**Содержание**

Введение………………………………………………………………………1

1 Общая часть……...…………………………………………………………12

1.1 Описание предметной области………...………………………………...12

1.2 Обзор аналогичных программных продуктов...………………………..15

2 Специальная часть………...………………………………………………..18

2.1 Аналитическая часть…………………...………………………………...18

2.1.1 Постановка задачи……...………………………………………………18

2.1.2 Разработка алгоритма работы бота………...………………………….19

2.1.3 Разработка диаграммы Use Case программы…………...…………….20

2.1.4 Обоснование выбора языка программирования…...…………………20

2.1.5 Обоснование выбора инструментальных средств……...…………….22

2.2 Практическая часть…………...………………………………………….23

2.2.1 Разработка Telegram-бота……………...………………………………23

2.2.2 Регистрация Телеграмм-бота…………...……………………………...26

2.2.3 Формирование запросов Telegram-бота……………...……………….27

2.2.4 Разработка информационной структуры………………...……………33

2.3 Описание Telegram-бота…………………………...…………………….35

2.4 Разработка инструкции пользователя………………………………...…36

2.5 Мероприятия по охране труда и противопожарной безопасности……39

2.5.1 Описание требований по технике безопасности при работе  
 с вычислительной техникой………………………………………………………39

2.5.2 Характеристика условий труда программиста……………………….39

2.5.3 Эргономические требования к рабочему месту……………………...40

Заключение……………………………………………………………………43

Список литературы…………………………………………………………...44

**Введение**

Telegram – кроссплатформенная система мгновенного обмена сообщениями (мессенджер) с функциями VoIP, позволяющая обмениваться текстовыми, голосовыми и видео сообщениями, стикерами и фотографиями, файлами многих форматов. Также можно совершать видео- и аудио звонки и трансляции в каналах и группах, организовывать конференции, многопользовательские группы и каналы. С помощью ботов функционал приложения практически не ограничен. Клиентские приложения Telegram доступны для Android, iOS, Windows Phone, Windows, macOS и GNU/Linux. Количество ежемесячных активных пользователей сервиса по состоянию на январь 2021 года составляет около 500 млн человек. В августе 2017 года в своём Telegram-канале Павел Дуров заявил, что количество пользователей мессенджера ежедневно увеличивается более чем на 600 тысяч.

При помощи специального API сторонние разработчики могут создавать «ботов», специальные аккаунты, управляемые программами. Типичные боты отвечают на специальные команды в персональных и групповых чатах, также они могут осуществлять поиск в интернете или выполнять иные задачи, применяются в развлекательных целях или в бизнесе.

Преимущества телеграмм ботов для бизнеса:

* экономия ресурсов;
* интерактивность при взаимодействии с клиентом;
* масштабируемость;
* минимизация рутинных трудоемких задач.

**Актуальность** разработки чат-ботов заключается в следующем:

* удобство эксплуатации;
* минимизация человеческого ресурса;
* автоматизация.

**Практическая значимость** заключается в финансовой выгоде, путем монетизации и\или оптимизации ресурсов автоматизованных с помощью бота, а также дальнейшее развитие продукта путем добавления нового функционала.

**Объектом исследования** является процесс изучения среды разработки телеграмм-ботов.

**Предметом исследования** является кроссплатформенный телеграмм-бот для системы технической поддержки системы OsTicket.

**Целью работы** является изучение сферы разработки телеграмм-ботов, а также разработка своего собственного телеграмм-бота.

**1 Общая часть**

* 1. **Описание предметной области**

Боты – это специальные программы, выполняющие различные функции и упрощающие жизнь их пользователей. Написанные для платформы Telegram, они предназначены для выполнения самых разных функций: от получения новостей до поиска информации и даже торговли акциями. Главное задачей бота является автоматический ответ после введенной ему пользователем команды. При этом, работая непосредственно через интерфейс Telegram, программа имитирует действия живого юзера, за счет чего пользование таким ботом гораздо удобно и понятно.

Именно поэтому, многие компании, развивающие бизнес через интернет, используют возможности ботов по нескольким причинам:

1. они позволяют задействовать очередной канал коммуникации с целевой аудиторией (в России Телеграмм пользуется около 10 миллионов человек);
2. они быстро выполняют однообразную работу, позволяя разгрузить наемных сотрудников, тем самым экономя деньги компании.

В Telegram используется один общий вид ботов, которых от обычных юзеров отличает только наличие приставки «bot» в имени. Сами же боты делятся на несколько направлений:

* Чат-боты. Представляют из себя простейший чат, имитирующий общение на заданную пользователем тематику;
* Боты-информаторы. Отдельный вид ботов, главная цель которых – информирование пользователя о тех или иных событиях (новости, мероприятия, публикации и т. п.);
* Игровые боты. Боты, в которых можно поиграть в различные игры. По большей части, это текстовые версии разных игр времен Combats.ru;
* Боты-ассистенты. Боты, разработанные различными онлайн-сервисами как дополнение к основной веб-версии.

На самом деле, четкого деления нет, т. к. некоторые боты содержат сразу несколько механик и успешно выполняют множество пользовательских задач. С их помощью можно переводить, обучаться, тестировать, искать информацию, играть в игры и даже пользоваться другими сервисами и взаимодействовать с вещами, имеющими выход в глобальную сеть (популярный нынче «интернет-вещей»).

Благодаря этому боты стали карманными помощниками, пользоваться которыми можно, даже не покидая мессенджер. Они предоставляют возможность решать элементарные задачи при помощи мгновенных команд, при всех этих программах не нуждаются в установке и не занимают отдельное место в памяти вашего девайса.

Алгоритм работы бот-утилит достаточно прост. Сообщения, команды и запросы, отправленные пользователями, передаются на программное обеспечение, запущенное на серверах разработчиков. Посреднический анонимный сервер Telegram обрабатывает шифрование и осуществляет обратную связь между утилитой и пользователем.

Взаимодействие между пользователем и ботом выглядит следующим образом:

* 1. пользователь бота отдает ему команду;
  2. бот передает команду на сервер;
  3. программа на сервере обрабатывает полученный от бота запрос;
  4. сервер отдает ответ боту;
  5. бот выводит ответ на экран приложения пользователю.

И этот цикл повторяется раз за разом, когда вы нажимаете на кнопки и взаимодействуете с любым телеграмм-ботом.

Вы общаетесь с серверами с помощью простого HTTPS-интерфейса, который представляет собой упрощенную версию API Telegram. По-другому этот интерфейс можно назвать программным каталогом или бот-алгоритмом. Новые bot-утилиты создаются с помощью специальной утилиты @BotFather, который значительно упрощает процесс разработки.

OsTicket - это система организация технической поддержки пользователей в режиме реального времени, включая отслеживание потоков электронной почты, вход на почтовый сервер по протоколу POP3, неограниченное количество адресов email, методы избежания зацикливания автоответчиков, и др.

**OsTicket -** это альтернатива более дорогих и сложных систем поддержки клиентов. Простой в настройке, лёгкий и надёжный Web-интерфейс. Но самое главное, это абсолютно бесплатно.

OsTicket сделана чтобы упростить прием запросов, а также улучшить эффективность обслуживания ваших клиентов, предоставляя персоналу все необходимые средства для быстрой и эффективной поддержки. Некоторые основные возможности включают:

* Web и Email Поддержка. Запросы могут быть созданы по email, через онлайн форму или по телефону (создается персоналом). Гибкая конфигурация и отображение;
* Авто оповещение. Автоматическая отправка сообщения, когда открыта новая заявка или получено сообщение. Настраиваемые почтовые шаблоны;
* Готовые ответы. Готовые ответы для часто задаваемых вопросов;
* Внутренние Сообщения. Добавление внутренних сообщений к заявкам для персонала;
* Темы обращения. Настройка тем обращений для веб заявок;
* Предупреждения и уведомления. Сотрудники и клиенты получают оповещения по электронной почте о ходе работ над заявками. Уведомления имеют гибкие настройки;
* Контроль доступа. Полный контроль над персоналом по средствам групп и отделов;
* Назначение и передача заявок. Назначение заявок сотруднику или отделу;
* Без регистрации. Нет необходимости в регистрации пользователей (для входа используется ID/email);
* История**.** Сохраняются все запросы и ответы.
  1. **Обзор аналогичных программных продуктов**

Боты «Телеграм» универсальны и выполняют множество задач на любой вкус. Найти информацию, скачать книгу, фильм, музыку, развлечь – это лишь маленький перечень возможностей ботов.

Скачать видео с YouTube – [@Save\_youbot](https://t.me/Save_youbot)

Это бот для скачивания видео с YouTube. С его помощью вы сможете загрузить видео любого объема прямиком в Telegram

Изображение выглядит как текст, снимок экрана, монитор, черный

Автоматически созданное описаниеРисунок 1 – Телеграм-бот для скачивания видео с YouTube

Yandex.Translate – [@YTranslateBot](https://t.me/YTranslateBot). Инструмент, интегрированный с переводчиком Яндекса.

Изображение выглядит как текст, снимок экрана, монитор, экран

Автоматически созданное описание Рисунок 2 – Переводчик интегрированный с Telegram ботом

osTicketTelegram – телеграмм-бот интегрированный с системой OsTicket

Изображение выглядит как текст

Автоматически созданное описание  
Рисунок 3 – Телеграм-бот интегрированный с системой OsTicket

**2 Специальная часть**

**2.1 Аналитическая часть**

2.1.1 Постановка задачи

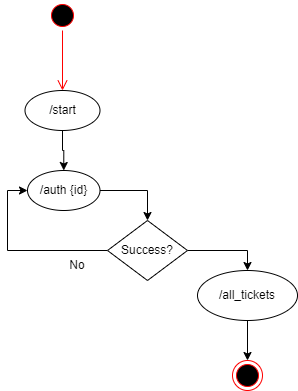
Создание бота включает в себя несколько этапов:

* скачивание и установка набора инструментов для разработки;
* загрузка и установка необходимых зависимостей;
* настройка среды программирования;
* регистрация бота;
* создание проекта;
* определение и создание команд;
* настройка и внедрение команд;
* написание кода и интеграция с внешними сервисами;
* тестирование и отладка.

2.1.2 Разработка алгоритма работы бота

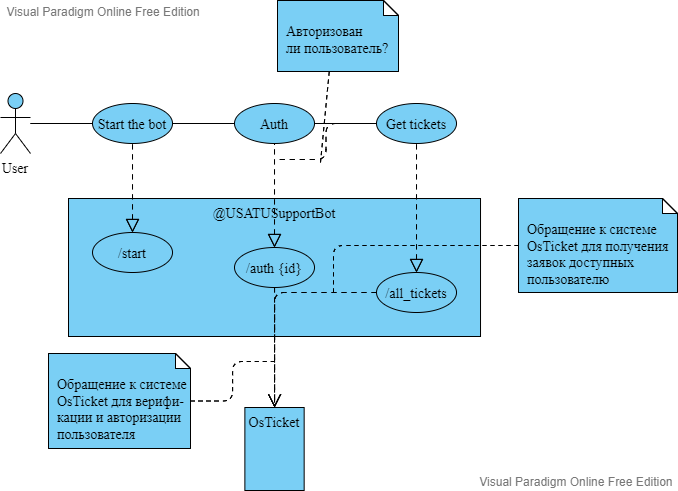
Бот обрабатывает команды, поступающие ему от пользователя (рисунок 4):

* «/start» – бот должен активизироваться и начать свою работу.
* «/auth {id}» - при активации бота пользователь авторизуется через данную команду вводя id, где id – идентификатор пользователя системы.
* «/all\_tickets» - при вводе данной команды, пользователь получает список доступных ему заявок в системе.

  
Рисунок 4 – Алгоритм работы бота

2.1.3 Разработка диаграммы Use Case программы

На диаграмме Use Case представлен актер и его роли, которые он выполняет. На рисунке 5 представлен 1 актер: user. Он может запустить бота, затем авторизоваться, а после получить список всех доступных ему заявок в системе.

****Рисунок 5 – Use Case диаграмма.

2.1.4 Обоснование выбора языка программирования

Для написания телеграмм-бота можно использовать разные технологии. Наибольшей популярностью для создания телеграмм-ботов пользуются Python, Java и еще с десяток языков программирования из рейтинга TIOBE. В итоге я оставил только два языка программирования которые полностью удовлетворяли мои потребности.

Python – высокоуровневый язык программирования общего назначения с динамической строгой типизацией и автоматическим управлением памятью, ориентированный на повышение производительности разработчика, читаемости кода и его качества, а также на обеспечение переносимости написанных на нём программ. Язык является полностью объектно-ориентированным в том плане, что всё является объектами. Необычной особенностью языка является выделение блоков кода пробельными отступами. Синтаксис ядра языка минималистичен, за счёт чего на практике редко возникает необходимость обращаться к документации. Сам же язык известен как интерпретируемый и используется в том числе для написания скриптов. Недостатками языка являются зачастую более низкая скорость работы и более высокое потребление памяти написанных на нём программ по сравнению с аналогичным кодом, написанным на компилируемых языках, таких как C или C++.

Java - это полнофункциональный объектно-ориентированный язык, который поддерживает все три «столпа» объектно-ориентированного программирования: инкапсуляцию, наследование и полиморфизм. Он имеет прекрасную поддержку компонентов, надежен и устойчив благодаря использованию «сборки мусора», обработки исключений, безопасности типов. Язык Java разрабатывался с нуля и вобрал в себя много полезных свойств таких языков, как C++, Visual Basic, а также Pascal и Delphi.

При этом необходимость обратной совместимости с предыдущими версиями отсутствовала, что позволило языку Java избежать многих отрицательных сторон своих предшественников. Telegram позволяет использовать любой язык программирования. Из всех языков программирования я решил выбрать Java, так как это очень перспективный язык и достаточно простой для освоения.

Основная идея Java – это универсальность. Можно реализовать все на одном языке и для любой платформы. Он неплохо подходит как для начинающих в качестве первого языка, так и для реализации крупных коммерческих проектов.

В качестве языка программирования для реализации данного дипломного проекта был выбран язык программирования Java из-за наличия в нем статической типизации.

2.1.5 Обоснование выбора инструментальных средств

Для разработки на Java существует несколько популярных сред разработки: Eclipse и IntelliJ Idea.

Eclipse – свободная интегрированная среда разработки модульных кроссплатформенных приложений. Развивается и поддерживается Eclipse Foundation.

Наиболее известные приложения на основе Eclipse Platform – различные «Eclipse IDE» для разработки ПО на множестве языков (например, наиболее популярный «Java IDE», поддерживавшийся изначально, не полагается на какие-либо закрытые расширения, использует стандартный открытый API для доступа к Eclipse Platform).

В данной работе была использована среда разработки – IntelliJ Idea. Эта программа обеспечивает высокое качество кода на протяжении всего цикла жизни ПО, от проектирования до внедрения. Какие бы приложения ни разрабатывались: для Windows, Windows Phone, Android, Ios и других платформ, Idea – это универсальное решение. Сегодня это основное и самое эффективное средство разработки решений для платформы Java. Idea продолжает традиции, заложенные предыдущими версиями продукта, и содержит множество новинок, предназначенных как для упрощения создания приложений для различных платформ отдельными, так и для повышения производительности всей проектной команды. Поэтому мною была выбрана именно эта среда разработки.

**2.2 Практическая часть**

2.2.1 Разработка Telegram-бота

Исходя из поставки задач были проведены следующие работы:

* установка IntelliJ Idea;
* установка JDK;
* создание Maven проекта;
* добавление зависимостей;
* написание кода;
* тестирование и исправление ошибок.

Устанавливаем IntelliJ Idea с последующей настройкой – рисунок 6.

Изображение выглядит как текст

Автоматически созданное описание  
Рисунок 6 – Установка IntelliJ Idea

Устанавливаем JDK – рисунок 7.

Изображение выглядит как текст

Автоматически созданное описание  
Рисунок 7 – Установка JDK

Рисунок 8 – создание Maven проекта.

Изображение выглядит как текст, снимок экрана, монитор, черный

Автоматически созданное описание  
Рисунок 8 – Создаем Maven проект

Добавляем необходимые зависимости в pom.xml – рисунок 9.

Изображение выглядит как текст

Автоматически созданное описание  
Рисунок 9 – Добавление необходимых зависимостей

2.2.2 Регистрация Телеграмм-бота

Что бы бот был доступен в телеграмме его необходимо зарегистрировать в другом бот, BotFather. При регистрации необходимо указать уникальный идентификатор для бота - @USATUSupportBot, в данном случае, и имя бот, USATU Support.

Изображение выглядит как текст, снимок экрана, монитор, экран

Автоматически созданное описание  
Рисунок 10 – Регистрация бота в телеграмме

После регистрации BotFather отправит в ответ уникальный идентификатор телеграмм-бота в системе телеграмм, в данном случае - *5185789510:*AAEBENUnLBle0bkpd1EMXbjM9E*\_i0YypLP4*

2.2.3 Формирование запросов Telegram-бота

Что бы наш бот мог обрабатывать команды, поступающие ему от пользователя, необходимо написать соответствующий код.

Для начала следует научить бота приветствовать пользователя, для этого он должен уметь обрабатывать команду «/start», ниже представлен код, обрабатывающий данную команду.

На рисунке 11 представлен результат работы данной команды.

Изображение выглядит как текст, монитор, снимок экрана, экран

Автоматически созданное описаниеРисунок 11 – Приветствие пользователя

Далее необходимо научить бота обрабатывать команду «/auth {id}», чтобы пользователь мог регистрироваться в нем. Ниже представлен код, обрабатывающий данную команду.

На рисунке 12 представлен результат работы данной команды.

Изображение выглядит как текст, монитор, снимок экрана, экран

Автоматически созданное описание  
Рисунок 12 – авторизация пользователя

Далее необходимо обучить бота возвращать заявки конкретного пользователя (пользователя что прошел успешную авторизацию).

На рисунке 13 показан результат работы.

Изображение выглядит как текст, снимок экрана, экран, компьютер

Автоматически созданное описаниеРисунок 13 – Получение заявок

2.2.4 Разработка информационной структуры

В качестве метода хранения данных используется база данных h2.

H2 – открытая кроссплатформенная СУБД, полностью написанная на языке Java.

Несмотря на малый размер (чуть более 1 МБ) H2 поддерживает следующие возможности «из коробки»:

* два режима работы (клиент-сервер, встроенный);
* два режима хранения данных (файловая система, память);
* поддержка планов выполнения запросов;
* поддержка кластеризации и репликации;
* шифрование данных;
* внешние (связанные) таблицы;
* драйвер odbc;
* полнотекстовый поиск;
* определение доменов;
* мультиверсионный конкурентный доступ;
* поддержка последовательностей;
* поддержка ключевых слов limit и offset в запросах;
* временные таблицы;
* вычисляемые столбцы;
* пользовательские агрегатные функции;
* пользовательские хранимые процедуры;
* сжатие clob/blob объектов;
* работа с csv файлами на чтение и запись;
* браузерная консоль управления;
* запуск как сервис windows.

Основные причины почему была выбрана именно эта база данных:

* легкая интеграция с java-приложениями;
* кроссплатформенность;
* большая защищенность, чем у нативных приложений;
* пользовательские функции и триггеры работают очень быстро;
* поддержка юникода.

СУБД весьма полезна в разработке, так как позволяет создавать тестовые наборы данных прямо в памяти, имеет развитые возможности запросов, многочисленные функции, совместимые с более тяжеловесными БД, легко настраивается, имеет подробную и понятную документацию. Для небольших программ СУБД возможно использовать саму по себе, так как она легко встраивается в приложение.

**2.3 Описание Telegram-бота**

Телеграм-бот имеет название «USATU Support» и имеет уникальный идентификатор в системе телеграмм @USATUSupportBot, по которому каждый пользователь телеграмм сможет воспользоваться им.

Что бы начать пользоваться ботом необходимо отправить ему команду «/start» и под начнет функционировать, отправив в ответ смайл и текстовое сообщение с приветствием.

Для полноценного использования бота пользователь должен авторизоваться в нем.

Для это необходимо отправить команду «/auth» с указанием идентификатора пользователя в системе osTicket. Если авторизация пройдет успешно – бот покажет меню дополнительных команд доступных только авторизованному пользователю. Если успешной авторизации не произошло, например пользователя нет в системе или введены не верные данные – бот покажет соответствующее сообщение.

При успешной авторизации бот представляет меню дополнительных команд, в котором есть команда «/all\_tickets». Используя данную команду, пользователь может получить все доступные ему заявки, созданные в системе osTicket. В случае если что-то пойдет не так, например произойдет сбой сети или по каким-то причинам система osTicket будет не доступна бот отправит в ответ соответствующее сообщение.

**2.4 Разработка инструкции пользователя**

Что бы запустить телеграмм-бота введите команду «/start», как показано на рисунке 14.

Изображение выглядит как текст, монитор, снимок экрана, экран

Автоматически созданное описаниеРисунок 14 – Команда «/start»

Далее, чтобы пользоваться полным функционалом бота, необходимо авторизоваться. Для этого введите команду «/auth {id}», где вместо {id} укажите идентификатор пользователя в системе osTicket, как показано на рисунке 15.

Изображение выглядит как текст, монитор, снимок экрана, экран

Автоматически созданное описаниеРисунок 15 – Команда «/auth»

Что бы получить заявки выберите в меню команду «/all\_tickets», как показано на рисунке 16.

Изображение выглядит как текст, снимок экрана, экран, компьютер

Автоматически созданное описание  
Рисунок 16 – команда «/all\_tickets».**2.5 Мероприятия по охране труда и противопожарной безопасности**

2.5.1 Описание требований по технике безопасности при работе с вычислительной техникой.

С развитием научно-технического прогресса немаловажную роль играет возможность безопасного исполнения людьми своих трудовых обязанностей. В связи с этим была создана и развивается наука о безопасности труда и жизнедеятельности человека.

Безопасность жизнедеятельности – это комплекс мероприятий, направленных на обеспечение безопасности человека в среде обитания, сохранение его здоровья, разработку методов и средств защиты путем снижения влияния вредных и опасных факторов до допустимых значений, выработку мер по ограничению ущерба в ликвидации последствий чрезвычайных ситуаций мирного и военного времени.

На рабочем месте должны быть предусмотрены меры защиты от возможного воздействия опасных и вредных факторов производства. Уровни этих факторов не должны превышать предельных значений, оговоренных правовыми, техническими и санитарно-техническими нормами. Эти нормативные документы обязывают к созданию на рабочем месте условий труда, при которых влияние опасных и вредных факторов на работающих либо устранено совсем, либо находится в допустимых пределах.

2.5.2 Характеристика условий труда программиста

Научно-технический прогресс внес серьезные изменения в условия производственной деятельности работников умственного труда. Их труд стал более интенсивным, напряженным, требующим значительных затрат умственной, эмоциональной и физической энергии. Это потребовало комплексного решения проблем эргономики, гигиены и организации труда, регламентации режимов труда и отдыха.

В настоящее время компьютерная техника широко применяется во всех областях деятельности человека. При работе за компьютером человек подвергается воздействию ряда опасных и вредных производственных факторов: электромагнитных полей инфракрасного и ионизирующего излучений, шума и вибрации, статического электричества. Недостаточность освещения приводит к напряжению зрения, ослабляет внимание, приводит к наступлению преждевременной утомленности. Чрезмерно яркое освещение вызывает ослепление, раздражение и резь в глазах. Неправильное направление света на рабочем месте может создавать резкие тени, блики, дезориентировать работающего. Все эти причины могут привести к несчастному случаю или профзаболеваниям, поэтому столь важен правильный расчет освещенности. В помещениях вычислительных центров необходимо применить систему комбинированного освещения. При выполнении работ категории высокой зрительной точности величина коэффициента естественного освещения должна быть не ниже 1,5%, а при зрительной работе средней точности должен быть не ниже 1,0%. Ученые считают, что как кратковременное, так и длительное воздействие всех видов излучения от экрана монитора не опасно для здоровья персонала, обслуживающего компьютеры. Однако исчерпывающих данных относительно опасности электромагнитного излучения от мониторов не существует, а вот ионизирующее очень опасно, если излучение имеет высокую частоту, оно способно разрушать ДНК и клетки, выбивая электроны из атомов. Такой вид излучения мы называем радиацией. Никакие приборы ни при какой мощности не смогут оказать ионизирующий эффект. Для этого нужна очень высокая частота (от 75000 гц).

2.5.3 Эргономические требования к рабочему месту

Проектирование рабочих мест, снабженных видеотерминалами, относится к числу важных проблем эргономического проектирования в области вычислительной техники. Рабочее место и взаимное расположение всех его элементов должно соответствовать антропометрическим, физическим и психологическим требованиям. Большое значение имеет также характер работы. В частности, при организации рабочего места программиста должны быть соблюдены следующие основные условия: оптимальное размещение оборудования, входящего в состав рабочего места и достаточное рабочее пространство, позволяющее осуществлять все необходимые движения и перемещения. Эргономическими аспектами проектирования видеотерминальных рабочих мест являются: высота рабочей поверхности, размеры пространства для ног, требования к расположению документов на рабочем месте, характеристики рабочего кресла, требования к поверхности рабочего стола, регулируемость элементов рабочего места. Главными элементами рабочего места программиста являются стол и кресло. Основным рабочим положением является положение сидя. Рабочая поза сидя вызывает минимальное утомление программиста. Рациональная планировка рабочего места предусматривает четкий порядок и постоянство размещения предметов, средств труда и документации. То, что требуется для выполнения работ чаще, расположено в зоне легкой досягаемости рабочего пространства. Дисплей размещается в центре, системный блок размещается в предусмотренной нише стола.

Для комфортной работы стол должен удовлетворять следующим условиям: высота стола должна быть выбрана с учетом возможности сидеть свободно, в удобной позе, при необходимости опираясь на подлокотники; нижняя часть стола должна быть сконструирована так, чтобы программист мог удобно сидеть, не был вынужден поджимать ноги; поверхность стола должна обладать свойствами, исключающими появление бликов в поле зрения программиста; конструкция стола должна предусматривать наличие выдвижных ящиков; высота рабочей поверхности рекомендуется в пределах 680-760мм.

Высота поверхности, на которую устанавливается клавиатура, должна быть около 650мм; большое значение придается характеристикам рабочего кресла. Так, рекомендуемая высота сиденья над уровнем пола находится в пределах 420–550 мм. Поверхность сиденья мягкая, передний край закругленный, а угол наклона спинки - регулируемый. Положение экрана определяется: расстоянием считывания (0,6…0,7м); углом считывания, направлением взгляда на 20 градусов ниже горизонтали к центру экрана, причем экран перпендикулярен этому направлению. Должна также предусматриваться возможность регулирования экрана: по высоте +3 см; по наклону от -10 до +20 градусов относительно вертикали; в левом и правом направлениях. Большое значение также придается правильной рабочей позе пользователя. При неудобной рабочей позе могут появиться боли в мышцах, суставах и сухожилиях. Требования к рабочей позе пользователя видеотерминала следующие: голова не должна быть наклонена более чем на 20 градусов; плечи должны быть расслаблены; – локти - под углом 80 -100 градусов; предплечья и кисти рук - в горизонтальном положении. Причина неправильной позы пользователей обусловлена следующими факторами: нет хорошей подставки для документов, клавиатура находится слишком высоко, а документы - низко, некуда положить руки и кисти, недостаточно пространство для ног.

В целях преодоления указанных недостатков даются общие рекомендации: лучше передвижная клавиатура; должны быть предусмотрены специальные приспособления для регулирования высоты стола, клавиатуры и экрана, а также подставка для рук. Во время пользования компьютером медики советуют устанавливать монитор на расстоянии 50–60 см от глаз. Специалисты также считают, что верхняя часть видеодисплея должна быть на уровне глаз или чуть ниже. Когда человек смотрит прямо перед собой, его глаза открываются шире, чем, когда он смотрит вниз. За счет этого площадь обзора значительно увеличивается, вызывая обезвоживание глаз.

**Заключение**

Мессенджеры плотно вошли в нашу жизнь. Люди общаются с друзьями и родственниками, устраивают деловые встречи или просто весело проводят время с друзьями находясь на расстоянии друг от друга. Помимо бытового использования ботов их можно использовать в бизнесе, например для системы технической поддержки или отслеживания посылок по всему миру.

В аналитической части была проведена установка JDK, IntelliJ IDEA и других компонентов.

В практической части был рассмотрен процесс построения телеграмм-бота. Далее предоставлена инструкция для обычного пользователя.

Для достижения поставленной цели были решены следующие задачи:

* настройка среды программирования;
* создание нового проекта;
* настройка и внедрение команд для бота;
* написание кода и интеграция со сторонними сервисами;
* тестирование и отладка;
* сдача бота на рецензирование.

Сначала работы над ботом была проделана большая работа, от настройки системы OsTicket до написания запросов к ней из исходного кода программы. Процесс был разбит на несколько этапов:

* настройка системы osticket;
* регистрация телеграмм-бота в телеграмме;
* разработка сценариев использования;
* разработка алгоритма работы;
* написание исходного кода и интеграция с системой osticket;
* подключение базы данных;
* сквозное тестирование;
* исправление ошибок, выявленных в ходе тестирования.

В процессе разработки телеграмм-бота были исследованы и изучены некоторые аспекты дизайна кода, тестирования, принципов написания хорошего программного кода.

Также в процессе написания кода были изучены механизмы интеграций разных сервисов, протокол http и https, принципы и механизмы интеграционного тестирования.

Был исследован процесс постановки задач и написания технической документации для использования конечным пользователем, а также визуализация алгоритмов работы программного продукта.

В ходе выполнения данной дипломной работы удалось ближе познакомиться с внутренними механизмами работы мессенджера Telegram, а также механизмов работы телеграмм-ботов в вышеупомянутом мессенджере.

Однако, в ходе разработки данной дипломной работы были исследованы не только механизмы работы телеграмм-ботов, а также механизмы работы самого мессенджера, но также и другие варианты реализации подобных механизмов в других популярных мессенджерах и платформах, например, подобные боты также могут быть реализованы на платформе Slack и\или Discord, а также в наиболее популярной мессенджере для мобильных устройств – WhatsApp.

Исходя из вышенаписанных слов, можно с уверенностью говорить о том что такой механизм как бот не ограничен одной платформой, механизмы интеграций изученные в ходе выполнения данной дипломной работы прекрасно переносятся и на другие мессенджеры и прочие платформы для общения людей.

**Список литературы**

1. ГОСТ 19.701–90. Схемы алгоритмов, программ, данных и систем. Условные обозначения и правила выполнения. - Взамен ГОСТ 19.002–80, ГОСТ 19.003–80; введ. 01.01.1992. - М.: Межгосударственный совет по стандартизации, метрологии и сертификации, 1991. - 23 с. - (Единая система программной документации).
2. ГОСТ 19.102–77. Стадии разработки. - Введ. 01.01.1980. - М.: Межгосударственный совет по стандартизации, метрологии и сертификации, 1991. - 2 с. - (Единая система программной документации).
3. ГОСТ 19.201–78. Техническое задание. Требования к содержанию и оформлению. - Введ. 01.01.1980. - М.: Межгосударственный совет по стандартизации, метрологии и сертификации, 1991. - 2 с. - (Единая система программной документации).
4. ГОСТ 19.402–78. Описание программы. Введ. 01.01.1980. - М.: Межгосударственный совет по стандартизации, метрологии и сертификации, 1991. - 2 с. - (Единая система программной документации).
5. ГОСТ 19.404–79. Пояснительная записка. Требования к содержанию и оформлению. - Введ. 01.01.1981. - М.: Межгосударственный совет по стандартизации, метрологии и сертификации, 1991. - 2 с. - (Единая система программной документации).
6. ГОСТ 19.504–79. Руководство программиста. Требования к содержанию и оформлению. - Введ. 01.01.1980. - М.: Межгосударственный совет по стандартизации, метрологии и сертификации, 1991. - 1 с. - (Единая система программной документации).
7. ГОСТ 19.505–79. Руководство оператора. Требования к содержанию и оформлению. - Введ. 01.01.1980. - М.: Межгосударственный совет по стандартизации, метрологии и сертификации, 1991. - 2 с. - (Единая система программной документации).
8. ГОСТ 2.105–95. Общие требования к текстовым документам. - Взамен ГОСТ 2.105–79, ГОСТ 2.906–71; введ. 01.07.1996. - Минск: Межгосударственный совет по стандартизации, метрологии и сертификации, 1995. - 19 с. - (Межгосударственный стандарт. Единая система конструкторской документации).
9. ГОСТ 7.32.- 2001 «Система стандартов по информации, библиотечному и издательскому делу «Отчет о научно-исследовательской работе».
10. Хорстманн Кей С. Java. Библиотека профессионала. Том 1. Основы. 2018. -864 с.
11. Joshua Bloch. Effective Java 3rd. Addison-Wesley Professional, 2017- 416 c.
12. Хабитуев, Б. В. Программирование на языке Java: практикум: учебное пособие / Б. В. Хабитуев. – Улан-Удэ: БГУ, 2020. – 94 с.
13. Herbert Schildt. Java: a beginners guide 8th. McGraw Hill, 2018. - 720 с.
14. Herbert Schildt. Java: The Complete Reference 11th. McGraw Hill, 2019. – 1248 с.
15. Kathy Sierra, Bert Bates. Headfirst Java: A Brain-friendly Guide 2nd. O'Reilly Media, 2018. - 720 c.
16. Brian Goetz. Java Concurrency in Practice 1st. Addison-Wesley Professional, 2018. – 424 с.
17. Lasse Koskela. Test Driven: TDD and Acceptance TDD for Java Developers: Practical TDD and Acceptance TDD for Java Developers 1st. Manning Publications, 2018. – 470 с.
18. Scott Oaks. Java Performance: The Definitive Guide: Getting the Most Out of Your Code 1st. O'Reilly Media, 2018. – 426 с.
19. GitHub репозиторий с телеграмм-клиентом: <https://github.com/rubenlagus/TelegramBots/wiki>
20. GitHub репозиторий с описанием запросов к системе OsTicket: <https://github.com/BMSVieira/osticket-api>