Муниципальное бюджетное общеобразовательное учреждение

"Лицей №174"

Учебно-исследовательский проект

**Школьный чат-бот**

Авторы: Аносов Алексей, 9 класс

Преснов Пётр, 9 класс

Юзофатов Владислав, 9 класс

Руководители: Снегирёва В.С. учитель информатики,

Отц А.С. старший scala разработчик

**ЗАТО г. Зеленогорск**

**2018**

# Введение

**Актуальность:**

В образовательных учреждениях нашей страны существует две проблемы связанные

с общением между школой и внешним миром.

1. Часто нужная информация в интернете лежит в местах, докуда тяжело добраться рядовому пользователю. Например, на вопрос: «Какие часы работы вашей школы» завучам, секретарям и учителям приходится отвечать по несколько раз на дню разным родителям или ученикам. Всё это из-за отсутствия открытой информации на такие вопросы.
2. При получении новых отметок, изменении расписания, появлении срочной новости о каких-либо олимпиадах или мероприятиях в электронном дневнике ученики и их родители не могут сразу узнать об этих обновлениях, так как это нигде ни оглашается, а тихо выставляется в электронный журнал.

Из-за этих проблем появилась необходимость в создании такого сервиса, который может докладывать информацию разным пользователям без привлечения в это дело работников школ и оповещать пользователей при каких-либо изменениях планов школы, получении оценок.

С ролью такого сервиса отлично справляются чат-боты, которые в настоящее время активно распространяются во все сферы современной жизни. Но для школ чат-боты это новое, неизведанное. Мы решили стать первопроходцами и реализовать такой сервис для нашей школы.

**Цели**: создать школьный чат-бот с привязкой к мессенджеру Вконтакте с регистрацией пользователей и разделения их на группы, обработкой запросов.

**Гипотеза**: при достаточном выделении времени на изучение работы с объектно-ориентированным языком программирования и базой данных можно создать школьный чат-бот

**Методы исследования**: теоретический (поиск литературы, абстрагирование, анализ и синтез специальной литературы, обобщение) и эмпирический (тестирование, эксперимент, сравнение).

**Предмет исследования**: алгоритм распознавания запроса, алгоритм регистрации пользователей чат-бота через мессенджер Telegram и подразделение их на группы, которым разрешена определённая область ресурсов.

**Объект исследования:** распознавание запроса, регистрации я пользователей.

**Конечный результат**: школьный чат-бот.

## 2.1. Выбор языка программирования

Нам необходимо было определиться на каком языке программирования удобнее и быстрее всего можно написать бота. Для этого мы провели тест, по результатам которого мы определились, что нам подходит язык Kotlin, который является усовершенствованным аналогом языка Java.

## 2.2. Выбор среды разработки

Для работы с языком Kotlin интернет предлагает две среды: Eclipseи IntellijIDEA. Необходимо было определиться, какую среду лучше использовать. По этому вопросу было найдено много статей, рассказывающих о преимуществах IntellijIDEA над Eclipse. Поэтому мы решили не отделяться от всех и тоже выбрали Intellij IDEA.

## 1.4. Выбор системы управления базой данных

Для реализации данного алгоритма нам было необходимо определиться с выбором системы управления базой данных для манипуляции над данными. Для этой цели мы выбрали базу данных H2, которая является простой и более производительной, чем многие другие СУБД\*. Стандартом языка для управления базой данных является SQL – язык структурированных запросов. SQL применяется для создания, модификации и управления данными в БД. СУБД (система управления базой данных) – совокупность базы данных и системы работы с ней.

**Привязка к Вконтакте**

Для начала нам необходимо было сделать так, чтобы наш бот мог отправлять сообщения в сам мессенджер. Для этого мы использовали предложенные самой платформой методы – это Long poll Api – который указывает к какому боту мы привязываем свой код и библиотека для привязки кода к самой платформе.

**Почему регистрация?**

Поскольку чат-бота мы делали для школы, в которой каждый пользователь - это учитель, ученик, родитель или администрация, то стала необходима функция регистрации и разделения пользователей по ролям для предоставления им прав на доступ к информации и для рассылки сообщений, уведомлений каждой группе лиц.

## Регистрация пользователей

1. Обычно в сервисах, использующих регистрацию, при самой регистрации и дальнейшем входе в учётную запись необходим логин и пароль. Но поскольку регистрация пользователя чат-бота происходит в чате, войти в который можно при условии, что есть учётная запись Вконтакте, то вход в учётную запись бота можно опустить и при регистрации связывать аккаунт Вконтакте с аккаунтом бота, чтобы не подтверждать свою личность дважды, а сделать это единожды.
2. Таким образом, мы решаем проблему школьного электронного журнала, связанную с тем, что вход в электронный журнал производится при каждой необходимости, однако теперь мы сможем заходить лишь один раз, и больше вход не потребуется.

## 1.2. Аутентификация пользователя

Последовательность действий для аутентификации ученика или его родителя при регистрации:

1. Проверка является ли пользователь причастным к данной школе

Для этого необходимо запросить информацию, доступную только этому пользователю, не меняющуюся со временем и хранящуюся в базе данных школы. Такой информацией является логин от электронного дневника и номер СНИЛСа

1. Проверка является ли пользователь тем, за кого он себя выдаёт

Она нужна для того, чтобы убедиться, что данные о пользователе не были украдены другим человеком, пытающимся выдать себя за другого. Для этой проверки нами было принято решение о необходимости личного подтверждения пользователя. Для этого будем отправлять информацию о пользователе, который хочет зарегистрироваться вышестоящим сотрудникам, которые подтверждают то, что человек, который на данный момент регистрируется в системе, именно тот, за которого себя выдаёт, путём звонка, либо личной беседы с пользователем.

## 1.3. Алгоритм регистрациипользователя

Составленный алгоритм регистрации пользователя можно посмотреть в приложении 1.

## 2.3. Структура хранения данных о пользователе в БД

Данные о пользователях хранятся в специальных таблицах в базе данных. Каждый пользователь обязательно занесён в таблицу пользователи (USERS) и одну из групповых таблиц (TEACHERS, STUDENTSилиPARENTS). При этом ученики и учителя привязаны к классам, а родители привязаны к ученикам

Так же есть команды, которые запрещены или разрешены каждому типу пользователей

## 1.5. Алгоритмы нечеткого поиска

При вводе какого-либо текста, а тем более при написании команды для обращения к боту, пользователи не редко ошибаются, что приводит к недопониманию со стороны бота, а это приводит к тому, что бот выдаёт неправильный ответ на запрос пользователя. Для того, чтобы исправить такой косяк, нами был придуман выход – алгоритм нечёткого поиска. С помощью его и функции поиска подходящего запроса из базы данных мы сможем выбрать тот самый запрос, на который боту следует ответить. (тут ещё какой-то текст про необходимость выбрать алгоритм нечёткого поиска).

## 1.6.1. Конечный автомат

Конечный автомат **-** это некоторая абстрактная модель, содержащая конечно**е** число состояний чего-либо. Существует множество различных способов задания алгоритма функционирования конечного автомата. Например, конечный автомат может быть задан в виде упорядоченной пятерки элементов некоторых множеств.

 M = (V,Q,q0,F,b), где

1. V – входной алфавит;
2. Q – множество внутренних состояний;
3. q0 – начальное состояние;
4. F – множество заключительных, или конечных состояний;
5. b – функция переходов.

Конечный автомат начинает работу в состоянии q0, последовательно считывая по одному символу. Считанный символ переводит автомат в следующее состояние.

## 1.6.2. Разновидность конечных автоматов

Детерминированный конечный автомат (ДКА)–это такой автомат, в котором нет дуг с меткой E предложение, не содержащие ни одного символа.

Недетерминированный конечный автомат практически ни чем не отличается от детерминированного, кроме существования нескольких переходов по одному символу из одного состояния.



Детерминированный конечный автомат:



- Недетерминированный конечный автомат.

## 1.6.3. Алгоритм Вагнера-Фишера O(min(n, m))

Алгоритм Вагнера-Фишера – это минимальное количество операций вставки символа, удалённого символа и замены символа на другой, необходимых для превращения одной строки в другую.

Применение алгоритма Вагнера-Фишера

1. Для исправления ошибок в слове;
2. Для сравнения текстовых файлов;
3. В биоинформатике для сравнения генов, хромосом и белков.

## 1.7. Редакционное предписание

Редакционное предписание – это последовательность действий, необходимых для получения из первой строки второй кратчайшим образом.

Обозначения действий

1. D – удалить;
2. I – вставить;
3. R – заменить;
4. M – совпадение;

Для нахождения кратчайшего расстояния необходимо использовать матрицу.

 Для наглядности можно воспроизвести матрицу в excel по формуле =MIN(B3+1;C2+1;B2+IF($A3=C$1;0;1))В нижнем правом углу будет минимальное расстояние Левенштейна. Обратите внимание, как можно оптимизировать алгоритм по памяти. Для работы нам можно хранить только текущую строчку и предыдущую.

## 1.8. Сравнение алгоритмов и выбор лучшего

Мы сравнили эти алгоритмы и пришли к такому выводу: для реализации такого алгоритма для чат-бота лучшим будет Алгоритм Вагнера-Фишера, так как он прост в реализации, затрачивает меньше ресурсов компьютера и не отстаёт от конечного автомата в скорости работы.

**Рассказ про реализацию**

Далее мы составили план работы, бюджет и риски проекта;

После этого реализовали все функции, про которые мы рассказали выше и готовы представить работу нашего бота.

## 2.10. Устойчивость

В нашем проекте могут быть заинтересованы образовательные учреждения, которые хотят иметь своего чат-бота. Этот проект никак не разовый. Его разработка и доработка может происходить постоянно.

## 2.11. Распространение результатов

В нашем проекте будут заинтересованы ребята которые занимаются программированием, и мы с радостью поделимся с ними своим опытом работы в языке программирования Kotlin.В дальнейшем наш проект будет развиваться и дальше.Летом мы собираемся расширить свой штат разработчиков, чтобы улучшать функционал чат-бота и создать школьный портал. Так же мы пользуем gradle для сборки проекта, поэтому к нам могут присоединиться ребята, которые пишут на Java.

# Заключение

На протяжении 4 месяцев мы занимались созданием чат-бота. За этот период мы:

1) Познакомились и изучили основные аспекты языка Kotlin

2) изучили язык структурированных запросов SQL

3) научились работать в среде разработки Intellij IDEA

4) научились работать с базой данных h2

5) разработали и реализовали собственный надёжный алгоритм регистрации и разделения по ролям для пользователей чат-бота.

6) Реализовали алгоритм нечёткого поиска

7) Собрали готовый продукт

Это означает, что мы выполнили задачи, достигли поставленной цели, подтвердили гипотезу, устранили проблемы.

Мы работали в группе разработчиков под наблюдением Отц А.С.- старший Scala разработчик. Мы предусмотрели риски проекта, которые минимизировали трудности и позволили сделать работу качественно. Мы предусмотрели риски исследования, что помогло быстрее и качественнее сделать работу. . Стоимость создания чат-бота составила 42 290руб., однако эти деньги окупают себя, так как проделанная нами работа поможет многим людям.

Полученные нами результаты могут быть полезны в области информатики для начинающих или опытных программистов, которые хотят писать программы на современном, быстром и удобном языке программирования, с которыми мы сможем поделиться опытом работы с SQL, Kotlin, h2 или IntellijIdea.

В перспективе, на основе уже изученных аспектов языка Kotlin и SQL, создать ещё больше функций для чат-бота, в том числе и привязать чат-бота к электронному журналу для рассылки уведомлений о мероприятиях и оценках.

#