**ДНІпровський національний університет**

**імені Олеся Гончара**

**ФАКУЛЬТЕТ ПРИКЛАДНОЇ МАТЕМАТИКИ**

**КАФЕДРА КОМП’ЮТЕРНИХ ТЕХНОЛОГІЙ**

**ЗВІТ**

**З НАВЧАЛЬНОЇ ПРАКТИКИ:**

**КОМП’ЮТЕРНО – ТЕХНОЛОГІЧНОЇ**

Освітньо-професійна програма

Комп’ютерне моделювання та технології програмування

Спеціальність 113 Прикладна математика

Галузь знань 11 Математика і статистика

Рівень вищої освіти перший (бакалаврський)

Студента 2 курсу групи ПА-21-2

Москаленка А.М.

(прізвище та ініціали)

Керівник ас. Лисиця Н.М.

(посада, вчене звання, науковий ступінь, прізвище та ініціал)

Кількість балів

Національна шкала

Члени комісії:

Зайцев В.Г.

(підпис) (прізвище та ініціали)

Зайцева Т,А.

(підпис) (прізвище та ініціали)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Лисиця Н.М.

(підпис) (прізвище та ініціали)

м. Дніпро, 2023 р.

ЗМІСТ

[ВСТУП 3](#_Toc138665260)

[ПОСТАНОВКА ЗАДАЧІ 5](#_Toc138665261)

[1 АЛГОРИТМ РОЗВ’ЯЗАННЯ ЗАДАЧІ 6](#_Toc138665262)

[1.1 Створення основи телеграм бота 6](#_Toc138665263)

[1.2 Додавання функції конвертації голосового повідомлення у текст (STT) 6](#_Toc138665264)

[1.3 Додавання функції конвертації тексту в голосове повідомлення (TTS) 7](#_Toc138665265)

[1.4 Тестування телеграм бота 8](#_Toc138665266)

[2 ОПИС ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ 9](#_Toc138665267)

[2.1 Використані технології для створення чат бота 9](#_Toc138665268)

[2.2 Інструкція користувача 9](#_Toc138665269)

[3 АНАЛІЗ ОТРИМАНИХ РЕЗУЛЬТАТІВ 10](#_Toc138665270)

[1.1 Аналіз логів 10](#_Toc138665271)

[1.2 Скріншоти роботи програми 10](#_Toc138665272)

[4 ЗВІТ О СПІВПРАЦІ КОМАНДИ 12](#_Toc138665273)

[ВИСНОВКИ 13](#_Toc138665274)

[СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ 14](#_Toc138665275)

# ВСТУП

Голосові повідомлення є зручним способом спілкування, особливо в ситуаціях, коли написання повідомлення може бути неефективним або незручним. Наприклад, під час водіння, занять спортом або коли ви не можете відволікатися на писання повідомлення.

Але не завжди є можливість прослуховувати голосові повідомлення. Наприклад, коли їдеш у громадському транспорті або сидиш у наповненому людьми офісі, лікарні, зупинці тощо.

Існує і обернена ситуація, коли не має можливості прочитати текстове повідомлення. Наприклад з такою ситуацією можуть стикнутися люди з обмеженими можливостями, які мають проблеми з читанням тексту.

Тому конвертація голосових повідомлень у текст та навпаки, конвертація тексту у голосові повідомлення є актуальним інструментом, який полегшує комунікацію, покращує доступність та забезпечує зручність для користувачів у різних ситуаціях.

Виходячи з актуальності проблеми, ми вирішили створити телеграм бота для конвертації голосових повідомлень у текст та конвертацією тексту у голосові повідомлення.

ЗВІТ З НАВЧАЛЬНОЇ КОМП’ЮТЕРНО – ТЕХНОЛОГІЧНОЇ ПРАКТИКИ складається з таких частин:

* ВСТУП, який обґрунтовує актуальність роботи; визначає цілі проведення наукового дослідження; галузь дослідження; методи дослідження або розрахунків;
* ПОСТАНОВКА ЗАДАЧІ;
* ОСНОВНА ЧАСТИНА (перелічити назви розділів за змістом)
* ВИСНОВКИ;
* СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ;
* ДОДАТКИ.

# ПОСТАНОВКА ЗАДАЧІ

1. Створити телеграм бота, який конвертує голосові повідомлення у текст і навпаки, текст у голосові повідомлення на мові програмування *Python*.
2. Бот має виконувати такі функції:

* перетворювати текст у голосове повідомлення;
* перетворювати голосові повідомлення у текст.

Бот повинен сприймати текст і голосові повідомлення та конвертувати їх російською мовою.

1. Виконати тести з налагодження чат бота та оптимізувати його роботу.

# АЛГОРИТМ РОЗВ’ЯЗАННЯ ЗАДАЧІ

## Створення основи телеграм бота

Для створення телеграм бота потрібно:

1. Написати сам скрипт на мові *Python*;
2. За допомогою телеграм бота *BotFather* створити індивідуальний телеграм токен;
3. Створити його обгортку, тобто наділити його можливостю виводити, наприклад, вітальне повідомлення.

## Додавання функції конвертації голосового повідомлення у текст (STT)

1. Завантажити директорію *vosk* з [даного веб сервісу](https://alphacephei.com/vosk/models);
2. Додаємо файл *ffmpeg*, який являє собою набір вільних бібліотек, які дозволяють конвертувати та передавати цифрові аудіо і відео записи і різних форматах. Тобто за допомогою цього файлу ми конвертуємо наш голосовий файл у текст;
3. Створюємо клас STT, та з’єднуємо з основною обгорткою бота;
4. Заповнюємо клас:

* оголошуємо default\_init (підтягуємо *vosk* і *ffmpeg*, та вказуємо їх шлях у проекті);
* функція \_\_init\_\_ (налаштовуємо моделі *vosk* та *ffmpeg*);
* функція \_check\_model (перевіряємо наявність моделей *vosk* і *ffmpeg*);
* функція audio\_to\_text (розпізнаємо аудіо та конвертуємо в текст), за допомогою process, передає ім’я вхідного файлу, частоту виборки, кількість каналів, кодек для перекодування. Якщо всі вище перелічені функції бот пройшов успішно (тобто true), тоді він розпочинає читати дані частинами і розпізнавати через модель. Повертає розпізнаний текст у вигляді str (рядка).

## Додавання функції конвертації тексту в голосове повідомлення (TTS)

1. Завантажити модуль *silero* з [гіт репозиторія](https://github.com/snakers4/silero-models);
2. Додаємо файл *ffmpeg*, який являє собою набір вільних бібліотек, які дозволяють конвертувати та передавати цифрові аудіо і відео записи в різних форматах;
3. Створюємо клас TTS, та з’єднуємо з основною обгорткою бота;
4. Заповнюємо клас:

* оголошуємо default\_init (підтягуємо *silero* і *ffmpeg*, та вказуємо їх шлях у проекті, виконавця та швидкість читання тексту);
* функція \_\_init\_\_ (налаштовуємо моделі *silero* та *ffmpeg*);
* функція \_check\_model (перевіряємо наявність моделей *silero* і *ffmpeg*);
* функція wav\_to\_ogg (конвертуємо аудіо в .ogg формат);
* функція ogg\_to\_wav (конвертуємо аудіо в .wav формат);
* функція \_get\_wav (конвертуємо текст у .wav файл);
* функція \_get\_ogg (конвертуємо текст у .ogg файл);
* функція \_nums\_to\_text (конвертує числа у літери);
* функція \_merge\_audio (об’єднує декілька файлів в один без перекодування, файли повинні бути однакового формату);
* функція rename\_file (перейменує in\_filename в output\_filename);
* функція text\_to\_ogg (конвертує текст в файл .ogg);
* функція text\_to\_wav (конвертує текст в файл .wav);
* функція wav\_to\_ogg\_bytes (конвертує аудіо у .ogg формат без зберігання даних у диск).

## Тестування телеграм бота

Попередження: чат бот працює лише з російською мовою.

1. Запускаємо телеграм бота, запускаючи файл *bot.py*;
2. Переходимо у телеграм бот за посиланням <https://t.me/Autobot_audio_bot>;
3. Запускаємо бота за допомогою команди “/start”;
4. Якщо бот виводить привітальне повідомлення, то обгортка телеграм бота працює.
5. Далі перевіряємо саму конвертацію повідомлень:
   * надсилаючи текст до бота, повинно прийти голосове повідомлення із переказаним текстом;
   * надсилаючи голосове повідомлення, повинен прийти від чат бота розпізнаний текст.
6. Якщо під час тривалого тестування чат бота не було виявлено багів (наприклад, бот не надсилає повідомлення у відповідь, або надсилає декілька разів поспіль, або надсилає некоректно конвертоване повідомлення тощо), то бот працює успішно.

# ОПИС ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

## Використані технології для створення чат бота

Цей проект написаний на мові програмування *Python* 3.9.7.

Ми використовуємо такі технології:

* телеграм бот *BotFather* – використовується для створення нових облікових записів ботів і керування існуючими ботами.
* файл *ffmpeg* – являє собою набір вільних бібліотек, які дозволяють конвертувати та передавати цифрові аудіо і відео записи і різних форматах;
* біблотека *vosk* – це бібліотека по розпізнаванню мови;
* модуль *silero –* це модуль, який займається озвучуванням тексту.

## Інструкція користувача

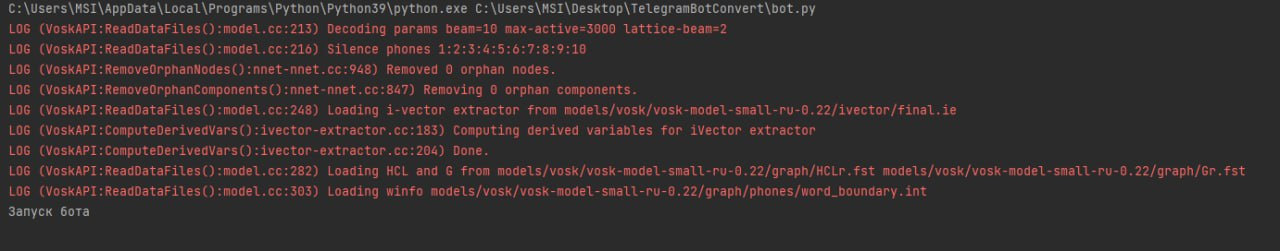
1. Запускаємо телеграм бота, запускаючи файл *bot.py*;
2. Переходимо у телеграм бот за посиланням <https://t.me/Autobot_audio_bot>;
3. Запускаємо бота за допомогою команди «/start»;
4. Якщо бот виводить привітальне повідомлення, то чат бот готовий до конвертації повідомлень;
5. Далі можемо виконати саму конвертацію повідомлень:
   * надсилаючи текст до бота, прийде голосове повідомлення із переказаним текстом (конвертація може зайняти деякий час);
   * надсилаючи голосове повідомлення, прийде від чат бота розпізнаний текст.

# АНАЛІЗ ОТРИМАНИХ РЕЗУЛЬТАТІВ

У результаті ми отримали бота, який конвертує текст у голосові повідомлення та обернену конвертацію, голосові повідомлення у текст.

Ми зіткнулися із проблемами некоректного перетворення голосового повідомлення у текст українською мовою, через відсутність добре оптимізованої бібліотеки *vosk*. Саме з цих причин нам довелося використовувати російську бібліотеку *vosk*.

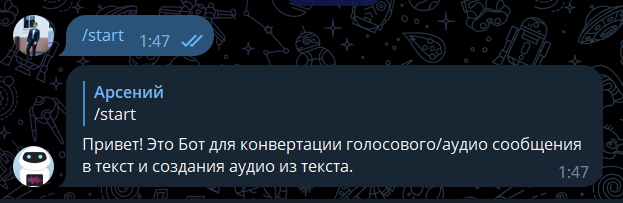
## Аналіз логів



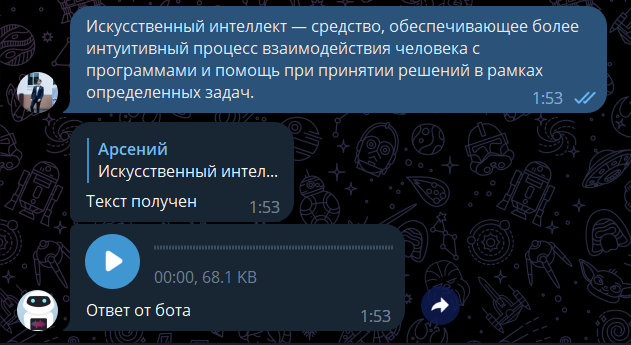
Малюнок 1 - логи програми

Аналізуючи логи після запуску програми, можна зробити висновок, що бот успішно запустився та працює.

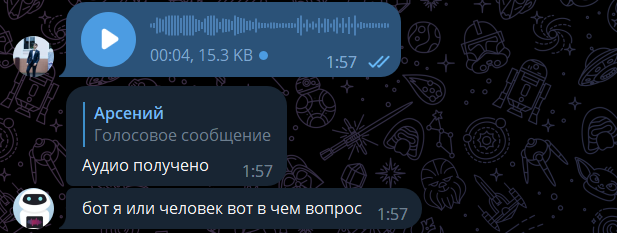
## Скріншоти роботи програми



Малюнок 2 - запуск бота



Малюнок 3 - конвертація текста у голосове повідомлення



Малюнок 4 - конвертація голосового повідомлення у текст

# ЗВІТ О СПІВПРАЦІ КОМАНДИ

Наша команда складається з трьох учасників, що мають досвід у розробці телеграм ботів за допомогою мови програмування *Python*. У команді були розподілені такі ролі:

* Розробники: Нечепоренко Олександр, Шаталович Димитрій;
* Тестувальник: Москаленко Арсеній.

Основною метою співпраці було розробка телеграм бота, який конвертує текст у голосові повідомлення та навпаки, для зручності комунікації між людьми.

Під час співпраці ми використовували систему управління версіями *Git*, яка дозволяла нам гнучко відстежувати, змінювати, керувати проектом. Ми проводили регулярні онлайн-зустрічі у корпоративній платформі *Microsoft Teams* на яких обговорювали проблеми та завдання, спільно планували наступні етапи роботи. Додатково для спілкування використовували такі комунікаційні інструменти, як *Telegram* та *Discord*.

У висновку можна зазначити, що ми успішно реалізували телеграм бот, який конвертує текст у голосові повідомлення та голосові повідомлення у текст. Співпраця нашої команди була дуже ефективною. Завдяки добрій комунікації в команді, ми успішно досягли всіх поставлених цілей та реалізували наш проект в рамках обмеженого терміну.

# ВИСНОВКИ

За результатом виконаної роботи можна зробити такі висновки:

1. Були розглянуті методи створення телеграм чат бота на мові *Python* з використанням додаткових бібліотек ті модулів, таких як *ffmpeg*, *vosk* і silero;
2. У результаті було створено обгортку телеграм бота та додано до нього функції конвертації тексту у голосові повідомлення та конвертації голосових у текст;
3. Для обробки запитів користувача були створені відповідні функції;
4. Проаналізовані логи, за якими можна судити успішний запуск чат бота;
5. За допомогою постійних тестів чат бота, було виконано його оптимізацію, шляхом усунення дублікатів повідомлень.

# СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. [Марк Лутц «Изучаем Python», том-1, 5-ое издание - 2019](https://rozetka.com.ua/158460002/p158460002/)
2. <https://core.telegram.org/bots/api>
3. <https://pypi.org/project/aiogram/>
4. <https://alphacephei.com/vosk/>
5. <https://pypi.org/project/torch/>
6. <https://pypi.org/project/python-dotenv/>
7. <https://pypi.org/project/num2words/>
8. <https://ffmpeg.org>