГОРИТМА ДЛЯ РАСЧЕТА ОПТИМАЛЬНЫХ МАРШРУТОВ НА ПОЛУОРИЕНТИРОВАННОМ ГРАФЕ ДОРОГ В МНОГОКРИТЕРИАЛЬНОЙ ЗАДАЧЕ</i>	<b>468</b>
<b>133.</b>	<i>Автор: Есипов Вадим Игоревич, студент МАИ, г. Москва Научный руководитель: к.ф.-м.н., доцент Гордеенко Анатолий Михайлович, зав. кафедрой МАИ</i>	<i>ОСОБЕННОСТИ ОРГАНИЗАЦИИ И МОНИТОРИНГА РЕСУРСОСБЕРЕЖЕНИЯ В ТРАНСПОРТНЫХ ПОДРАЗДЕЛЕНИЯХ ОРГАНИЗАЦИЙ</i>	<b>472</b>
<b>134.</b>	<i>Автор: Журавлев Дмитрий Александрович, студент ГБПОУ МО «Серпуховский колледж» Научный руководитель: Черникова Лилия Валентиновна, преподаватель</i>	<i>ПРИМЕНЕНИЕ МЕТОДОВ МАТЕМАТИЧЕСКОГО ПРОГРАММИРОВАНИЯ КАК ОСНОВА ПОСТРОЕНИЯ СИСТЕМ ПОДДЕРЖКИ ПРИНЯТИЯ УПРАВЛЕНЧЕСКИХ РЕШЕНИЙ В МАЛОМ И СРЕДНЕМ ПРЕДПРИНИМАТЕЛЬСТВЕ</i>	<b>475</b>
<b>135.</b>	<i>Автор: Завгородний Никита Игоревич, студент 3-го курса государственного университета «Дубна», филиал «Протвино» Научный руководитель: Кульман Татьяна Николаевна, к.т.н., доцент кафедры Информационных технологий университета «Дубна», филиал «Протвино»</i>	<i>СОЗДАНИЕ ЧАТ БОТА ОРГАНАЙЗЕРА НА ПЛАТФОРМЕ .NET CORE</i>	<b>478</b>
<b>136.</b>	<i>Автор: Зябкин Максим Алексеевич, студент 3 курса филиала «Протвино» Государственного университета «Дубна» Научный руководитель: к.э.н. Захарова Лидия Ивановна, доцент кафедры Информационных технологий филиала «Протвино» Государственного университета «Дубна»</i>	<i>ИНФОРМАЦИОННЫЕ СИСТЕМЫ В ОБРАЗОВАНИИ: ОТЕЧЕСТВЕННАЯ ПРАКТИКА И ЗАРУБЕЖНЫЙ ОПЫТ</i>	<b>482</b>

3. Методы принятия управленческих решений: учебник для бакалавров / Л.А. Трофимова, В.В. Трофимов. – м.: Издательство Юрайт, 2013. – 335с. – Серия: Бакалавр. Базовый курс

20.15.05

## СОЗДАНИЕ ЧАТ БОТА ОРГАНИЗАТОРА НА ПЛАТФОРМЕ .NET CORE

**Автор:** Завгородний Никита Игоревич, студент 3-го курса государственного университета «Дубна», филиал «Протвино»

**Научный руководитель:** Кульман Татьяна Николаевна, к.т.н., доцент кафедры Информационных технологий университета «Дубна», филиал «Протвино»

### Аннотация

Рассматривается создание чат бота организатора. Описываются выбор сервиса, проектирование базы данных, непосредственно разработка бота. В настоящее время бот эксплуатируется и доступен в социальной сети «ВКонтакте».

### Annotation

The creation of a chat bot organizer is considered. Describes the choice of service, database design, bot development. Currently, the bot is operated and available on the «VKontakte» social network.

**Ключевые слова:** базы данных, БД, чат бот, SQL Server, T-SQL, ВКонтакте, ВК, MySQL, PostgreSQL, кроссплатформенность, .NET CORE, Long Poll API.

**Keywords:** data base, DB, chat bot, SQL Server, T-SQL, VKontakte, VK, MySQL, PostgreSQL, cross-platform, .NET CORE, Long Poll API.

Чат бот – это результат взаимодействия человека и искусственного интеллекта: робот автоматически отвечает на сообщения пользователя. Чат боты используются для множества целей: для выполнения повседневных задач, развлечений и многих других. Размещают их в мессенджерах и социальных сетях, – Facebook Messenger, Telegram, “ВКонтакте”, Slack и т.д.

**Целью работы** является создание чат бота организатора для социальной сети “ВКонтакте”.

Среднесуточная аудитория, по данным на август 2017 года, “ВКонтакте” составляет более 80 миллионов посетителей. По данным SimilarWeb в сентябре 2019 года, сайт “ВКонтакте” занимал 12 место по популярности в мире.

**Актуальность:** Тайм-менеджмент весьма сложная задача для многих современных и активных людей, умение управлять своим временем, зачастую, является секретом успешности. Стремительное развитие технологий и быстрота течения времени делают грамотное распределение времени серьёзной и актуальной проблемой.

**Постановка задачи:** создание чат бота организатора с функционалом:

- Создания обычных/ежедневных/еженедельных/ежемесячных/ежегодных событий через заданное время, в определённое время;
- Удаления событий по названию, либо по идентификатору;
- Поиска событий по времени/по названию;
- Уведомления о созданных событиях.

Платформа .NET Core была выбрана из-за своей кроссплатформенности,- бота без проблем можно запустить на Windows, Linux и macOS. Благодаря такому выбору платформы можно создать полностью независимую версию бота, либо предварительно установить соответствующую версию .NET Core и использовать портативную версию консольного приложения.

Для хранения данных бот использует базу данных. Изначально разработка велась для СУБД MySql (Тестирование проходило на MariaDB), в последствии была так же добавлена поддержка PostgreSQL.

“ВКонтакте” для работы с ботами предоставляет на выбор Callback API и Bots Long Poll API, был выбран второй из-за своей гибкости.

Для взаимодействия с VK API используется библиотека VkNet, в ней реализованы методы VK API для .NET. Для взаимодействия с базой данных используются библиотеки MySql.Data и Npgsql.EntityFrameworkCore.PostgreSQL. Распознавание голосовых сообщений происходит благодаря API wit.ai.

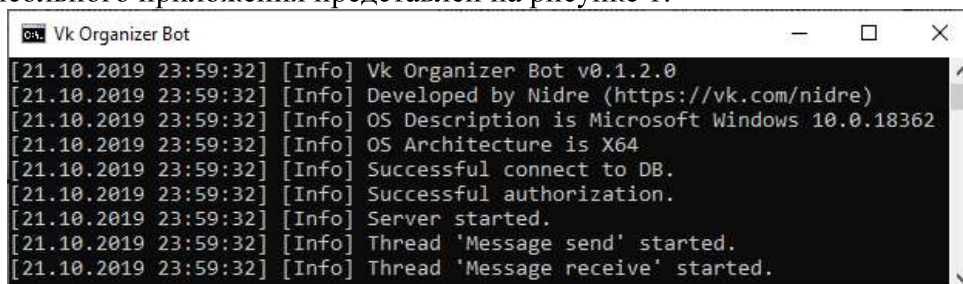
Перед началом написания шаблона, были продуманы основные шаблоны сообщений, которыми пользователь будет оперировать при взаимодействии с ботом, они следующие:

1. Напомни мне через/в `_time_` `_event_`
2. Через `_time_` напомни мне `_event_`
3. Когда `_event_`
4. Через сколько `_event_`
5. Во сколько `_event_`
6. Удалить/удали `_event_`
7. Привет/начать
8. Помощь

Благодаря продумыванию сообщений, была спроектирована примерная логика анализа сообщений и началась разработка.

Разработка началась с создания информативного и удобного для отладки консольного приложения, вся информация о функционировании бота выводится в консоль с временем и записывается в лог, для каждого сеанса создаётся свой лог файл.

Вид консольного приложения представлен на рисунке 1.



```
ca. Vk Organizer Bot
[21.10.2019 23:59:32] [Info] Vk Organizer Bot v0.1.2.0
[21.10.2019 23:59:32] [Info] Developed by Nidre (https://vk.com/nidre)
[21.10.2019 23:59:32] [Info] OS Description is Microsoft Windows 10.0.18362
[21.10.2019 23:59:32] [Info] OS Architecture is X64
[21.10.2019 23:59:32] [Info] Successful connect to DB.
[21.10.2019 23:59:32] [Info] Successful authorization.
[21.10.2019 23:59:32] [Info] Server started.
[21.10.2019 23:59:32] [Info] Thread 'Message send' started.
[21.10.2019 23:59:32] [Info] Thread 'Message receive' started.
```

Рисунок 1. Вид консольного приложения

Был создан файл конфигурации в формате JSON, с этим форматом удобно работать в программе, и он хорошо читабелен пользователем, в него выведены обязательные переменные, вроде ключей доступа API и необязательные, вроде включения/отключения отладочных сообщений. Благодаря этому файлу, бота можно развернуть на любой системе за пару минут.

Файл конфигурации представлен на рисунке 2.

Далее была спроектирована база данных. Диаграмма базы данных MySql представлена на рисунке 3. Для PostgreSQL она незначительно отличается.

Далее началась непосредственно разработка чат бота. Приложение использует один основной поток для ввода команд в консоль, для чат бота используются ещё два потока: для принятия сообщений и для уведомления о событиях. Так же в зависимости от параметра в конфигурации, при получении сообщения оно может отправиться для анализа в один из 16 доп. потоков или анализироваться в этом же.

Основной функцией приложения для анализа сообщения является Message.Analyze, логика анализа следующая:

1. Проверяем длину приложения.

- а. Если длина некорректная, сообщаем об этом пользователю.

```
{
  "DebugMode": true,
  "ServerStatus": 1,
  "AccessToken": "",
  "GroupId": 0,
  "WetAiKey": "",
  "WaitEventsIntervalSec": 25,
  "SendEventsIntervalMs": 1000,
  "MultiThreadingForReceive": true,
  "DBEngine": "MySQL",
  "DBHost": "localhost",
  "DBPort": "3306",
  "DBLogin": "root",
  "DBPass": "",
  "DBName": "vkbot"
}
```

Рисунок 2. Файл конфигурации



Рисунок 3. Диаграмма базы данных

2. Если длина корректная, то удаляем знаки препинания с начала и конца предложения (при анализе они будут только мешать), разделяем сообщения на слова и проверяем первое слово на наличие слов вроде “Когда” или “Удали”.
  - а. При нахождении “Удали” удаляются шаблонные слова и словосочетания вроде “Удали”, “событие”, всё что осталось это id события или название события, передаём строку в следующий класс для удаления.
  - б. При нахождении “Когда” удаляются шаблонные слова и словосочетания вроде “Когда”, “у меня”, всё что осталось это название события, передаём строку в следующий класс для поиска.
3. Если слов не найдено, то удаляем лишние слова, вроде “Напомни мне”, “Создай событие”, ищем корректное слово “Через”, корректное – значит после него идёт время или “Сколько”.
  - а. При нахождении “Сколько” удаляем его, остальное – строка события, отправляем в следующий класс для поиска.
  - б. Если после “Через” стоит время, вырезаем правильно строку времени, остальное – строка события, отправляем обе строки в следующий класс для создания события.
4. Если “Через” не нашлось, ищем “Во вторник” или “Во сколько”.
  - а. При нахождении “Во вторник” ищем время и правильно вырезаем строку времени, остальное – строка события, отправляем в следующий класс для создания события.
  - б. При нахождении “Во сколько” удаляем его, остальное – строка события, отправляем в следующий класс для поиска.
5. Далее ищем корректное “В”.
  - а. При нахождении корректного “В” правильно вырезаем строку времени, остальное – строка события, отправляем в следующий класс для создания события.

6. Если ни одного корректного из этих 3 слов не нашлось, проверяем готовые шаблоны предложений типа “Что ты умеешь”.
7. Если ничего не нашлось, сообщаем пользователю об ошибке.

“Следующий класс” должен преобразовать и проверить строки на корректность и передать их в класс взаимодействия с БД и ждать ответа, отправить этот ответ пользователю. Полную схему взаимодействия классов программы можно увидеть на рисунке 4.

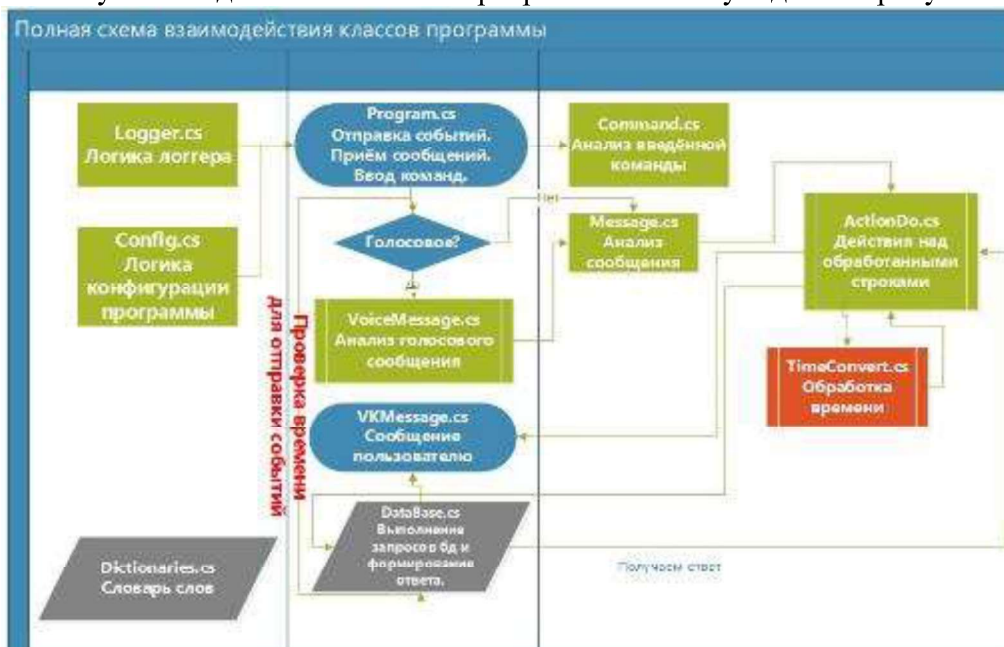


Рисунок 4. Схема взаимодействия классов программы

Бот был протестирован на Windows 10, Ubuntu 15.04, Ubuntu 18.04, были применены некоторые правки для корректной работы на разных ОС. Были привлечены несколько человек, каждый искал ошибки в работе, предлагал идеи. Все выявленные ошибки были исправлены. Некоторые предложения по функционалу были приняты и доработаны.

**Результатом** проделанной работы является:

2. В настоящее время бот создан, эксплуатируется, и доступен в группе по адресу <https://vk.com/vkorganizer>
3. Спроектирована база данных, структура была записана в приложение. Поддержка СУБД с синтаксисом MySQL, PostgreSQL.
4. Создание кроссплатформенного приложения портативного чат бота на платформе .NET Core с возможностью запуска на Windows, Linux и macOS.

Разработанный чат бот позволит использовать своё время эффективнее, грамотно составлять план своей деятельности, ставить цели и достигать своих целей в рамках предусмотренного времени. Чат бота можно использовать на любой платформе, поддерживаемой социальной сетью “ВКонтакте”, для доступа к которой достаточно доступа в интернет и любого браузера.

#### Использованные ресурсы:

1. Документация | Разработчикам -- VK API -- <https://vk.com/dev/manuals>
2. VKNET ВКонтакте API для .NET (C#) -- <https://vknet.github.io/vk>
3. NuGet Gallery | MySql.Data 8.0.18 -- <https://www.nuget.org/packages/MySql.Data/8.0.18/>
4. NuGet Gallery | Npgsql.EntityFrameworkCore.PostgreSQL 3.0.1 -- <https://www.nuget.org/packages/Npgsql.EntityFrameworkCore.PostgreSQL/3.0.1/>
5. Wit — HTTP API - <https://wit.ai/docs/http/20170307>