

## Разработка текстового мессенджера с фильтром слов

*Скандаленко Александр Валерьевич*

*Приамурский государственный университет им. Шолом-Алейхема*

*студент*

*Лучанинов Дмитрий Васильевич*

*Приамурский государственный университет им. Шолом-Алейхема*

*Старший преподаватель кафедры информационных систем, математики и правовой информатики*

### Аннотация

В данной статье рассмотрена проблема отсутствия пользовательского фильтра сообщений в современных мессенджерах. В качестве исследования были изучены известные мессенджеры и способы их применения, а также необходимое программное обеспечение. В результате были разработаны программы-мессенджеры, использующие собственный настраиваемый цензурный фильтр.

**Ключевые слова:** мессенджер, Object Pascal, Delphi.

## Development of a text messenger with a word filter

*Skandalenko Aleksandr Valerievich*

*Sholom-Aleichem Priamursky State University*

*student*

*Luchaninov Dmitry Vasilyevich*

*Sholom-Aleichem Priamursky State University*

*Senior lecturer of the Department of information systems, mathematics and law informatics*

### Abstract

This article deals with the problem of the absence of a custom message filter in modern messengers. As a study, the well-known messengers and ways of their application as well as the necessary software were studied. As a result, messenger programs using their own custom censorship filter were developed.

**Keywords:** messenger, Object Pascal, Delphi.

В современных реалиях человеку необходимо общаться с другими людьми, которые находятся на расстоянии. Для таких целей были придуманы программы-мессенджеры. Однако, у каждого человека есть слова, которые он считает оскорбительными.

Цель данного исследования – создание мессенджера с фильтром слов на языке ObjectPascal в среде Delphi.

В статье О.В. Андронниковой был рассмотрен такой мессенджер, как Telegram [1]. А.И. Соловьёв предложил использовать Telegram как медиаресурс [2]. Н.С. Дорохов создал веб-приложение, являющееся корпоративным мессенджером [3]. В.В. Полещук и К.Р. Абдалимова нашли применение чату “Wechat” в качестве практических занятий иностранным языкам (в данном случае – китайскому) [4]. А.В. Владзимирский рассмотрел все плюсы и минусы использования мессенджера “Whatsapp” в клинической медицине [5]. Также, в мессенджере “Telegram” А.В. Линд и А.А. Махов нашли эффективный инструмент для работ с целевыми аудиториями [6].

Существует несколько видов мессенджеров: текстовые, голосовые и видео-мессенджеры.

Видео-мессенджеры чаще всего представляют из себя видеотрансляцию от собеседника к собеседнику и обратно. Некоторые такие мессенджеры поддерживают групповые трансляции и возможность записывать видео и проигрывать его в любой момент времени после записи.

Голосовые мессенджеры представляют из себя мессенджер, основной функцией которого является запись голосового сообщения и отправка его собеседнику. Войс-чаты (voice chat) являются подгруппой голосовых чатов, т.к. их основная функция – передача голосовых сообщений в прямой трансляции, т.е. в момент записи сообщения.

Текстовые мессенджеры являются наиболее распространённым видом мессенджеров, т.к. их основная функция – передача текстовых сообщений. В таких мессенджерах проще всего реализовать такой функционал как: групповой обмен сообщениями, фильтрацию сообщений, обмен документами и изображениями. Также, текстовый мессенджер является наиболее распространённой частью смешанных мессенджеров, т.е. мессенджеров, сочетающих в себе функционал других мессенджеров. К примерам смешанных мессенджеров можно отнести Skype, сочетающего в себе функционал текстового, голосового и видео-мессенджера, Discord, сочетающего в себе голосовой и текстовый мессенджеры а также функционал видео-мессенджера, реализованного в виде захвата изображения с экрана и трансляции его собеседнику, и WhatsUp и Telegram, текстовых мессенджеров, способных записывать голосовые сообщения и передавать их, а также имеющих функционал передачи файлов.

Весь функционал текстового мессенджера сводится к трем задачам:

1. Подключиться к серверу/клиенту.
2. Передать сообщение.
3. Принять сообщение (при условии, что собеседник отправил ответное сообщение).

По такому принципу построены все мессенджеры. Однако если задача с передачей/приёмом сообщения частично лежит на пользователе, т.к. именно он должен писать/читать/отвечать на сообщения, то задача с подключением – это полностью задача программы. Поэтому процесс

подключения, обычно, скрывают от пользователя и выполняют в фоновом режиме, чтобы пользователя ничто не отвлекало от написания сообщений. Пример такого мессенджера №1 показан на рисунке 1.

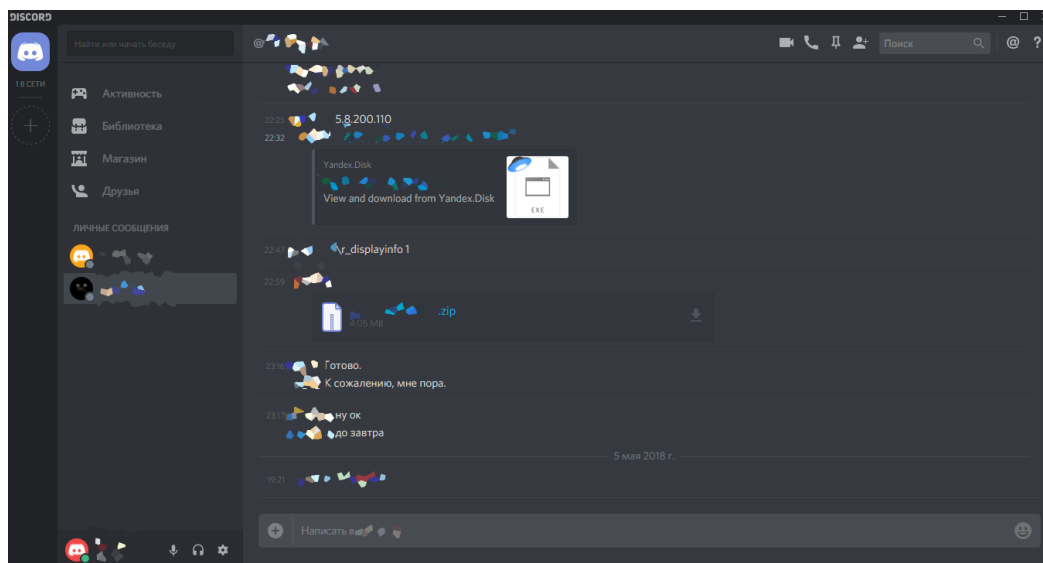


Рисунок 1. Discord

В нижнем окне пользователь пишет сообщение и отправляет его получателю. Есть функционал, связанный с файловой передачей, передачей звуковых сообщений и так называемые «смайлики», являющиеся картинками, которыми можно показать реакцию пользователя на сообщение. Также есть функция видео-трансляций и закрепления сообщений. Подключение к конкретному пользователю происходит в фоне и не отвлекает пользователя. При подключении к новому пользователю, программа предлагает добавить его в список «друзей» для облегчения дальнейшего поиска этого пользователя. Дополнительно, есть функционал групповых обменов сообщениями. Функционал активности, библиотеки и магазина является дополнением, не связанным с обменом сообщениями. Данный мессенджер стал популярным благодаря игровой индустрии, т.к. игрокам, играющим вместе в одну игру, требуется связь для координации действий и решений. Внутренние игровые способы общения зачастую неудобны, а до создания этого мессенджера, другие популярные мессенджеры были либо неудобными, либо сильно нагружали ЭВМ, что сказывалось как на процессе игры, так и на физическом состоянии ЭВМ. Пример №2 показан на рисунке 2.

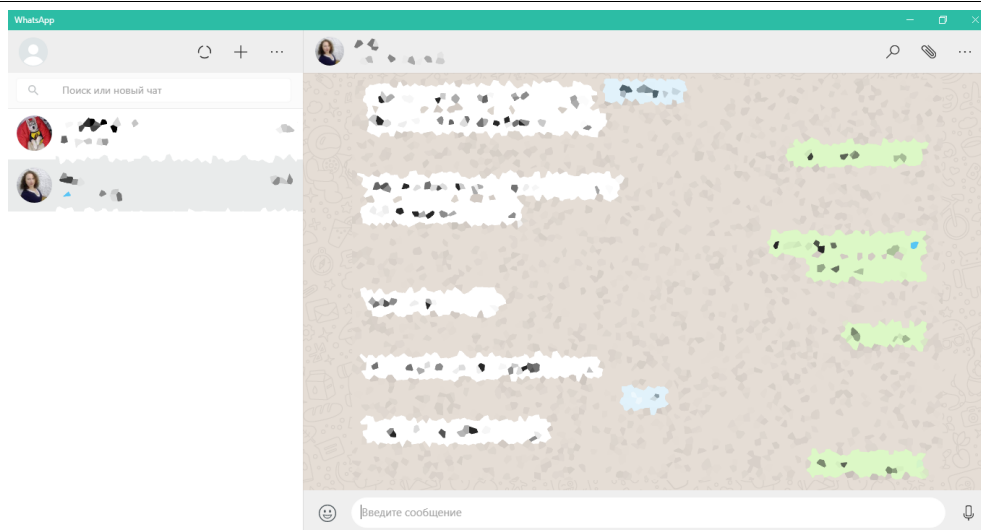


Рисунок 2. WhatsApp

Функционал практически идентичен предыдущему примеру, за исключением отсутствия прямой трансляции голоса, передачи видео с экрана и дополнительных функций вроде активности, библиотеки и магазина, однако присутствует возможность записать голосовое сообщение и отправить его собеседнику. Данный мессенджер, по сравнению с предыдущим примером, довольно прост в реализации и обращении, что и послужило причиной его популярности. У данного мессенджера изначально была реализация только на мобильных платформах, однако, с ростом популярности мессенджера, было создано данное приложение на персональных компьютерах. Но, у этого приложения есть существенный минус – для работы в этом приложении требуется мобильная платформа с предустановленной на ней мобильной версией мессенджера, которая будет синхронизироваться с мессенджером на персональных компьютерах. Пример №3 показан на рисунке 3.

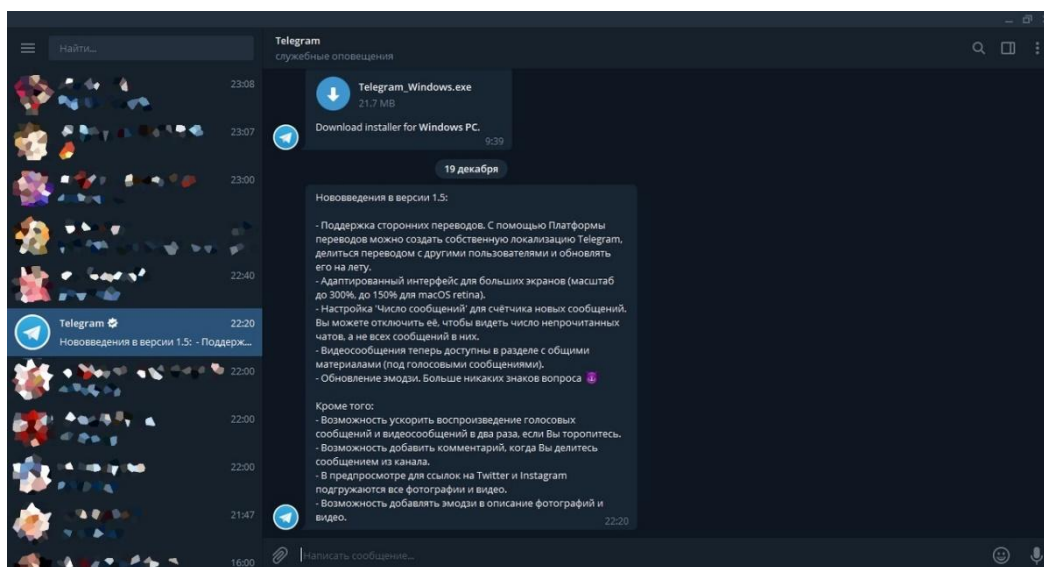


Рисунок 3. Telegram

Функционал аналогичен двум предыдущим мессенджерам. Есть возможность записывать и отправлять голосовые сообщения. Как и предыдущий пример, этот мессенджер был изначально разработан для мобильных платформ. У данного мессенджера есть функция синхронизации с мобильной версией мессенджера, однако, в отличие от предыдущего примера, отпала строгая необходимость в мобильной версии, т.к. версия мессенджера для персональных компьютеров может работать как отдельно, так и в синхронизации с версией мессенджера для мобильных платформ.

Запускаем программу – сервер.

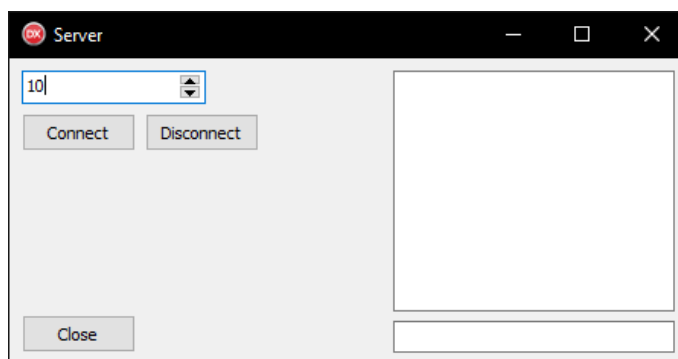


Рисунок 4. Программа-сервер

В этой программе в верхнем левом окне устанавливаем порт, по которому будут передаваться сообщения между сервером и клиентом. Нажимаем кнопку «Connect». Внешне ничего не изменится.

Теперь запускаем программу – клиент.

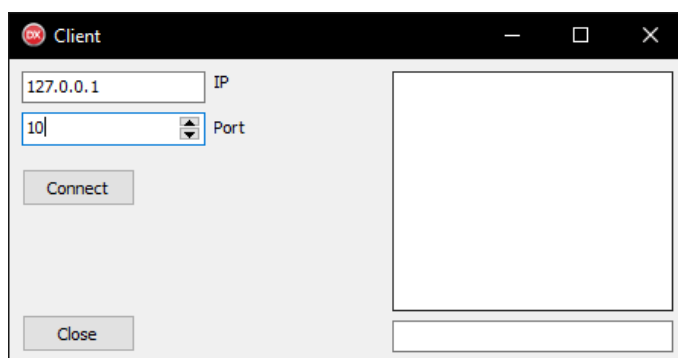


Рисунок 5. Программа-клиент

По умолчанию, в окне «IP» будет стоять ip локальной сети (127.0.0.1), но его можно заменить на ip ЭВМ, на котором будет запущена программа-сервер. Вводим в поле «Port» номер порта, на котором сервер открыл канал связи. Нажимаем кнопку «Connect». Внешне ничего не изменится.

Теперь по данному каналу между сервером и клиентом установлена связь. Связь будет поддерживаться только между первым присоединившимся клиентом и сервером. Набирать сообщение нужно в нижнем правом поле ввода. Отправка сообщения осуществляется нажатием на клавиатуре

клавиши «Enter». Исходящие сообщения будут помечены символом “>”. Входящие – “<”.

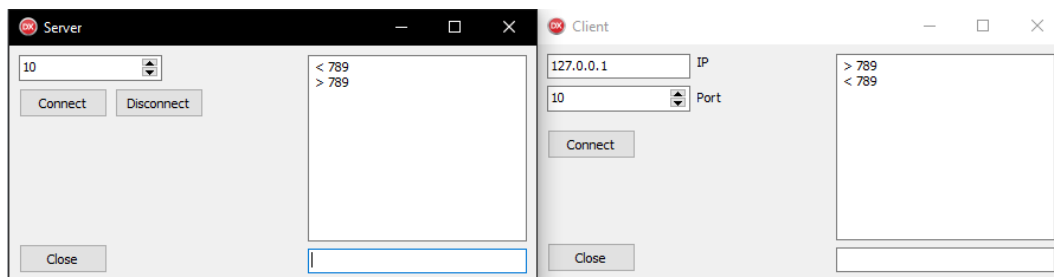


Рисунок 6. Пример отправки – приёма сообщений

В папках с программами есть текстовые файлы «Sensor.txt». Эти файлы являются списком сообщений или частей сообщений, которые клиент или сервер хотят убрать из своих входящих сообщений.

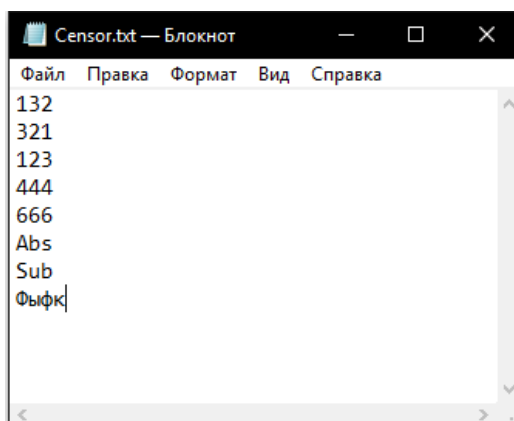


Рисунок 7. Пример файла «Sensor.txt» для сервера

Если клиент попытает отправить сообщение, содержащее одно из этих «слов», то данное «слово» будет заменено на «[REDACTED]».

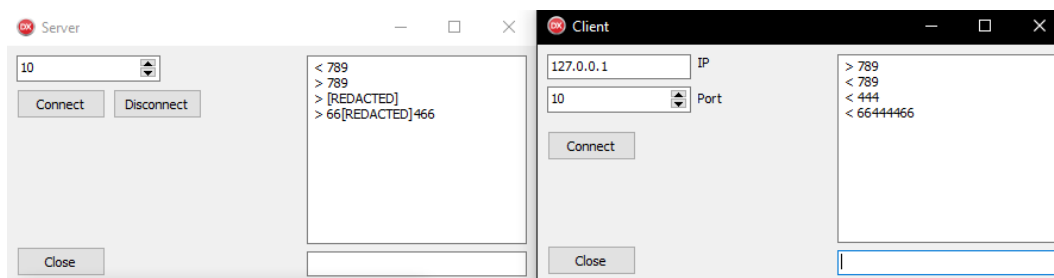


Рисунок 8. Пример работы алгоритма цензуры

Если изменить файл «Sensor.txt» во время работы программы, то программа автоматически обновит свой внутренний список цензурных слов.

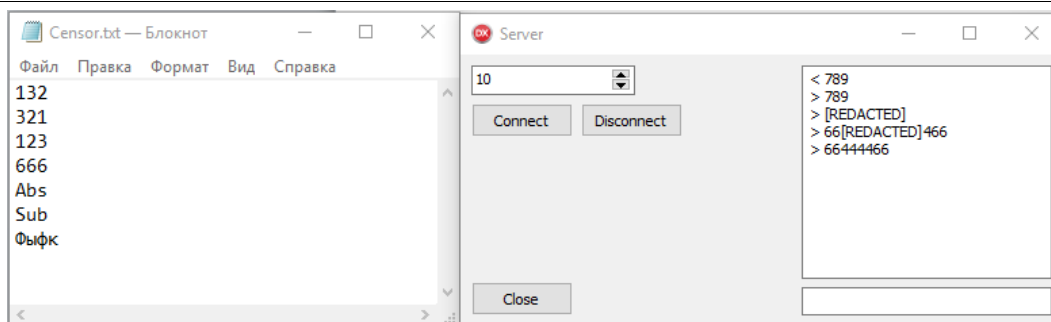


Рисунок 9. Пример обновлённого списка цензуры и работы программы-сервера

На рисунке 9 можно увидеть, что клиент повторно отправил своё последнее сообщение, но уже после обновления списка цензуры у сервера. А так как по новому списку цензуры сообщение не является и не содержит в себе запрещённых частей, сообщение дошло до сервера без изменений. Этот же алгоритм работает и у клиента для сообщений, отправленных со стороны сервера.

При нажатии кнопки «Close» происходит закрытие программы. При этом, если программа-клиент была закрыта до того, как закроется программа-сервер, у сервера высветится сообщение о том, что клиент отсоединился от канала связи.

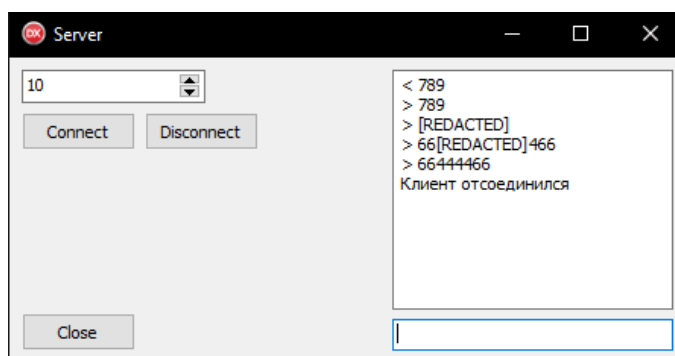


Рисунок 10. Пример сообщения об отключении клиента от канала связи

Нажатие кнопки «Disconnect» отключает сервер от канала связи. Клиент, при этом, останется подключён к каналу, но не сможет передавать сообщения серверу. Это необходимо для переключения сервера на другой канал связи.

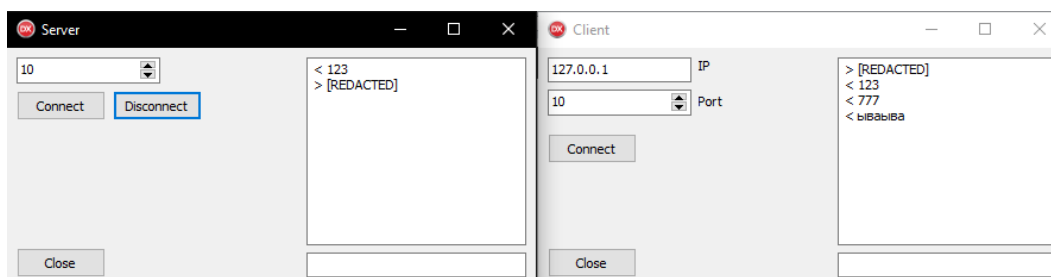


Рисунок 11. Сервер отсоединился от канала связи после первых 2 сообщений

В результате изучения теории о языке программирования ObjectPascal были созданы программы для создания одноканального текстового мессенджера с фильтром цензуры. Данный мессенджер может быть удобен для общения строго между 2 людьми.

Delphi является программой-компилятором ObjectPascal, способная создавать простые программы с графическим наполнением. Данная программа будет удобной как для обучения программированию на языке ObjectPascal, так и для создания небольших некоммерческих программ или программ с узкой кратковременной специализацией. Язык ObjectPascal является довольно простым и удобным языком программирования, т.к. он не очень сильно отличается от Pascal, который довольно часто изучают по школьной программе для 7-11 классов. Данный язык практически интуитивно понятен и потому практически не требует особо сложных навыков программирования.

### **Библиографический список**

1. Андронникова О.В. Мессенджер "Telegramm" как канал коммуникации // В сборнике: Журналистика, мультимедиа: информационный и социокультурный потенциал Материалы I Международной научно-практической конференции, посвященный памяти Г.М. Соловьева. 2018. С. 180-183.
2. Соловьев А.И. Международный мессенджер Telegram как медийный ресурс // В книге: Веб-программирование и интернет-технологии WebConf2018 тезисы докладов 4-й Международной научно-практической конференции. 2018. С. 133-134.
3. Дохоров Н.С. Веб-приложение – корпоративный мессенджер // В книге: Веб-программирование и интернет-технологии WebConf2018 тезисы докладов 4-й Международной научно-практической конференции. 2018. С. 133-134.
4. Полещук В.В., Абдалимова К.Р. Мессенджер "Wechat" как средство овладения практическими навыками китайского языка // В сборнике: На перекрестке Севера и Востока (методологии и практики регионального развития) Материалы II Международной научно-практической конференции. Ответственный редактор: Леонова О.А. 2017. С. 263-265.
5. Владзимирский А.В. Мессенджер "Whatsapp" в клинической медицине: риски и преимущества // Здравоохранение. 2016. № 8. С. 42-47.
6. Линд А.В., Махов А.А. Мессенджер "Telegram" как эффективный инструмент в работе с целевыми аудиториями // В сборнике: Актуальные проблемы коммуникации: теория и практика Материалы VIII Всероссийской научно-практической конференции. 2016. С. 119-126.