Министерство науки и образования Украины

Днепропетровский национальный университет железнодорожного транспорта им. академика В.Лазаряна

Кафедра КИТ

Отчет по практике

Выполнила:

студентка 941 группы

Шевченко И.В

Днепропетровск

2016

Оглавление

[1. ПРИЗНАЧЕННЯ, ПОСТАНОВКА ЗАДАЧІ ТА ОГЛЯД ПРОГРАМНИХ АНАЛОГІВ ТА ЛІТЕРАТУРИ 3](#_Toc450257809)

[1.1 Призначення та область застосування 3](#_Toc450257810)

[1.2 Постановка задачі 3](#_Toc450257811)

[1.3 Огляд програмних аналогів 3](#_Toc450257812)

[1.4 Огляд літератури 6](#_Toc450257813)

[ЛІТЕРАТУРА 13](#_Toc450257814)

1. ПРИЗНАЧЕННЯ, ПОСТАНОВКА ЗАДАЧІ ТА ОГЛЯД ПРОГРАМНИХ АНАЛОГІВ ТА ЛІТЕРАТУРИ
   1. Призначення та область застосування

Програмний комплекс, що розробляється, призначений для керування програмним забезпеченням на базі голосових команд людям з вадами зору, а також всі бажаючі скористатися програмним продуктом.

* 1. Постановка задачі

Необхідно розробити програмний продукт для керування програмним забезпеченням на базі голосових команд. Який дозволить самостійно користуватися комп’ютером та програмним забезпеченням на комп’ютері.

Програма має надавати можливість виклику інших программ, виконання елементарних функцій та праця с редакторами.

Програма повинна давати змогу зберігати нові команди користувача та надавати можливість повторного використання команд.

Результатом роботи програми має буди коректно виконанні команди користувача, можливість перегляду бази команд та внесення змін. Також на розпізнання голосової команди відводиться певна кількість хвилин, після якої біде зрозуміло розпізнана команда чи ні.

Програма має відповідати наступним вимогам: проста у використанні, зрозумілий та дружній інтерфейс. Основною вимогою є можливість розширення програми.

* 1. Огляд програмних аналогів

Задача системи аналізу звукової інформації допомогти інвалідам , та покращити в якомусь плані користування комп’ютером.

Метою роботи є скорочення часу користування користувача з вадами зору комп’ютера , облегшити взаємодію користувача з комп’ютером.

Перед початком роботи перш за все був виконаний огляд та аналіз аналогів. Для огляду програмних аналогів були розглянуті деякі  сайти, на яких пропонували скачати допоміжні програми для користування комп'ютером. В результаті огляду були розглянуті наступні аналоги програмного продукту, що розробляється: програма **«Горинич»,** «Typle 2.0», «Vocal Joystick». Нижче приведена інформація про найбільш поширені аналоги: голосового керування комп’ютером:

* + 1. **«Горинич»**

Дана програма перша російськомовна система автоматичного розпізнання мови. Програма для диктування і голосового управління комп’ютером російською мовою. Можливості даної програми заключаються в наступних функціях: введення тексту з допомогою голосу російською та англійською мовами, голосове управління периферійними обладнанням, голосове управління окремими функціями системи Microdoft Windows, голосове управління функціями текстових редакторів і прикладних програм дозволяє оформляти документи, дипломні роботи включаючи формули) для школярів, абітурієнтів, студентів. Один з головних плюсів програми це те, що система «Горинич» здійснює автоматичний контроль правопису: у текстах, які введені з її допомогою, виправляє орографічні помилки. Під час роботи користувач як і раніше може користуватися клавіатурою і мишкою. Чим частіше користувач диктуватиме, тим більше система «звикає» до голосу користувача.

1.1.2 «Typle 2.0»

Дана програма дозволяє керувати голосом деякими командами комп'ютера, з її допомогою можна запустити будь-яку програму або сторінку в інтернеті, використовуючи тільки голос користувача. Це дозволяє використовувати «Typle 2.0» людям, що говорять різними мовами, вживаючи «свої» голосові команди для запуску тієї чи іншої функції. Користуватися програмою просто, тим більш що інтерфейс серед декількох локалізацій має і російську. Для використання даної програми рекомендують використовувати хороший мікрофон – з малим рівнем шумів і широкими частотними характеристиками. Потім, після запуску Typle, потрібно записати команду, вибрати програму, до якої ця команда відноситься, і назвати її. Всі потрібні дії докладно описано в головному вікні.

Вигляд вікна програми зображено на рис. 1.1.

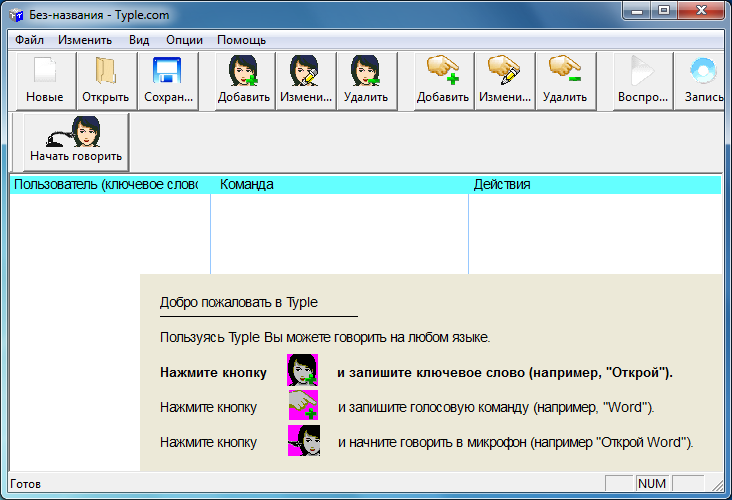


Рисунок 1.1 – Вигляд головного вікна програми «Typle 2.0»

* + 1. «Vocal Joystick».

«Vocal Joystick» - розробили вчені каліфорнійського університету, з її допомогою голосове управління комп’ютером стає набагато простішим. І хоча інтерфейс програми повністю англійський, окремі команди (а їх не мало) завдаються не з допомогою слів, а окремих слів.

Програма розпізнає до 100 звуків в секунду і миттєво відображає результати на екрані монітора. Вимовою різних голосових звуків можна задавати напрямок переміщення курсора в одному з восьми напрямків. Користувачі можуть безперешкодно переходити від одного до іншого голосового звуку, а гучністю голоса регулювати швидкість руху курсору. Звуками «ДО» і «Ч» можна імітувати натискання та звільнення кнопки миші.

Різні версії Vocal Joystick призначенні для серфінгу в інтернеті і відеоігор. Існує також версія для управління працездатною рукою, ця технологія соже бути використана для контролю клкетронного управління в інвалідному візку.

Вигляд вікна програми зображено на рис. 1.2.

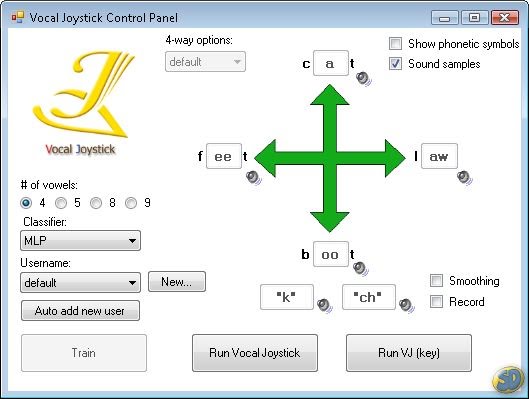


Рисунок 1.2 – Вигляд головного вікна програми «Vocal Joystick»

* 1. Огляд літератури

Розробка програмного продукту потребує ознайомлення з літературними джерелами за різними напрямками роботи, а саме: пристрої читання аудіо сигналів, різноманітні аспекти проектування і розробки програми (проектування БД, розробка ОО моделі, вибір и проектування структур даних, вибір форматів збереження мультимедійної інформації, проектування інтерфейсу користувача).

Загальна мета розробки програми аналізу звукової інформаціх допомогти людям з вадами зору користуватися комп’ютером. У використання користувачем комп’ютера також треба звертати увагу на багато факторів, наприклад, на життєвий досвід і особливості попереднього етапу використання комп’ютера, загальний фонд знань індивіда, традиції та особливості користування комп’ютера, специфіку голосу та вимови слів і т. п. Облік перерахованих факторів повинен сприяти оптимізації розрізнання голосових команд комп’ютером, зокрема. Одним з факторів, що визначає успішність розпізнання голосових команд є пристрої читання аудіо сигналів .

Впроваджувані пристрої читання аудіо сигналів мають допомогти у використанні зчитування та розпізнання аналогового чи цифрового звуку. Звук представляється в звуковій апаратурі у вигляді непереривного електричного сигналу, або у закодовонаму цифровому вигляді. Апаратура, в якій робочим сигналом є непереривним електричним сигналом, описує звукові коливання. Принцип дії мікрофона і динаміка основана на взаємодії перемінного тока, протікающий в електричній цепі котушці індуктивності, с магнітнім полем постійного магніта [7].

Обробка сигналів може вирішити багато задач, серед яких підвищення якості зображення, розпізнавання і синтез мови, стиснення даних для іх наступного зберігання і передачі і т.п. Синтез і розпізнавання голосу застосовуються для зв’язку між користувачами і машинами. Замість рук і глаз ви можете застосовувати рот та в уші. Для комп’ютерного синтезу мови застосовується два методи: цифровий запис і емуляція мовленнєвого такта. При цифровому записі людський голос оцифровується і зберігається в памяті компьютера в сжатому вигляді[8].

Аспекти проектування і розробки програми наведені в таких літературних джерелах. З проектування баз даних (БД):

Реляційні бази даних розроблені для швидкого збереження і отримання великих обсягів інформації. Кожен рядок даних у таблиці ідентифікується унікальним "ключем", який називається первинним ключем. Найчастіше, первинний ключ це автоматично збільшує (автоінкрементне) число (1,2,3,4 і т.д). Дані в різних таблицях можуть бути пов'язані разом при використанні ключів. Значення первинного ключа однієї таблиці можуть бути додані в рядки (записи) іншої таблиці, тим самим, пов'язуючи ці записи разом.

Використовуючи структуровану мова запитів (SQL), дані з різних таблиць, які пов'язані ключем, можуть бути обрані за один раз[9].

У проекті бази даних, яка створена з урахуванням правил реляційної моделі даних, кожен шматочок інформації, наприклад, ім'я користувача, зберігається тільки в одному місці. Це дозволяє усунути необхідність роботи з даними в декількох місцях. Дублювання даних називається надмірністю даних і цього слід уникати в хорошому проекті бази даних.

Вказівки для правильного проектування реляційних баз даних зібрані в 5 груп, які називаються нормальними формами.

Нормальні форми - це рекомендації з проектування баз даних. Ви не зобов'язані дотримуватися всіх п'яти нормальних форм при проектуванні баз даних. Тим не менш, рекомендується нормалізувати базу даних в деякій мірі тому, що цей процес має ряд істотних переваг з точки зору ефективності та зручності поводження з вашою базою даних:

* У нормалізованої структурі бази даних ви можете робити складні вибірки даних відносно простими SQL-запитами;
* Цілісність даних. Нормалізована база даних дозволяє надійно зберігати дані.
* Нормалізація запобігає появі надмірності збережених даних. Дані завжди зберігаються тільки в одному місці, що робить легким процес вставки, оновлення та видалення даних. Є виняток з цього правила. Ключі, самі по собі, зберігаються в кількох місцях тому, що вони копіюються як зовнішні ключі в інші таблиці.
* Масштабованість - це можливість системи справлятися з майбутнім зростанням. Для бази даних це означає, що вона повинна бути здатна працювати швидко, коли число користувачів і обсяги даних зростають. Масштабованість - це дуже важлива характеристика будь-якої моделі бази даних і для РСУБД[10].

Наведені вище книги з розробки БД будуть використані при проектуванні БД для команд голосового розпізнання.

З розробки об’єктно-орієнтованої моделі:

Об'є́ктно-орієнто́ване програмування (ООП) — одна з [парадигм](https://uk.wikipedia.org/wiki/%D0%9F%D0%B0%D1%80%D0%B0%D0%B4%D0%B8%D0%B3%D0%BC%D0%B8_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F) [програмування](https://uk.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F), яка розглядає програму як множину «об'єктів», що взаємодіють між собою. Основу ООП складають три основні концепції: [інкапсуляція](https://uk.wikipedia.org/wiki/%D0%86%D0%BD%D0%BA%D0%B0%D0%BF%D1%81%D1%83%D0%BB%D1%8F%D1%86%D1%96%D1%8F), [успадкування](https://uk.wikipedia.org/wiki/%D0%A3%D1%81%D0%BF%D0%B0%D0%B4%D0%BA%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F_(%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F)) та [поліморфізм](https://uk.wikipedia.org/wiki/%D0%9F%D0%BE%D0%BB%D1%96%D0%BC%D0%BE%D1%80%D1%84%D1%96%D0%B7%D0%BC_(%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F)). Одною з переваг ООП є краща [модульність](https://uk.wikipedia.org/wiki/%D0%9C%D0%BE%D0%B4%D1%83%D0%BB%D1%8C_(%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F)) програмного забезпечення[11]. На відміну від традиційних поглядів, коли програму розглядали як набір [підпрограм](https://uk.wikipedia.org/wiki/%D0%9F%D1%96%D0%B4%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%B0), або як перелік [інструкцій](https://uk.wikipedia.org/wiki/%D0%86%D0%BD%D1%81%D1%82%D1%80%D1%83%D0%BA%D1%86%D1%96%D1%8F) комп'ютеру, ООП програми можна вважати сукупністю об'єктів. Відповідно до парадигми об'єктно-орієнтованого програмування, кожний об'єкт здатний отримувати [повідомлення](https://uk.wikipedia.org/wiki/%D0%9E%D0%B1%D0%BC%D1%96%D0%BD_%D0%BF%D0%BE%D0%B2%D1%96%D0%B4%D0%BE%D0%BC%D0%BB%D0%B5%D0%BD%D0%BD%D1%8F%D0%BC%D0%B8), обробляти дані, та надсилати повідомлення іншим об'єктам. Кожен об'єкт — своєрідний незалежний автомат з окремим призначенням та відповідальністю.

На думку [Алана Кея](https://uk.wikipedia.org/wiki/%D0%90%D0%BB%D0%B0%D0%BD_%D0%9A%D0%B5%D0%B9), розробника мови [Smalltalk](https://uk.wikipedia.org/wiki/Smalltalk" \o "Smalltalk), якого вважають одним з «батьків-засновників» ООП, об'єктно-орієнтований підхід полягає в наступному наборі основних принципів:

* Все є об'єктами.
* Всі дії та розрахунки виконуються шляхом взаємодії (обміну даними) між об'єктами, при якій один об'єкт потребує, щоб інший об'єкт виконав деяку дію. Об'єкти взаємодіють, надсилаючи і отримуючи повідомлення. Повідомлення — це запит на виконання дії, доповнений набором аргументів, які можуть знадобитися при виконанні дії.
* Кожен об'єкт має незалежну пам'ять, яка складається з інших об'єктів.
* Кожен об'єкт є представником (екземпляром, примірником) класу, який виражає загальні властивості об'єктів.
* У класі задається поведінка (функціональність) об'єкта. Таким чином усі об'єкти, які є екземплярами одного класу, можуть виконувати одні й ті ж самі дії.
* Класи організовані у єдину деревоподібну структуру з загальним корінням, яка називається ієрархією [успадкування](https://uk.wikipedia.org/wiki/%D0%A3%D1%81%D0%BF%D0%B0%D0%B4%D0%BA%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F_(%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F)). Пам'ять та поведінка, зв'язані з екземплярами деякого класу, автоматично доступні будь-якому класу, розташованому нижче в ієрархічному дереві.

Наведені вище книги будуть використовуватися при проектуванні програми.

Тестування програмного забезпечення ([англ.](https://uk.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D1%96%D0%B9%D1%81%D1%8C%D0%BA%D0%B0_%D0%BC%D0%BE%D0%B2%D0%B0) Software Testing) — це процес технічного дослідження, призначений для виявлення інформації про якість продукту відносно контексту, в якому він має використовуватись. Техніка тестування також включає як процес пошуку [помилок](https://uk.wikipedia.org/wiki/%D0%91%D0%B0%D2%91) або інших дефектів, так і випробування програмних складових з метою оцінки. Може оцінюватись[12]:

* відповідність вимогам, якими керувалися проектувальники та розробники;
* правильна відповідь для усіх можливих вхідних даних
* виконання функцій за прийнятний час
* практичність
* сумісність з програмним забезпеченням та операційними системами
* відповідність задачам замовника.

Оскільки число можливих тестів навіть для нескладних програмних компонент практично нескінченне, тому стратегія тестування полягає в тому, щоб провести всі можливі тести з урахуванням наявного часу та ресурсів. Як результат програмне забезпечення (ПЗ) тестується стандартним виконанням програми з метою виявлення [баґів](https://uk.wikipedia.org/wiki/%D0%91%D0%B0%D2%91" \o "Баґ) (помилок або інших дефектів).

Інтерфейс (від англ. Interface — поверхня розділу, перегородка) — сукупність засобів, методів і правил взаємодії (управління, контролю і т. д.) між елементами системи. Цей термін використовують у багатьох галузях науки й техніки. Його значення належить до будь-якої сполуки взаємочинних сутностей (як природничих, так апаратних і людино-машинних). Під інтерфейсом розуміють не тільки пристрої, але й правила ([протокол](https://uk.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D1%82%D0%BE%D0%BA%D0%BE%D0%BB_%D0%BF%D0%B5%D1%80%D0%B5%D0%B4%D0%B0%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F_%D0%B4%D0%B0%D0%BD%D0%B8%D1%85)) взаємодії цих пристроїв.

Інтерфейси є основою взаємодії в сьогочасних інформаційних системах. Якщо інтерфейс якого-небудь об'єкту (персонального комп'ютера, програми, функції) не змінюється (стабільний, стандартизований), це дає можливість модифікувати сам об'єкт, не перероблюючи його принципи взаємодії з іншими об'єктами[12].

В обчислювальний системі взаємодія може здійснюватися на користувацькому, програмному й апаратному рівнях. Відповідно до цього інтерфейси можуть існувати як:

* Спосіб взаємодії фізичних пристроїв
* Спосіб взаємодії віртуальних пристроїв
* Спосіб взаємодії людина-машина

У методичних рекомендаціях описуються цілі роботи, структура і зміст кожного варіанту, особливості завдань, рекомендації з проведення роботи, перевірці та оцінці результатів виконання окремих завдань і роботи в цілому.

Waveform Audio File Format (WAVE, WAV, від англ. Waveform - «у формі хвилі») - формат файлу-контейнера для зберігання запису оцифрованого аудіо потоку. Цей контейнер як правило використовується для зберігання не стисненого звуку в імпульсно-кодовій модуляції. Однак контейнер не накладає жодних обмежень на використовуваний алгоритм кодування

Вище розглянуто два формати збереження звуку. Найбільш вдалим форматом при використанні можна вважати amr, так як цей формат при стиснені не змінює свого звучання при відтворенні серед інших. А ще він має відносно малий об’єм при збереженні. Тому я звернула на цей формат свою увагу при розробці програми.

У форматі MP3 використовується алгоритм стиснення з втратами, розроблений для істотного зменшення розміру даних, необхідних для відтворення запису і забезпечення якості відтворення звуку дуже близького до оригінального (на думку більшості слухачів), хоча аудіофіли говорять про відчутне розходженні. При створенні MP3 із середнім бітрейтом 128 кбіт / с в результаті виходить файл, розмір якого приблизно дорівнює 1/11 від оригінального файлу з CD-Audio[13].

При виконанні аналізу літературних джерел були виявлені та вивчені новітні методики проектування та розробки програмних продуктів, зокрема таких програмних засобів які є фізично розподіленими, тобто генерування та обробка інформації виконується на різних фізичних засобах. Також була доведена необхідність реалізації налаштування інтерфейсу користувача.

# ЛІТЕРАТУРА

1. Радзишевский А.Ю. Основі аналогового и цифрового звука.- Русский язык.2006.- 288 с.
2. Cмит С. Цифровая обработка сигналов.Практичное руководство для инженеров и научных работников.-Русский язык.2012 – 718 с.
3. Керівництво з проектування баз даних [Електронний ресурс] // Режим доступа до сайту:

http://habrahabr.ru/post/193136/

1. Роб П. Системы баз данных проектирование, реализация и управление 5-е издание / Питер Роб, Корлос Корнел. – БХВ-Петербург, 2004. – 1040 с. Розенталь Д.С. Синтаксис и пунктуація. Правила и упражнения / Розенталь Денис. – Оникс, 2010. – 96 с.
2. [Попова М.](http://www.ruslania.com/entity-1/context-577/author-30205.html) Итоговое тестирование в формате экзамена / М. Попова. – Академия развития, 2010.- 128 с.
3. Торрес Р. Практическое руководство по проектированию пользовательского интерфейса / Торрес Роберт. – Вильямс, 2002. – 400 с.
4. Формат MP3 [Електронный ресурс] // Режим доступа до журн.: <http://ru.wikipedia.org/wiki/MP3>.
5. Формат AMR [Електронный ресурс] // Режим доступа до журн.: http://ru.wikipedia.org/wiki/AMR\_(сжатие\_звука).