ТЕХНІЧНИЙ КОЛЕДЖ ТЕРНОПІЛЬСЬКОГО НАЦІОНАЛЬНОГО ТЕХНІЧНОГО УНІВЕРСИТЕТУ ІМЕНІ ІВАНА ПУЛЮЯ

Циклова комісія програмних систем і комплексів

**КУРСОВА РОБОТА**

з дисципліни:

**«Розробка клієнт серверних застосувань»**

на тему: **«Розробка клієнт-серверного додатку для керування чат-ботом тижня комп’ютерних дисциплін»**

Студента  4  курсу групи ОПК-421 напряму підготовки 6.050101 «Комп’ютерні науки» спеціальності 5.05010101 «Обслуговування програмних систем і комплексів»

Покидко О.В.

(прізвище та ініціали)

Керівник: викладач   Капаціла  І.Б.         
Національна шкала \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Кількість балів: \_\_\_\_\_\_ Оцінка: ECTS \_\_\_\_\_\_

Члени комісії: \_\_\_\_\_\_\_\_\_\_\_ І.Б. Капаціла

(підпис)

\_\_\_\_\_\_\_\_\_ Р.О. Слободян

(підпис)

м. Тернопіль – 2019

ЗМІСТ

[ВСТУП 5](#_Toc11057792)

[1 ЗАГАЛЬНИЙ РОЗДІЛ 6](#_Toc11057793)

[1.1 Аналітичний огляд існуючих рішень 6](#_Toc11057794)

[1.2 Технічне завдання 6](#_Toc11057795)

[2 РОЗРОБКА ТЕХНІЧНОГО ТА РОБОЧОГО ПРОЕКТУ 9](#_Toc11057796)

[2.1 Постановка задачі на розробку програмного забезпечення 9](#_Toc11057797)

[2.2 Опис та обґрунтування вибору структури та методу організації  вхідних та вихідних даних 10](#_Toc11057798)

[2.3 Розробка алгоритму 10](#_Toc11057799)

[2.4 Визначення інформаційних зв’язків 10](#_Toc11057800)

[2.5 Написання текстів програм 10](#_Toc11057801)

[2.6 Тестування та налагодження програм 10](#_Toc11057802)

[3 СПЕЦІАЛЬНИЙ РОЗДІЛ 11](#_Toc11057803)

[3.1 Інструкція з інсталяції програмного забезпечення 11](#_Toc11057804)

[3.2 Інструкція з використання тестових наборів 11](#_Toc11057805)

[3.3 Інструкція з експлуатації програмного комплексу 11](#_Toc11057806)

[ВИСНОВКИ 12](#_Toc11057807)

[ПЕРЕЛІК ПОСИЛАНЬ 13](#_Toc11057808)

[Додаток А Лістинг файлу «main.cpp» 14](#_Toc11057809)

[Додаток Б Лістинг файлу «mainwindow.h» 15](#_Toc11057810)

[Додаток В CD-диск із програмним продуктом 16](#_Toc11057811)

ВСТУП

Велику роль в сучасному спілкуванні грають системи миттєвого обміну повідомленнями (месенджери), такі як Telegram, Viber, Skype і т.д. Основне їх призначення - обмін повідомленнями в реальному часі через мережу Інтернет. Можуть передаватися як текстові повідомлення, так і звукові сигнали, зображення, відео. Свою популярність вони здобули в зв'язку з наявністю як текстових, так і голосових чатів, можливістю створення конференцій на велику кількість людей. Більшість месенджерів мають в наявності додатки під мобільні пристрої, що дозволяє порівняти їх з вбудованою функцією надсилання SMS, проте месенджери не вимагають окремої плати за відправку повідомлення, також месенджери дозволяють відправляти мультимедійні дані. У порівнянні з соціальними мережами, які так само зазвичай включають функцію миттєвої відправки повідомлень, месенджери простіші у використанні, та в них відсутні типові для соціальних мереж функції, таки як створення спільнот, розміщення інформації про себе і т.д.

Одним з представників є відносно молодий додаток Telegram, який був створений 14 серпня 2013 року. Спочатку Telegram користувався популярністю переважно у людей інтелектуальних професій. Широка публіка вже встигла привикнути до WhatsApp і Viber, а новинка, у якої не було українськомовної та російськомовної версії, користувалась попитом IT-фахівців і зарубіжних країн - в основному розвиваються, Італії, Іспанії і Бразилії.

Не задовго після набуття чинності указу президента №133/2017 17 травня 2017 року, який передбачав блокування популярної серед молоді соціальної мережі “ВКонтакте”, Telegram випускає оновлення 4.4 у якому офіційно були добавленні Українська та Російська мови. Це значно поширило популярність месенджера серед студентів та учнів, які раніше мали групові чати у “ВКонтакте”.

Telegram в себе включає кращі риси всіх сучасних месенджерів, призначених для спілкування. Однак однією з головних особливостей програми Telegram є створення бота.

Бот - спеціальна програма, що виконує автоматично і / або за заданим розкладом будь-які дії через текстовий чат, призначений для людей. Взаємодія з ботом відбувається у персональному чаті, або за допомогою звертань, які починаються на символ “@”, якщо бот знаходиться у груповому чаті. Зазвичай бот позитивно позначається на працездатності групового чату, автоматизуючи однотипну роботу. Головне завдання ботів в чаті це допомога в модерації та адмініструванні, однак, функціонал ботів цим не обмежений, у зв'язку з цим боти можуть виконувати безліч поставлених завдань будь-якого характеру. Telegram надає зручний API на багатьох мовах програмування, як наслідок цього, сфера створення ботів досить швидко розвивається, на даний момент існує безліч розроблених ботів в різних сферах, таких як: новини, інтеграції з інших сервісів, фото та відео, музика.

Гарним прикладом автоматизації можна назвати величезну кількість ботів для прийому заявок на доставку їжі, замовлення столиків в ресторанах, розсилки реклами і багато іншого. Такі боти дозволяють збільшити прибуток компаній, так як бот може обробляти заявки з більшою швидкістю, ніж людина, і зменшити навантаження на робочий персонал.

Таким чином, метою даної роботи є створення Telegram бота з використанням Telegram Bot API для проведення тижня комп’ютерних дисциплін у технічному коледжі ТНТУ ім. Івана Пулюя.

# ЗАГАЛЬНИЙ РОЗДІЛ

## Аналітичний огляд існуючих рішень

По перше «бот» (скорочення від «робот») - програма, яка автоматично, по команді або розкладом виконує різні дії. Простіше кажучи, програма для здійснення рутинних операцій. Причому робить це через ті ж інтерфейси, що і звичайний користувач, як би імітуючи реального користувача.

Основна задача ботів – виконання одноманітної роботи і це не тільки економить сили і час людини, але вони роблять її на більш високих швидкостях.

У ботів краща реакція і точність дій в порівнянні з людиною - це знаходить застосування в комп'ютерних іграх, інтернет-аукціонах, рекламі, електронній біржової торгівлі і так далі. Боти застосовуються для імітації людської діяльності, зокрема, в чатах. Це так звані «чат-боти».

Чат-бота можна налаштувати на видачу адекватних відповідей на людській мові. Звичайно, за умови, що він розпізнає і підтримує такого роду команди.

Боти в Telegram - це різновид чат-ботів. За правилами всі їх імена повинні закінчуватися словом «bot». За своєю суттю - це ті ж призначені для користувача аккаунти, якими замість людей керують програми.

Вони допомагають виконувати різні дії: перекладати і коментувати, навчати і тестувати, шукати і знаходити, питати і відповідати, грати і розважати, транслювати, вбудовуватися в інші сервіси і платформи, взаємодіяти з датчиками і речами, підключеними до інтернету, і всі ці боти безкоштовні. Ботів Telegram можна «Додати в групу» (Add To Group), або «Поділитися» (Share). І це далеко не всі можливості, які представляє платформа для їх створення. В умілих руках боти можуть стати дуже потужним ресурсом для організації свого часу і автоматизації повторюваних дій.

Почати роботу з ботом просто: досить вибрати його з каталогу, перейти за посиланням або знайти по імені через пошук і вступити з ним в переписку. Найчастіше, для запуску бота вводиться команда /start або пропонується натиснути кнопку старту на віртуальній клавіатурі. Для роботи можуть використовуватися програмовані кнопки віртуальної клавіатури, за рахунок яких бот перетворюється в міні-додаток з інтуїтивно зрозумілим інтерфейсом.

Далі бот надішле вам інформацію про себе, інструкції, а також список команд або виведе доступні команди-кнопки на екран.

Боти можуть бути вкрай корисні у всіх сферах життя. За допомогою ботів можна пов'язувати об'єкти матеріального світу з користувачем. Особливо великі перспективи малюються в зв'язку з розвитком інтернету речей (Internet of Things).

З'єднання і датчиками дозволяє реалізувати концепцію «розумного будинку» навіть без великих фінансових витрат. Наприклад, українські розробники розробили ботів для популярних служб поштового зв’язку, Укрпошти та Нової пошти, вони дозволяють відслідковувати відправлення та отримувати сповіщення про місце знаходження відправлення, а російські розробники навчили домашні лічильники води спілкуватися з власником квартири через месенджер Telegram. Тобто людині досить запросити у бота дані по лічильникам і отримати телеметричні дані прямо до себе на смартфон.

А хтось використовує ботів Telegram для організації взаємодії людей, наприклад, дозволяючи замовнику бачити хід робіт і контролювати робочий процес.

Щоб зробити бота з персональними налаштуваннями, знадобляться спеціальні знання. Перш за все, знання англійської мови, щоб розібратися в описі можливостей ботів і інтерфейсі взаємодії з ботами (Bot API).

Так як бот для проведення тижня комп’ютерних дисциплін є досить спеціалізованим, і розробляється конкретно для певного закладу, аналогів у нього немає. Розробка спрямована на економію часу організаторів, звільняючи їх від рутинної роботи по реєстрації команд і проведенні “Техно квесту”, а також підрахунку набраних балів.

## Технічне завдання

### Найменування та область застосування

Найменування програми – «Розробка клієнт-серверного додатку для керування чат-ботом тижня комп’ютерних дисциплін».

Стисла назва - «WOCD bot».

Область застосування програми – реєстрація команд та проведення етапу для тижня комп’ютерних дисциплін.

### Призначення розробки

Підвищення ефективності праці роботи викладачів чи іншої керуючої особи, яка виконує реєстрацію та маршрутизацію команд являється експлуатаційним призначенням даної розробки.

Відповідно функціональне призначення – реєстрація та маршрутизація команд, керування переліком локацій, вивід конкурсної таблиці.

### Вимоги до програмного забезпечення

**Вимоги до функціональних характеристик**

Програма для керування чат-ботом тижня комп’ютерних дисциплін повинна забезпечити наступне функціонування:

* Реєстрація команд. Данні про команду повинні зберігатися в базі даних, а саме назва команди, прізвище та ім’я членів команди, та номер телефону капітана. Щоби не нагромаджувати велику кількість даних в одній таблиці, дані учасників будуть зберігатися в віддільній таблиці. Таким чином таблиця team\_list, це таблиця (див.табл. 1.1) з 4 стовпців і певної кількості рядків, яка відповідає кількості зареєстрованих груп, а таблиця members (див.табл. 1.2) – з 3 стовпців і певної кількості рядків, яка відповідає кількості зареєстрованих учасників.

Таблиця 1.1 – Зразок таблиці зареєстрованих команд для програми WOCD bot

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Назва | ID капітана | ID членів команди |
| 1 | Blue Tigers | 1 | 2,3,4,5,6,7,8 |
| 2 | r/programmerhumor | 9 | 10,11,12,13,14,15,16 |

Таблиця 1.2 – Зразок таблиці зареєстрованих учасників для програми WOCD bot

|  |  |  |
| --- | --- | --- |
| ID | Прізвище та Ім’я | Номер телефону |
| 1 | Покидко Олександр | +380966311636 |
| 2 | Романець Андрій | NULL |

* Маршрутизація команд. Маршрутизація зареєстрованих команд під час проведення Техно-квесту визначеними локаціями за допомогою текстових підказок, фраз. Данні про локації повинні зберігатися в базі даних. Таким чином, це таблиця (див. табл. 1.3) з 4 стовпців і певної кількості рядків, яка відповідає кількості локацій.

Таблиця 1.3 – Зразок таблиці локацій для програми WOCD bot

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Кабінет | Відповідь | Підказка |
| 1 | 104 | Коротке замикання | А чи давно вас струмом било ? |
| 2 | 208 | Вірус | Картинка з розширенням .exe |

**Часові характеристики**

Встановлено, що відповідь на запит користувача повинен приходити протягом 3 секунд.

**Вимоги до надійності**

Програмний комплекс повинен мати наступні вимоги до надійності:

* використання обробки виняткових ситуацій;
* контроль вводу даних на перебільшення розміру допустимого вводу;
* захист від несанкціонованого доступу до інформаційної бази;
* супровід дій користувача чіткими і зрозумілими повідомленнями.

**Умови експлуатації**

В склад технічних засобів повинні входити: монітор (діагоналлю не менше 15”), клавіатура, мишка, IBM-сумісний персональний комп’ютер або сервер із такими мінімальними характеристиками:

* процесор – Intel Pentium;
* оперативна пам’ять - 2 ГБ;
* обсяг дискової пам’яті – 2 ГБ.

Допускається робота сервера починаючи з Windows 7 та на Linux базованих операційних системах. Робота сервера вимагає встановлений інтерпретатор Python версії 3.7 або вище.

### Вимоги до програмної документації

По закінченню розробки програмного забезпечення потрібно підготувати таку документацію:

* інструкція інсталяції програми;
* загальні відомості про можливості програми;
* інструкція з експлуатації.

### Техніко-економічні показники

Розрахунок економічної ефективності і вартості розробки програмного продукту не проводиться.

Приблизне число використань розробленої програми в рік – 3-5 раз.

### Стадії та етапи розробки

Розробка клієнт-серверного додатку для керування чат-ботом тижня комп’ютерних дисциплін буде мати такі стадії:

* Аналіз вимог.

Метою аналізу є максимально повний опис поставленої задачі. Усі дані, що надходять потрібно проаналізувати і систематизувати, важливо також врахувати всі технічні обмеження, які можуть виникнути на стороні замовника. Підсумком даного етапу має стати створення докладної специфікації, що відповідає всім вимогам замовника. Також слід звернути увагу і на інші чинники, які можуть ускладнювати процес розробки.

* Проектування.

На етапі проектування необхідно визначитися з мовою програмування, фреймворками на яких буде розроблятися додаток, а також ознайомлення з BOT API платформи Telegram. Результатом етапу проектування є визначення функціональних відношень.

* Розробка і програмування.

Програмування передбачає чотири основні стадії:

1) розробка алгоритмів – фактично, створення логіки роботи програми;

2) написання вихідного коду;

3) компіляція – перетворення в машинний код;

4) тестування і налагодження – юніт-тестування.

### Порядок контролю та прийому

Прийом розробленого програмного забезпечення повинен відбуватися на об’єкті Замовника в терміни, які зазначені в індивідуальному завданні.

Для прийому роботи Виконавець повинен представити:

* діючу програму, яка повністю відповідає даному технічному завданню;
* вихідний програмний код, записаний разом із програмою на оптичний носій інформації.

Прийом програмного забезпечення повинен відбуватися перед комісією з двох чоловік (один з яких – Замовник) у такій послідовності:

* доповідь Виконавця про виконану роботу;
* демонстрація Виконавцем роботи програми;
* контрольні випробовування роботи програми;
* відповіді на запитання і зауваження комісії.

# РОЗРОБКА ТЕХНІЧНОГО ТА РОБОЧОГО ПРОЕКТУ

## Постановка задачі на розробку програмного забезпечення

Основна задача розробки – реалізувати реєстрацію команд, які братимуть участь в квесті, отримуючи повідомлення від платформи Telegram, використовуючи систему сповіщення про події – Webhook, де данні передаються у вигляді JSON (див. рис. 2.1) на попередньо задану адресу і порт 8443.

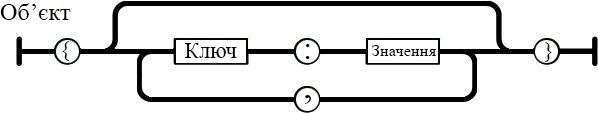


Рисунок 2.1 – Схематичне представлення об’єкта JSON

Так як кожне отримане повідомлення буде розглядатися, як нове і не пов’язане з попередніми, необхідно запам’ятовувати у індивідуальні сесії дії користувачів для поетапного опрацювання вхідних даних.

Для цього необхідно реалізувати:

1. Видалення попередньо встановленого Webhook-а;
2. Встановлення нового Webhook-а;
3. Підняття веб серверу, який буде обробляти запити.
4. Обробник вхідних даних.

По завершенню користувачем реєстрації, яку може проводити користувач тільки по персональному ключі, данні повинні появитися в базі даних і користувача повинні супроводжувати повідомлення протягом усіх етапів роботи з додатком, а при виникненні помилки повідомляти про неї користувача.

## Опис та обґрунтування вибору структури та методу організації  вхідних та вихідних даних

Під час розробки додатку не будуть використовуватись готові бібліотеки по роботі з BOT API, такі як python-telegram-bot, pyTelegramBotAPI та AIOGram #TODO добавити посилань, виправити коли буде більше. Для отримання даних буде використовуватись Webhook, який, на відмінну від Long pulling методу, дає кращий результат в швидкості отримання даних. Використання Webhook передбачає створення веб сервера, який буде отримувати данні про надходження повідомлення у вигляді HTTP POST запиту, який буде включати JSON об’єкт Update[5] #TODO добавити посилань, виправити коли буде більше.

Для реалізації веб сервера використовуватимуться:

* Flask – легкий фреймворк для створення вед додатків на мові програмування Python;
* Gevent – мережа бібліотек для створення швидкого веб серверу з підтримкою SSL сертифікатів.

Тепер сервер зможе приймати HTTP запити (див. рис. 2.2), які будуть надходити з сервера Telegram, кожен раз коли користувач надсилатиме повідомлення у чат з ботом.

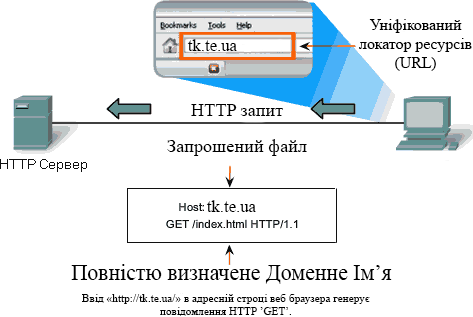


Рисунок 2.2 – HTTP протокол з використанням GET

Для відправлення повідомлень користувачу, необхідно користуватись методом sendMessage, обов’язковими атрибутами якого є: chat\_id – унікальний ідентифікатор чату між ботом і користувачем, його можна отримати з попередньо розглянутого JSON об’єкту, та сам текст повідомлення.

## Розробка алгоритму

### Зовнішнє проектування програми

У зв’язку з використанням API ми обмежені попередньо розробленими 66 методами та 96 об’єктами даних, цього достатньо щоби додаток.

Запити до Telegram Bot API повинні відправлятися через протокол HTTPS, використовувати кодування UTF-8 і відповідати наступному шаблоні: https://api.telegram.org/bot<Персональний токен бота>/<Назва методу>. Підтримуються POST та GET запити, а для передачі параметрів можна використовувати:

* URL строковий запит;
* application/x-www-form-urlencoded;
* application/json (виключення завантаження файлів);
* multipart/form-data (використовувати для завантаження файлів).

Для передачі файлів існує два основних правила:

* при наданні прямого URL посилання на файл, Telegram завантажить і надішле його. При цьому обмеження для розмірів файлів - 5 МБ для фотографій і 20 МБ для інших типів;
* надсилаючи файл з використанням multipart/form-data, обмеження для розмірів – 10 МБ для фото та 50 МБ для всіх інших типів файлів.

Основні методи які будуть використовуватись при розробці:

* deleteWebhook – цей метод видаляє попередньо встановлений Webhook. Не приймає параметрів. При успішному спрацюванні повертає True;
* setWebhook – цей метод дозволяє встановити посилання для отримання оновлень. Приймає наступні параметри (див. табл. 2.1). При успішному спрацюванні повертає True;

Таблиця 2.1 – Параметри методу setWebhook

|  |  |  |  |
| --- | --- | --- | --- |
| Параметр | Тип | Обов’язковий | Опис |
| url | String | Так | HTTPS посилання для отримання оновлень |
| certificate | InputFile | Ні | Сертифікат відкритого ключа для перевірки з кореневим сертифікатом |

* getWebhookInfo – цей метод дозволяє взнати поточний статус Webhook-а. Не приймає параметрів. При успішному спрацюванні повертає об’єкт WebhookInfo, якщо Webhook не встановлений, то поле url об’єкта WebhookInfo буде порожнім;
* sendMessage – метод дозволяє відсилати повідомлення користувачу. Приймає наступні параметри (див. табл. 2.2). Повертає об’єкт Message при успішному спрацюванні;

Таблиця 2.2 – Параметри методу sendMessage

|  |  |  |  |
| --- | --- | --- | --- |
| Параметр | Тип | Обов’язковий | Опис |
| chat\_id | Int або String | Так | Унікальний ідентифікатор необхідного чату або каналу |
| text | String | Так | Текст повідомлення, що надсилається |
| reply\_markup | ReplyKeyboardMarkup або  ReplyKeyboardRemove | Ні | Опція додаткового інтерфейсу  (див. рис. 2.3) |

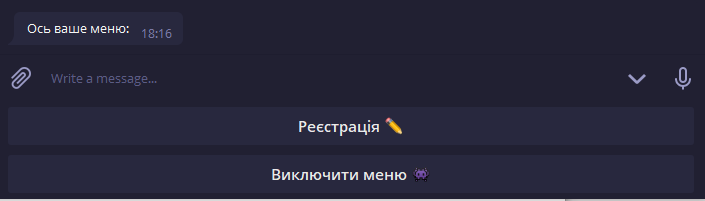


Рисунок 2.3 – Приклад використання ReplyKeyboardMarkup для створення спеціальної клавіатури.

* sendPhoto – метод дозволяє відсилати фотографію користувачу. Приймає наступні параметри (див. табл. 2.3). Повертає об’єкт Message при успішному спрацюванні;

Таблиця 2.3 – Параметри методу sendPhoto

|  |  |  |  |
| --- | --- | --- | --- |
| Параметр | Тип | Обов’язковий | Опис |
| chat\_id | Int або String | Так | Унікальний ідентифікатор необхідного чату або каналу |
| photo | InputFile або String | Так | Фото, що надсилається. Приймається file\_id якщо фотографія вже розміщена на серверах Telegram. Або загрузка нового фото використовуючи multipart/form-data |
| caption | String | Ні |  |
| reply\_markup | ReplyKeyboardMarkup або  ReplyKeyboardRemove | Ні | Опція додаткового інтерфейсу |

### Проектування логіки програми

## Визначення інформаційних зв’язків програмних компонентів

## Написання текстів програм

## Тестування та налагодження програм

# СПЕЦІАЛЬНИЙ РОЗДІЛ

## Інструкція з інсталяції програмного забезпечення

## Інструкція з використання тестових наборів

## Інструкція з експлуатації програмного комплексу

ВИСНОВКИ

ПЕРЕЛІК ПОСИЛАНЬ

1. Грицюк Ю.І. Об’єктно-орієнтоване програмування мовою С++ [Текст]: навчальний посібник / Ю.І. Грицюк, Т.Є. Рак. – Львів: Вид-во Львівського ДУ БЖД, 2011. – 404 с.
2. Build with Qt [Електронний ресурс]. – Електорон. дан. – Режим доступу: <https://www1.qt.io/built-with-qt/>. – Get started with Qt.
3. Qt Creator Manual [Електронний ресурс]. – Електорон. дан. – Режим доступу: <http://doc.qt.io/qtcreator/index.html> – Qt Documentation.
4. Asdasd
5. Telegram Bot API – Getting updates – Update [Електронний ресурс]. – Електорон. дан. – Режим доступу: <https://core.telegram.org/bots/api#update> – Telegram Bot API.

Додаток А  
Лістинг файлу «main.cpp»

Додаток Б  
Лістинг файлу «mainwindow.h»

Додаток В  
CD-диск із програмним продуктом