

**ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ**

Кафедра інженерії програмного забезпечення

Пояснювальна записка

до бакалаврської кваліфікаційної роботи
на ступінь вищої освіти бакалавр

на тему: «Розробка програмного забезпечення Question-Answering системи на основі навчально-методичних матеріалів дисципліни "Програмування C++" мовою Python»

Виконав: студент 4 курсу, групи ПД– 41

спеціальності

121 Інженерія програмного забезпечення

(шифр і назва спеціальності)

Агашков А.Ю.

(прізвище та ініціали)

Керівник Золотухіна О.А.

(прізвище та ініціали)

Рецензент _____

(прізвище та ініціали)

Нормоконтроль _____

(прізвище та ініціали)

ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ
Навчально-науковий інститут інформаційних технологій

Кафедра Інженерії програмного забезпечення

Ступінь вищої освіти - «Бакалавр»

Спеціальність - 121 Інженерія програмного забезпечення

ЗАТВЕРДЖУЮ

Завідувач кафедри

Інженерії програмного
забезпечення

_____ О.В.Негоденко

« ____ » _____ 2023 року

ЗАВДАННЯ
НА БАКАЛАВРСЬКУ РОБОТУ СТУДЕНТУ

Агашков Андрій Юрійович

(прізвище, ім'я, по батькові)

1. Тема роботи: «Розробка програмного забезпечення Question-Answering системи на основі навчально-методичних матеріалів дисципліни "Програмування C++" мовою Python»

Керівник роботи _____ к.т.н., доц., Золотухіна О.А.

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом вищого навчального закладу від "24" лютого 2023 року №26.

2. Строк подання студентом роботи 01.06.2023

3. Вихідні дані до роботи:

- 3.1.Офіційна документація бібліотек Python.

- 3.2.Матеріали PyCharm Community.

- 3.3.Опис методів та алгоритмів для обробки текстів природньою мовою.

- 3.4.Матеріали дисципліни «Програмування C++».

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити):

- 4.1.Аналіз та огляд існуючих додатків та інструментів для реалізації системи.

- 4.2.Розробка алгоритмів та структури додатку з системою question-answering.

- 4.3.Програмна реалізація та тестування додатку.

5. Перелік графічного матеріалу
 - 5.1 Мета, об'єкт, предмет дослідження
 - 5.2 Задачі дипломної роботи
 - 5.3 Аналіз аналогів
 - 5.4 Вимоги до програмного забезпечення
 - 5.5 Програмні засоби реалізації
 - 5.6 Діаграма варіантів використання
 - 5.7 Діаграма класів
 - 5.8 Екранні форми
 - 5.9 Апробація результатів дослідження
 - 5.10 Висновки

6. Дата видачі завдання 24.02.2023

КАЛЕНДАРНИЙ ПЛАН

| № з/п | Назва етапів бакалаврської роботи | Строк виконання етапів роботи | Примітка |
|-------|---|-------------------------------|----------|
| 1 | Підбір науково-технічної літератури | 25.02.2023-2.03.2023 | Виконано |
| 2 | Аналіз та дослідження існуючих аналогів | 3.03.2023-18.03.2023 | Виконано |
| 3 | Вимоги до додатку | 19.03.2023-22.03.2023 | Виконано |
| 4 | Аналіз засобів реалізації | 23.03.2023-29.03.2023 | Виконано |
| 5 | Проектування застосунку | 30.03.2023-15.04.2023 | Виконано |
| 6 | Розробка програмного забезпечення | 15.04.2023-16.05.2023 | Виконано |
| 7 | Оформлення роботи, висновки | 17.05.2023-19.05.2023 | Виконано |
| 8 | Розробка демонстраційних матеріалів | 19.05.2023-21.05.2023 | Виконано |
| 9 | Попередній захист | 22.05.2023-26.05.2023 | Виконано |
| 10 | Здача роботи | 01.06.2023 | |

Студент _____

(підпис)

А.Ю. Агашков

(прізвище та ініціали)

Керівник роботи _____

(підпис)

О.А.Золотухіна

(прізвище та ініціали)

РЕФЕРАТ

Текстова частина бакалаврської роботи: 63с., 18 рис., 2 дод., 2 табл., 16 джерел.

РУСCHARM, PYTHON, ОБРОБКА ЗАПИТІВ, «ПРОГРАМУВАННЯ C++», QUESTION-ANSWERING СИСТЕМА, NLP, PYQT6.

Об'єкт дослідження – процес надання відповідей на питання студентів, що вивчають дисципліну «Програмування C++».

Предмет дослідження – програмне забезпечення question-answering системи на основі навчально-методичних матеріалів дисципліни «Програмування C++».

Мета роботи - спрощення процесу надання відповідей на питання студентів, що вивчають дисципліну «Програмування C++», за рахунок використання програмного забезпечення з технологією question-answering, реалізованого мовою Python.

Методи дослідження – методи обробки та аналізу запитів користувачів, методи побудови пре-тренованої моделі для видачі відповідей, що задовольняють вимогам запиту, патерни та евристики проектування інтерфейсів.

В роботі було реалізовано програмне забезпечення, що використовує технологію question-answering. Для цього було використано мову програмування Python, його бібліотеку PyQt6 для створення функціонального інтерфейсу, Stanza для обробки текстової інформації. Також було використано sqlite3, яка дозволяє створювати та використовувати бази даних і їх збережені значення.

Галузь використання – програму може використовувати студент-першокурсник, чи будь-яка інша людина, яка хоче вивчати дисципліну «Програмування C++».

ЗМІСТ

| | |
|---|----|
| ВСТУП..... | 9 |
| 1 АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ | 11 |
| 1.1 Особливості та проблеми вивчення дисципліни «Програмування C++» | 11 |
| 1.2 Огляд методів обробки тексту | 12 |
| 1.3 Особливості систем question-answering..... | 14 |
| 1.4 Аналіз аналогів | 17 |
| 1.4.1 W3School | 18 |
| 1.4.2 Sololearn | 20 |
| 1.4.3 ChatGPT..... | 21 |
| 1.5 Визначення вимог до програмного забезпечення..... | 24 |
| 2 ПРОЕКТУВАННЯ ТА РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ | 26 |
| 2.1 Огляд програмних засобів реалізації..... | 26 |
| 2.1.1 Мова програмування Python | 26 |
| 2.1.2 Середовище розробки PyCharm..... | 27 |
| 2.1.3 PyQt6 | 28 |
| 2.1.4 SQLite3 | 29 |
| 2.1.5 Stanza | 31 |
| 2.2 Моделювання вимог до програмного забезпечення..... | 32 |
| 2.3 Розробка алгоритму обробки текстового запиту | 34 |
| 2.4 Розробка класів..... | 35 |
| 2.5 Розробка моделі з відповідями | 37 |
| 2.6 Проектування та розробка інтерфейсу | 38 |
| 3 ОПИС ФУНКЦІОНУВАННЯ ТА ТЕСТУВАННЯ СИСТЕМИ..... | 40 |
| 3.1 Порядок роботи з системою..... | 40 |
| 3.2 Тестування програмного забезпечення | 43 |
| ВИСНОВКИ | 45 |
| СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ | 46 |
| ДОДАТОК А ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ..... | 48 |
| ДОДАТОК Б ЛІСТИНГИ ПРОГРАМНИХ МОДУЛІВ | 54 |

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

GUI - Graphic User Interface

NLTK - Natural Language Toolkit

QA - question-answering система

БД – база даних

ООП – об'єктно-орієнтоване програмування

ПЗ – програмне забезпечення

ВСТУП

Найбільш популярними спеціальностями, вступники яких мають найвищі середні бали з екзаменів ЗНО серед галузі знань під номером 12 «Інформаційні технології» - це спеціальності 121 «Інженерія програмного забезпечення» та 126 «Інформаційні системи та технології». За даними інформаційного ресурсу vstup.info, лише на ці дві спеціальності в сумі абітурієнти подали більше 1000 заявок в 2022-му році, що є великим числом в порівнянні з іншими напрямками[1]. Переважна більшість студентів, що йдуть навчатись в університети на наведені вище спеціальності, на першому курсі першого семестру стикаються з дисципліною «Програмування C++». Мову програмування C++ не можна назвати простою.

Мова програмування C++ виділяється наступними показниками:

- низькорівневість. Програміст має повний контроль над програмою – наприклад, регулювання використання пам'яті, що при невмілому застосуванні може призвести до припинення працездатності всього комп'ютера.
- Статична типізація. Кожній змінній чи об'єкту привласнюється свій тип даних.
- Технології. C++ підтримує різні типи та стилі програмування – узагальнене програмування, ООП, процедурне тощо.

Ці показники дозволяють C++ займати передові позиції в способах вивчення нейромереж (бібліотеки Deep learning), а також значна частина ігрової індустрії, адже дана мова програмування дозволяє гнучкими методами оптимізувати великі проекти під швидкодію, не втрачаючи в якості.

Наведені вище пункти показують частину проблем, з якими стикається пересічний студент-першокурсник. Попри наявну навчально-матеріальну інформаційну базу, процес вивчення дисципліни важко назвати простим. Враховуючи відсутність попереднього досвіду в програмуванні, навіть проста різниця у використанні чисельних типів даних, дається складним шляхом.

Саме тому було вирішено створити програмне забезпечення, яке студент може запустити паралельно з виконанням лабораторних робіт, перегляду інформаційно-навчальних матеріалів, чи просто у свій вільний час. Це ПЗ відповідатиме на поставлені користувачем запитання, що значно полегшить шлях до розуміння теми, принципу роботи, особливостей мови програмування.

Об'єкт дослідження – процес надання відповідей на питання студентів, що вивчають дисципліну «Програмування C++».

Предмет дослідження – програмне забезпечення question-answering системи на основі навчально-методичних матеріалів дисципліни «Програмування C++».

Мета роботи - спрощення процесу надання відповідей на питання студентів, що вивчають дисципліну «Програмування C++», за рахунок використання програмного забезпечення з технологією question-answering, реалізованого мовою Python.

Досягнення поставленої мети досягається за рахунок вирішення наступних задач:

1. Провести аналіз підходів до обробки користувацьких запитів в Question-Answering системах, в тому числі, запитів українською мовою.
2. Провести огляд застосунків, які можна використовувати для вивчення дисципліни «Програмування C++» в форматі «питання-відповідь».
3. Провести аналіз навчально-методичних матеріалів дисципліни «Програмування C++» та визначити особливості формування відповідей.
4. Сформувати функціональні та нефункціональні вимоги до програмного забезпечення.
5. Спроекувати та реалізувати програмне забезпечення Question-Answering системи, яка дозволяє формувати відповіді на питання студентів на основі навчально-методичних матеріалів дисципліни «Програмування C++».
6. Провести тестування програмного забезпечення.

Практичне значення додатку полягає в полегшенні вивчення матеріалів дисципліни «Програмування C++».

1 АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ

1.1 Особливості та проблеми вивчення дисципліни «Програмування С++»

С++ - мова програмування [2] середнього рівня, яка має достатній набір інструментів для користування програмістам різного рівня – від новачків чи студентів, які тільки почали свій шлях в ІТ сфері, до професіоналів, що розробляють та впроваджують серйозні й складні програмні рішення та продукти. Дана мова програмування відповідає стандартам ISO (International Standards Organization) та ANSI (American National Standards Institute)[2]. Включає в себе різні концепції та парадигми програмування: ООП (об'єктно-орієнтоване програмування), процедурне програмування та узагальнене. Дані можливості надають програмісту широкий інструментарій абстракцій, і в той же час вимагає високої точності у використаних функціях, типах даних, об'єктах тощо. С++ також надає додаткові можливості, такі як перевантаження операторів, винятки, шаблони та інші, що робить його потужним і гнучким інструментом для розробки програм з використанням ООП підходу.

Студенти-першокурсники спеціальностей 121 «Інженерія Програмного Забезпечення», 113 «Прикладна математика», 126 «Інформаційні системи та технології» являють собою переважний пласт початкових програмістів, що на перших курсах проходять дисципліну «Програмування С++». Інші спеціальності, такі як 122 «Комп'ютерні науки», 124 «Системний аналіз», та інші математичні й ІТ напрямки, можуть містити в собі цю дисципліну. Дана мова програмування не є простою для вступу, особливо студентам та пересічним громадянам, які проявляють бажання у вивченні С++.

Наступні пункти є найбільш проблемними в людей, що вивчають С++ в університетах:

- Складний синтаксис: мова програмування С++ має багатий синтаксис і велику кількість можливостей, що може створювати певні труднощі для

початківців. Синтаксичні правила, вказівники, посилання, шаблони, стандартні бібліотеки - це всі аспекти, з якими студентам потрібно ознайомитись і розібратись.

- Концепції об'єктно-орієнтованого програмування: C++ є мовою, яка тісно пов'язана з підтримкою об'єктно-орієнтованого програмування. Для студентів, які ще не мають досвіду з цим підходом, може бути складно зрозуміти концепції класів, об'єктів, успадкування, поліморфізму та інших аспектів ООП.

- Брак практичного досвіду: Навчання програмуванню C++ вимагає багато практичної роботи та вирішення задач. Брак практичного досвіду може зробити вивчення більш складним, оскільки студентам може бути важко застосувати теоретичні концепції до практичних завдань.

- Велика кількість матеріалу: C++ є потужною мовою програмування з великою кількістю функцій, бібліотек та концепцій. Для студентів може бути важко освоїти всі аспекти мови, особливо за короткий період навчання.

1.2 Огляд методів обробки тексту

З розвитком діджиталізації, експоненціально зросли вимоги до систем роботи з текстовою інформацією. В залежності від потреб конкретних установ, можливе використання як звичайних текстових редакторів, так і складних текстових процесорів.

Текстовий редактор – це програмне забезпечення, що призначене для введення, виведення та збереження тільки текстової інформації (літери, цифри, знаки пунктуації тощо). Прикладом може виступати текстовий редактор Блокнот, що встановлений за замовчуванням.

Текстовий процесор – це програмне забезпечення, яке дає змогу працювати з текстами та графічними елементами (картинки, відео-елементи, графіки, діаграми). Такі системи мають лише візуальну складову роботи з текстами – переведення з малого регістру у великий, стиль та розмір шрифтів, їх кольори, тіні. До більш функціональних опцій можна віднести можливість створювати

зміст, для навігації по всьому документу, залишати примітки та коментарі, робити гіперпосилання, аналіз тексту на наявність лексичних чи граматичних помилок тощо.

Вищенаведений набір доступних варіантів обробки тексту є достатнім для підготування рефератів чи дипломів, книжок, звітів. На даному програмному забезпеченні майже відсутня взаємодія між комп'ютерною частиною – окрім зберігання та внесення змін, зроблених користувачем. Жоден текстовий процесор чи текстовий редактор не має функціоналу, який виходить за межі візуального представлення інформації.

Для вирішення цієї проблеми популярним рішенням є спеціальні алгоритми обробки тексту [8]. Вони базуються на більш глобальному понятті, як NLP(Natural Language Processing). За визначенням, це область штучного інтелекту, що допомагає розуміти, маніпулювати та інтерпретувати людську мову комп'ютерам, які за означенням не мають подібної можливості.

Нижче наведені основні методи обробки тексту:

- Токенизація. Процес розбивання речень на окремі частинки, які розділені пробілом. В результаті, виходить список слів та всіх розділових знаків, або виключно слів, відкинувши всі пунктуаційні та інтонаційні знаки.

- Лематизація та стемінг. Різні процеси, які проводять задля досягнення одного й того ж результату: звуження варіацій одного й того ж смислового слова. Під стемінгом розуміють зведення слів з різними закінченнями до основи (кореня) слова. Наприклад, «станції» та «станціями» будуть розумітись як одне й те саме «станці». Лематизація, в свою чергу, нівелює розбіжності форми слова різних відмінків, роблячи всіх їх зрозумілим як початкова форма (для іменників – називний відмінок, для дієслів – інфінітив).

- Усунення стоп-слів: Стоп-слова - це загальні слова, які не несуть значення в текстовому аналізі, такі як "і", "в", "на" тощо. Усунення стоп-слів з тексту допомагає сконцентруватись на більш суттєвих термінах та покращує якість та швидкість аналізу.

- Векторизація[14]. Дозволяє програмному засобу оцінити важливість того чи іншого слова в реченні, наборі речень. Різні алгоритми по-різному витягують одиницю оцінювання – тільки слова, чи разом з фразеологізмами. Найпопулярнішими є TextRank(важливість текстів серед інших), TF-IDF(Term Frequency – Inverse Document Frequency, оцінює значущість слів в реченні, будуючи матрицю оцінок), Rake(Rapid Automatic Keywords Extraction, може витягати слова та фрази з конкретного файлу чи документа, ігноруючи інші).

- Розпізнавання іменованих сутностей[15] (NER, Named Entity Recognition). Розпізнавання іменованих сутностей виявляє та класифікує різні типи сутностей у тексті, такі як особи, місця, організації, дати, числа тощо. Це корисно для виділення важливих та значущих елементів у тексті.

- Аналіз настроїв. Використовується при аналізі настроїв людей, наприклад, під час опитування споживачів, статистичному зборі задоволення життям тощо. Проста версія включає в себе поділення на 3 оцінки – негативне, нейтральне та позитивне. Складніші версії використовують бали для подальшої оцінки. Найпоширенішим способом являється «наївний Баєс»(Naïve Bayes).

- Text Summarization[16]. Може використовуватись як вилучення тексту, при цьому створюючи невеликий список найголовніших речень, так і створення нового (абстракція) змісту, який передає суть оригінального тексту. Зазвичай, використовують при обширних дослідженнях. На прикладах, використовуються наступні алгоритми: LexRan, TextRank.

- Класифікація та кластеризація тексту: Класифікація тексту включає призначення категорії або мітки до тексту на основі його змісту. Кластеризація тексту групує подібні тексти разом без попереднього знання про категорії.

1.3 Особливості систем question-answering

Перші системи question-answering [3] беруть свій початок з 1961-го року, і за своєю суттю були першими спробами втілити NLP у вузькоспеціалізовані області. Найстаріший відомий приклад – BASEBALL, який видавав відповіді на нескладні

запитання, що пов'язані з бейсболом: назва команди, рахунок, в яких місяців був матч тощо.

Система question-answering – це програмне забезпечення або модель машинного навчання, яке призначене для автоматичної відповіді на питання, задані людьми, на основі наданої колекції текстової інформації. Відповідь може братись як з одного документа, який зберігається на фізичному носії, так і з групи документів, інформаційних ресурсів, що зберігаються в онлайн режимі. Зазвичай, QA системи використовуються для забезпечення швидкого та точного доступу до інформації, що міститься у великих корпусах тексту, таких як електронні книги, статті, веб-сторінки або навчальні матеріали.

Існує безліч різних question-answering систем, але глобально, всіх їх можна поділити на 2 групи: вузькоспеціалізовані та загальні. Перші містять дані про конкретні галузі чи сфери діяльності (медицина, англійська мова, математика). Гарно підходять під побудову побідної системи, яка, наприклад, автоматизує відповіді на найбільш часті запитання, або ж слугує помічником для вирішення конкретних задач. Другі ж дають відповіді більш широкої спрямованості. Щоб розробити подібну систему, потрібно витратити величезну кількість часу на зіставлення моделі з питаннями та відповідями, тренування розпізнавання відповідей залежно від контекстів тощо.

Спеціалісти, які досліджують та вивчають процеси області системи question-answering вивели наступний список проблемних питань:

- типи питань. Питання можуть відрізнятися одне від одного, в результаті чого й пошук відповідей на них буде відрізнятися.
- Основа матеріальної бази знань. Знаходження повної бази даних матеріалів, яка б охоплювала список можливих питань.
- Виділення відповідей. Яким би складним питання не було, воно має отримати ту відповідь, яку покладено всередину його змісту. Складність полягає в контексті, методу пошуку, виду матеріалів тощо.
- Відповіді в реальному часі. Найкраще залізо може мати проблеми з ручним рядковим пошуком відповіді серед петабайтів інформації.

- Обробка питань. Однакова інформація може оброблятися різними способами. Звідси впливає запит на створення ефективних та швидких методів обробки та розуміння запитань.

- Підтримка декількох мов. Найчастіше розробляють системи під англійську мову, а потім вже під всі інші, в залежності від соціумів, які потребують систему надання відповідей.

- Інтерактивність. Налаштування системи діалогу, де користувач може уточнити контекст та важливі деталі до поставленого запиту, аби question-answering система краще розуміла, яку саме відповідь потрібно виводити.

Основна задача QA системи полягає у розумінні питання користувача, пошуку відповідей у текстових джерелах та відповіді на ці питання. Для досягнення цієї мети, QA системи використовують різні підходи та техніки, включаючи:

- Обробка природними мовами (NLP): QA системи використовують техніки NLP для розуміння та аналізу питань, розбиття тексту на токени, лематизації, стемінгу, векторизації тощо.

- Пошук інформації: QA системи використовують методи пошуку, такі як індексування тексту, векторні моделі та інші алгоритми, для знаходження потенційних відповідей у великих текстових колекціях.

- Машинне навчання: QA системи можуть використовувати моделі машинного навчання, зокрема нейромережі, для виявлення та класифікації питань, аналізу контексту та генерації відповідей.

- Аналіз семантики: QA системи можуть використовувати аналіз семантики для розуміння смислу запитання та текстових джерел, що допомагає знайти більш точні відповіді.

- Інтеграція з базами даних: QA системи можуть бути інтегровані з базами даних, що містять конкретні факти або інформацію, яка може бути використана для надання відповідей на запитання.

QA системи можуть мати різні рівні складності та функціональності, від простих систем, що відповідають на конкретні запитання, до складних систем, які

здатні аналізувати текст, розуміти контекст та генерувати докладні відповіді.

1.4 Аналіз аналогів

Знаходити відповіді на власні запитання – незалежно від тематики та специфіки теми, можна наступними шляхами:

- ручний перегляд відповідної літератури в офлайн/онлайн режимах;
- спитати в людей;
- використовувати програмне забезпечення/онлайн ресурс із технологією question-answering.

Перегляд літератури може бути двох виглядів. Перший – це пошук у фізичних бібліотеках друковані варіанти книжок. Найстарший спосіб самостійного пошуку інформації, який в дуже рідкісних випадках може бути єдиним способом знаходження відповіді. Але здебільшого, дуже вимогливий до часу, який потребується приділити цьому процесу. Другий – пошук онлайн-ресурсів. Це не обов'язково буде офіційна література, а скоріше, набір наукових та публіцистичних статей, матеріально-навчальні бази, семінари, форуми тощо. Спосіб більш дієвий та функціональний, адже браузер, в якому пишеться запит, підбирає та надає інформацію що тематично та контекстно пов'язана. Проте, щоб такий спосіб знаходити інформацію не займав надто багато часу, користувачу потрібно розуміти той самий контекст та тематику запитання. А запропоновані відповіді можуть збити з пантелику, даючи різні відповіді на одне й те саме запитання, викладення інформації, яка не стосується затребуваної користувачем тощо.

Запитувати людей – спосіб дізнання відповідей, який був створений разом зі створенням перших способів комунікації. На сьогоднішній день досі є легітимним способом дізнаватись нове, бути в курсі останніх новин, можливості комфортного проведення часу в соціумі. Такий варіант майже унеможливорює процес вивчення нової інформації в її первинному варіанті. Яким би досконалим та валідним експертом не був доповідач, він лиш ретранслює всю ту інформацію, якою володіє. Інтонація, жестикуляція, міміка, тембр та глибина голосу – лише

частина фактів, які можуть як спотворити викладену інформацію, так і спотворити її прийняття слухачем.

На сьогоднішній день, використовувати технологію question-answering є найбільш продуктивним способом знайти конкретну інформацію на конкретне запитання. Враховуючи тему дипломної роботи, було проаналізовано ресурси, які надають відповіді, маючи систему question-answering.

1.4.1 W3School

W3School[13] – один з найбільш популярних ресурсів по вивченню мов програмування та не тільки. Він містить в собі обширний список дисциплін: HTML, CSS, Python, C++, Canvas, SQL, Git, Excel тощо.

Для вивчення програмування мовою C++, W3School має наступний список функціоналу:

- Список тем, який містить в собі короткий опис по внутрішньому функціоналу. Також містить приклади програмних кодів та можливість самому запустити код і подивитись результат компіляції.
- Можливість поставити запитання у відповідному рядку. Для отримання чіткої відповіді потрібно дуже конкретно та максимально коротко сформулювати запитання.

Як видно по рисункам 1.1-1.3, представлена пошукова система робить неможливим подивитись формулювання поточного запитання. Як і неможливо подивитись історію власних пошуків, і тим більше відповіді на них.

Також, ресурс не підтримує українську локалізацію (автоматичний переклад в браузері існує, але місцями дуже і дуже неточний, через що спотворюється пояснення). І враховуючи той факт, що W3School являється веб-сайтом, його неможливо використовувати без мережі інтернет. У військових умовах України, цей аспект не можна недооцінювати.

| | |
|--------------------|--------------------------|
| C++ Tutorial | C++ Functions |
| C++ HOME | C++ Functions |
| C++ Intro | C++ Function Parameters |
| C++ Get Started | C++ Function Overloading |
| C++ Syntax | C++ Recursion |
| C++ Output | C++ Classes |
| C++ Comments | C++ OOP |
| C++ Variables | C++ Classes/Objects |
| C++ User Input | C++ Class Methods |
| C++ Data Types | C++ Constructors |
| C++ Operators | C++ Access Specifiers |
| C++ Strings | C++ Encapsulation |
| C++ Math | C++ Inheritance |
| C++ Booleans | C++ Polymorphism |
| C++ Conditions | C++ Files |
| C++ Switch | C++ Exceptions |
| C++ While Loop | C++ How To |
| C++ For Loop | Add Two Numbers |
| C++ Break/Continue | C++ Examples |
| C++ Arrays | C++ Examples |
| C++ Structures | C++ Compiler |
| C++ References | |
| C++ Pointers | |

Рисунок 1.1 – Список доступних тем на W3School

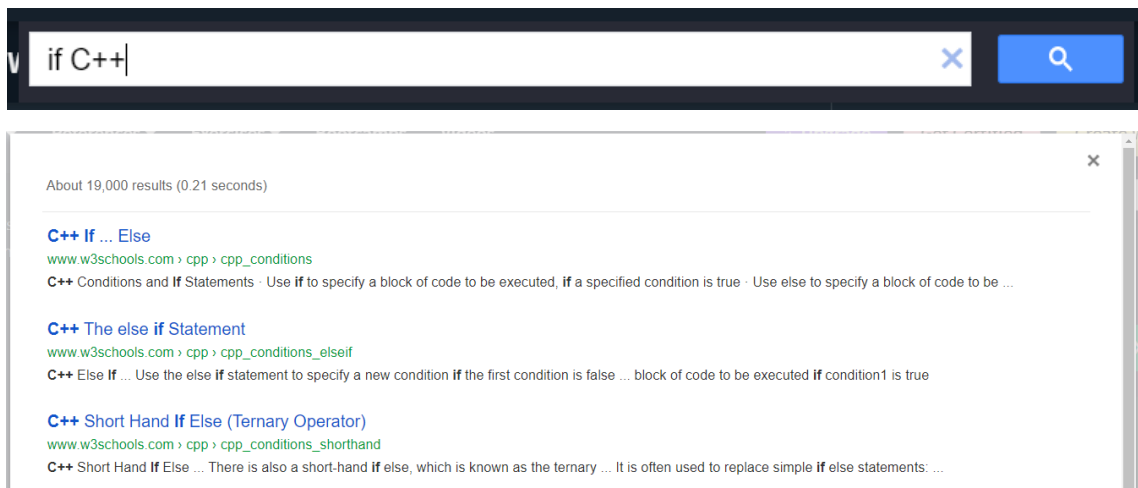


Рисунок 1.2 – Приклад відповіді на коротке запитання у W3School

No Results

🔍 Search for як зрозуміти if C++ on Google

Рисунок 1.3 – Приклад відповіді на більш довге запитання у W3School з урахуванням попереднього контексту

1.4.2 Sololearn

Sololearn – мобільний додаток, що дещо схожий за функціоналом до попереднього аналогу, але має меншу матеріальну базу: Python, Java, C++, JavaScript та інші. Містить структуру, яка схожим чином, послідовно викладає матеріал користувачу. На прикладі з дисципліною «Програмування C++» можемо побачити структуру та зовнішній вигляд розділів (рис.1.4).

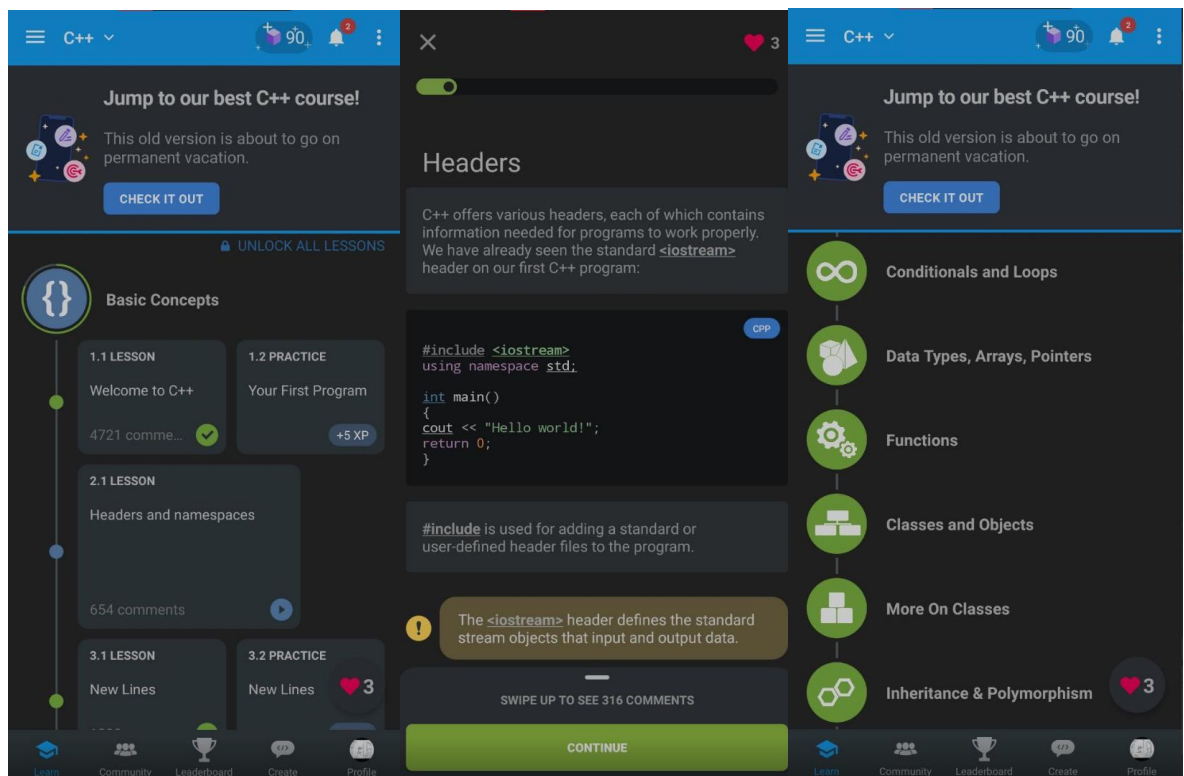


Рисунок 1.4 – Приклади структури вивчення дисципліни «Програмування C++» в Sololearn

З позитивних моментів виділено:

- наявність інтерактивного викладення матеріалу: короткий опис, програмний код та перевірка знань, що були отримані одразу після прочитання міні-розділу;
- наявність прогресу, що може мотивувати повертатись в застосунок та вчитись надалі;
- наявність спільноти, в якій можна дивитись популярні топіки, вирішення певних задач, та власноруч писати пости із запитаннями, враженнями тощо;

- можливість отримати сертифікат по закінченню курсу;
- частковий доступ без Інтернету.

З негативних сторін додатку виділено:

- обмежений функціонал в додатку - застосунок постійно нагадує про можливість придбати преміум (який коштує 110грн за місяць);
- відсутність можливості шукати якісь терміни, ставити запитання самому застосунку. в результаті, без знання розділу, в якому може знаходитись відповідь, відшукати відповідь майже неможливо;
- скорочена матеріально-навчальна база. чимало можливостей продемонстровано або обрізано, або взагалі не згадано під час проходження курсу;
- відсутня українська локалізація, навіть без можливості автоматичного перекладу;
- пошук в спільноті по тематикам працює дуже неналежним чином, відображені відповіді можуть взагалі не містити того, що запитувалось.

1.4.3 ChatGPT

ChatGPT [4] – чат-бот зі штучним інтелектом, який дуже різко увірвався на ринок, та заповнив собою широчезну нішу question-answering засобів. Був створений лабораторією OpenAI та запущений в листопаді 2022-го року. Буквально за півроку набув світової популярності, піднявши рівень взаємодії з question-answering системою на новий рівень. Рисунок 1.5, 1.6 демонструють зовнішній вигляд інтерфейсу користувача, що споживає послуги від OpenAI.

Головні переваги застосунку виділено наступними пунктами:

- Діапазон можливих запитань. ChatGPT здатен формулювати код, писати тексти для промови чи рефератів, аналізувати програмний код та виправляти помилки всередині нього, та багато яких інших способів роботи з текстом.
- Можливість зберігати чати, та створювати нові. Дозволяє зберігати гілку питань-відповідей, залишаючи лише ті запити, які стосуються заголовку/обраної тематики.



Рисунок 1.5 – Головна сторінка лабораторії OpenAI

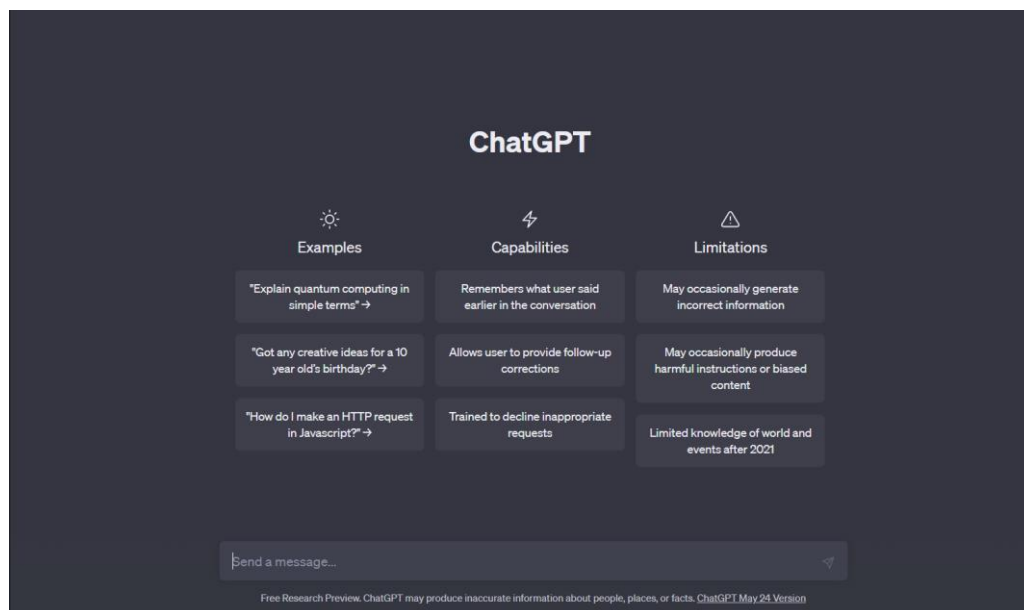


Рисунок 1.6 – Головна сторінка ChatGPT

- Швидкість відповіді. Чат-бот генерує відповідь, відштовхуючись від власної бази даних, на очах.
- Обробка запитань природньою мовою. Чат-бот дуже добре розуміє запитання та можливі його доповнення, які мав на увазі користувач.
- Генерація простих відповідей на складні питання. Чат-бот здатен формулювати відповіді таким чином, щоб набір використаних слів та термінів був зниженим та зрозумілим ширшому прошарку населення. Наприклад, із запитом «Поясни, чому кипить вода з точки зору фізики» та «Поясни, чому кипить вода семирічній дитині» від відповідь з різним ступенем важкості.
- Чат-бот вчиться на своїх же помилках.

- Можливість вести зв'язний діалог. Чат-бот розуміє контекст попередніх запитів, та може відповідати в цьому контексті.

- Мультиплатформеність. Можливість користуватись як в браузері з мережею Інтернет, так і скачати на десктоп/ноутбук/телефон/планшет програмне забезпечення для автономного користування поза мережею.

До списку недоліків було віднесено наступні пункти:

- Інколи він відмовляється розуміти питання, постійно прокручуючи неправильну відповідь. Або уникає вирішення проблеми та вирішує лише частково поставлену задачу. Часто це проявляється з програмуванням, коли користувач просить комплексне рішення, а не підказку на декілька рядків.

- Неможливо видалити окремі відповіді. Деякі результати чат-бота можуть руйнувати логіку і порядок відповідей, або не стосуватись вказаної теми чи напрямку.

- Відсутня можливість гарантовано отримати суміжні відповіді.

Враховуючи всі вищенаведені характеристики, позитивні та негативні сторони кожного інфокомунікаційного ресурсу, було сформовано таблицю вимог, згідно ключових потреб для програмного забезпечення, яке використовуватиме технологію question-answering по навчально-методичним матеріалам «Програмування C++».

Таблиця 1.1 – Результати характеристик додатків що допомагають з вивченням дисципліни «Програмування C++»

| Найменування | W3School | SoloLearn | ChatGPT | Розроблений додаток |
|--|----------|--------------|--|---------------------|
| Назви засобів | | | | |
| Україномовна локалізація | - | - | + | + |
| Платформа | Браузер | Android, iOS | Браузер, Windows 10, iOS, Mac, Android | Windows 10 |
| Матеріал відповідно навчальної програми дисципліни «Програмування C++» | + | + | + | + |

Продовження таблиці 1.1 – Результати характеристик додатків що допомагають з вивченням дисципліни «Програмування C++»

| Найменування | W3School | SoloLearn | ChatGPT | Розроблений додаток |
|--|-----------|----------------------|--------------------------------------|---------------------|
| Назви засобів | | | | |
| Можливість дивитись поставлені раніше запитання | - | - | + | + |
| Можливість поставити запитання українською мовою | - | - | + | + |
| Доступ без мережі інтернет | Відсутній | Обмежений функціонал | Тільки при попередньому завантаженні | Присутній |
| Можливість отримання відповідей, що суміжні з поставленим запитанням | + | - | - | + |

1.5 Визначення вимог до програмного забезпечення

Згідно аналізу предметної області, аналізу аналогів та виписування їх переваг та недоліків в сфері програмного забезпечення з question-answering системами, було сформовано наступні функціональні вимоги до застосунку, що розробляється в супроводі з дипломною роботою:

- можливість введення користувачем тексту запитання з клавіатури;
- наявність засобів попередньої обробки тексту запиту;
- формування текстової відповіді, релевантної ключовим словам, наявним в користувацькому запиті, на основі навчально-методичних матеріалів дисципліни «Програмування C++»;
- можливість відображення у відповіді на питання програмного коду за темою;
- можливість пошуку відповіді за суміжними темами;
- можливість відображення переліку питань, які користувач ставив раніше,

відповіді на них, можливість видалення цих запитань.

До нефункціональних вимог було винесено наступні пункти:

- локалізація українською мовою;
- робота без підключення до мережі Інтернет;
- підтримка платформи Windows 10;
- модель, яка використовується для видачі відповідей, має зберігатись у базі даних на пристрої;
- перелік попередніх питань та відповідей має зберігатись у базі даних на носіїві програмного забезпечення.

2 ПРОЕКТУВАННЯ ТА РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

2.1 Огляд програмних засобів реалізації

2.1.1 Мова програмування Python

Python – мова програмування зі суворою динамічною типізацією даних. Станом на 2023-й рік ця мова програмування набула світового масштабу завдяки своїм чисельним перевагам в простоті використання та ефективності роботи в різних галузях та напрямках.

Нижче наведені основні причини, чому Python[6] такий популярний, чим ця мова програмування виділяється серед інших та чому вона продовжує розвиватись:

- Легкість використання: Python має простий та зрозумілий синтаксис, що робить його легким для вивчення та розуміння. Це дозволяє розробникам швидко писати, тестувати і налагоджувати код, що є важливим у процесі розробки question-answering систем.

- Багата екосистема: Python має чудовий та широкий вибір бібліотек з інструментарієм, які сприяють розробці question-answering систем. Наприклад, бібліотеки, такі як NLTK, SpaCy та Stanza, надають потужні інструменти для обробки природної мови, включаючи токенізацію, частину мови, семантичний аналіз та інше. Крім того, існують спеціалізовані бібліотеки, такі як Transformers, які надають готові моделі для роботи з питаннями та відповідями.

- Машинне навчання та глибоке навчання: Python є однією з найпопулярніших мов програмування для розробки моделей машинного навчання та глибокого навчання. Це дозволяє розробникам використовувати потужні алгоритми та нейромережі для побудови question-answering моделей. Бібліотеки, такі як TensorFlow, Keras і PyTorch, надають зручні інтерфейси для реалізації цих моделей.

- Велика спільнота та ресурси: Python має активну та велику спільноту розробників, що робить його легким для знаходження підтримки, ресурсів та

відповідей на питання. Існують форуми, документація, статті та навчальні матеріали, які можна використовувати для покращення навичок у розробці question-answering систем.

- Переносимість та інтеграція: Python працює на багатьох платформах, включаючи Windows, macOS та різні дистрибутиви Linux. Крім того, він добре інтегрується з іншими мовами програмування та інструментами, що дозволяє легко поєднувати його з існуючими системами та інфраструктурою.

Підсумовуючи, Python представляє собою потужний та зручний інструмент для розробки question-answering систем завдяки своїй простоті в кодуванні, великій спільноті програмістів, підтримці машинного навчання та глибокого навчання та мультиплатформеності.

2.1.2 Середовище розробки PyCharm

PyCharm Community[5] є популярним та потужним інтегрованим середовищем розробки (IDE) для програмування на мові Python. Нижче наведені основні причини використовувати PyCharm Community для розробки на Python:

- Повна функціональність: PyCharm Community надає багато функцій і інструментів, що допомагають під час розробки на Python. Він має розумний редактор коду з автодоповненням, підсвічуванням синтаксису та інструментами форматування коду, дозволяє компілювати код частинами, що значно підвищує ефективність виправлення помилок. Дана IDE також підтримує налагодження, систему контролю версій, аналіз коду, візуальне створення і редагування інтерфейсу користувача та чимало інших корисних функцій, які можна підлаштовувати під конкретні вимоги.

- Підтримка віртуальних середовищ: PyCharm Community вбудовує в себе підтримку віртуальних середовищ, наприклад, завдяки VirtualEnv, що дозволяє легко створювати та управляти ізольованими середовищами для вашого проекту. Це дозволяє уникнути конфліктів залежностей і забезпечує високий рівень комфорту для розробки.

- Інтеграція з інструментами розробки: PyCharm Community інтегрується з

популярними інструментами розробки, такими як системи контролю версій (наприклад, Git), інструменти тестування (наприклад, pytest) та багато інших. Це спрощує взаємодію з цими інструментами та полегшує роботу в командному середовищі, економить час на комунікацію всередині робочого колективу та передачі всіх змін до командного проекту.

- Розширення та плагіни: PyCharm Community підтримує плагіни, які дозволяють розширювати його функціональність. Існує широкий вибір плагінів, які можна використовувати для підтримки різних фреймворків, інструментів та розширення функціональності середовища розробки згідно потребами проекту.

- Підтримка спільноти та документація: PyCharm має велику спільноту користувачів, що дає широкий діапазон ресурсів та знавців для вирішення певних проблем, які виникають під час розробки. Крім того, існує безліч документації та навчальних матеріалів, які допомагають вивчити особливості роботи з PyCharm та покращити особисті навички розробки на Python.

Підбиваючи підсумки, PyCharm Community являє собою зручне середовище для розробки як малих одиночних програм, так і великих командних проектів. Його переваги дозволяють економити час на виправлення нефункціональних задач, зберігаючи чисте візуальне бачення коду, його легкість читання та керування ним.

2.1.3 PyQt6

PyQt6[9] є потужною бібліотекою, яка надає засоби для розробки графічного інтерфейсу користувача (GUI) в програмах на мові Python. Дана бібліотека включає в себе близько 1000 класів, які реалізовані Пайтон модулями. Основними перевагами використання PyQt6 можна представити як:

- Комплексність і повнота: PyQt6 заснована на популярній бібліотеці Qt, що дозволяє розробникам та програмістам створювати повнофункціональні та професійні GUI додатки. Вона надає доступ до всіх можливостей Qt, включаючи різноманітні елементи управління (кнопки, поля введення, таблиці, слайдери, чекбокси), віджети, що дозволяють розташовувати всі елементи , графічний вивід,

роботу з подіями.

- **Переносимість:** PyQt6 підтримує багато платформ, включаючи Windows, macOS, Linux і багато інших. Це означає, що GUI додатки, розроблені з використанням PyQt6, можна легко переносити на різні операційні системи без необхідності переписування коду.

- **Можливості налаштування і стилізації:** PyQt6 надає багатий набір засобів для налаштування та стилізації GUI додатків. Ви можете використовувати каскадні таблиці стилів (CSS) для стилізації вигляду елементів інтерфейсу. PyQt6 також підтримує масштабування та локалізацію, що дозволяє створювати користувацький інтерфейс, пристосований до різних потреб і налаштувань користувачів.

- **Інтеграція з мовою Python:** PyQt6 є повноцінною бібліотекою Python і надає зручні інструменти для роботи з мовою Python. Вона дозволяє легко використовувати функціональність Python, таку як обробка подій, робота зі списками та словниками, виконання динамічного коду та інші можливості.

- **Документація та спільнота:** PyQt6 має велику кількість документації, прикладів коду та ресурсів, які допомагають розробникам вивчити та використовувати бібліотеку. Крім того, існує активна спільнота PyQt, яка надає підтримку, відповіді на запитання та розробляє різноманітні розширення та інструменти.

Загалом, PyQt6 є потужним інструментом для розробки GUI додатків з використанням мови Python. Вона надає широкі можливості, переносимість і підтримку спільноти, що робить її привабливим вибором для розробки GUI додатків у Python.

2.1.4 SQLite3

SQLite3[10] являє собою вбудовану бібліотеку C, що надає дискову базу даних, і не потребує з'єднання з сервером, з можливістю внутрішнього зберігання даних. Надає чимало різних переваг і часто використовується для невеликих проєктів. Присутня можливість переносу бази даних в більші бази, такі як Oracle,

чи PostgreSQL. Нижче перераховані основні причини, чому варто використовувати `sqlite3` в програмуванні застосунку з використанням системи `question-answering`:

- Простота використання: SQLite3 є дуже простою в використанні базою даних. Вона не вимагає окремого сервера чи налаштувань, оскільки база даних зберігається в одному файлі.

- Вбудована підтримка: Бібліотека `sqlite3` є частиною стандартної бібліотеки Python, що означає, що її не потрібно додатково встановлювати. Легкий імпорт `sqlite3` дозволяє почати використовувати SQLite3 бази даних у своїх програмах Python.

- Кросплатформність: SQLite3 підтримується на багатьох платформах, включаючи Windows, macOS, Linux та інші. Це дозволяє переносити ваші програми, які використовують SQLite3, між різними операційними системами без змін в коді.

- Ефективність та швидкодія: представлена бібліотека добре працює з невеликими обсягами даних або простими запитами. Забезпечує швидкий доступ до даних та нескладними способами їх керування

- Надійність та стабільність: SQLite3 має доведену надійність та стабільність. Вона пройшла широкі тести та використовується в різних проектах протягом багатьох років. Дані бази даних також можуть бути легко резервовані та відновлювані.

- Розширюваність: SQLite3 підтримує можливість розширення функціональності через використання розширень та налаштувань.

Підсумовуючи загальні наведені переваги SQLite3, можна зробити висновок, що дана бібліотека Python чудово підійде під використання `question-answering` програмного забезпечення, надаючи швидку видачу значень конкретних рядків з таблиці, можливість їх швидкого редагування та переміщення, і самого створення нових таблиць.

2.1.5 Stanza

Stanza [11] є потужною бібліотекою для обробки природної мови, яка має підтримку української мови. Рисунок 2.1 демонструє наявність української мови в списку доступних мов для роботи з текстами[12].










| | | | | | | |
|---------------|-----|------|-------|---|---|---|
| Turkish | tr | imst | 1.0.0 |  |  |  |
| Ukrainian | uk | iu | 1.0.0 |  |  |  |
| Upper Sorbian | hsb | ufal | 1.0.0 |  |  |  |

Рисунок 2.1 – Українська мова в Stanza

Нижче наведено основні причини використання Stanza для question-answering системи з підтримкою української мови:

- Мовна підтримка. Stanza надає можливість розпізнавати, аналізувати та робити мітки в тексті на рівні речень, слів і морфологічних ознак для української мови. Це включає в себе розпізнавання частин мови, лематизацію, визначення залежностей та іменованих сутностей. Така мовна підтримка є необхідною для розуміння та обробки текстів українською мовою в контексті question-answering системи.

- Якість аналізу. Stanza використовує сучасні методи та моделі для проведення аналізу природної мови. Вона базується на навчальних даних, що включають українські тексти, що дозволяє отримувати більш точні результати при аналізі тексту українською мовою. Це допомагає покращити якість обробки тексту та забезпечує більш точні результати для question-answering системи.

- Простота у використанні. Stanza надає простий та зручний інтерфейс для використання у програмах. Вона має добре задокументовану API, яка дозволяє легко інтегрувати її в існуючий код question-answering системи.

- Швидкодія. Stanza відома своєю швидкодією та ефективністю. Вона використовує передові методи обробки природної мови, які оптимізовані для швидкого виконання на багатьох процесорах. Це особливо важливо для системи question-answering, де швидкість обробки запитів є критичним фактором.

Загалом, використання Stanza для question-answering системи з підтримкою української мови дозволяє отримати потужні інструменти для обробки тексту, які забезпечують якість, точність та ефективність. Це допомагає покращити здатність системи відповідати на запитання користувачів та забезпечує зручну роботу з українським текстом.

2.2 Моделювання вимог до програмного забезпечення

Діаграма варіантів використання (Use Case Diagram) є графічним інструментом моделювання, який використовується для зображення взаємодії між акторами (користувачами або зовнішніми системами) та системою. Вона демонструє функціональність системи з точки зору її користувачів, описує можливі варіанти використання застосунку.

Основна мета діаграми варіантів використання - це зрозуміти, які дії можуть виконувати актори та як вони взаємодіють з системою. Діаграма включає в себе наступні елементи:

- Актори: Актори представляють ролі або сутності, які взаємодіють з системою. Це можуть бути люди, інші системи або навіть зовнішні об'єкти. Вони зображуються у вигляді піктограм або простої фігури людини.

- Варіанти використання: Варіанти використання або "сценарії" описують конкретні дії або функції, які можуть бути виконані системою. Вони представлені у вигляді овалів або прямокутників і містять короткий опис того, що вони описують.

- Взаємодія: Взаємодія між акторами та варіантами використання показується за допомогою відрізків, що сполучають актора та варіант використання, який він викликає.

- Включення та розширення: Включення (include) та розширення (extend) - це механізми, що дозволяють описувати включення чи розширення одного варіанту використання в іншому.

Діаграма варіантів використання дозволяє зорієнтуватися у функціональних

можливостях системи, виокремити основні варіанти використання та ідентифікувати акторів, що взаємодіють з системою. Вона є потужним інструментом для аналізу вимог та визначення основних функцій системи.

Згідно вимог до програмного забезпечення, було складено діаграму варіантів використання програмного забезпечення, що зображена на рисунку 2.2.

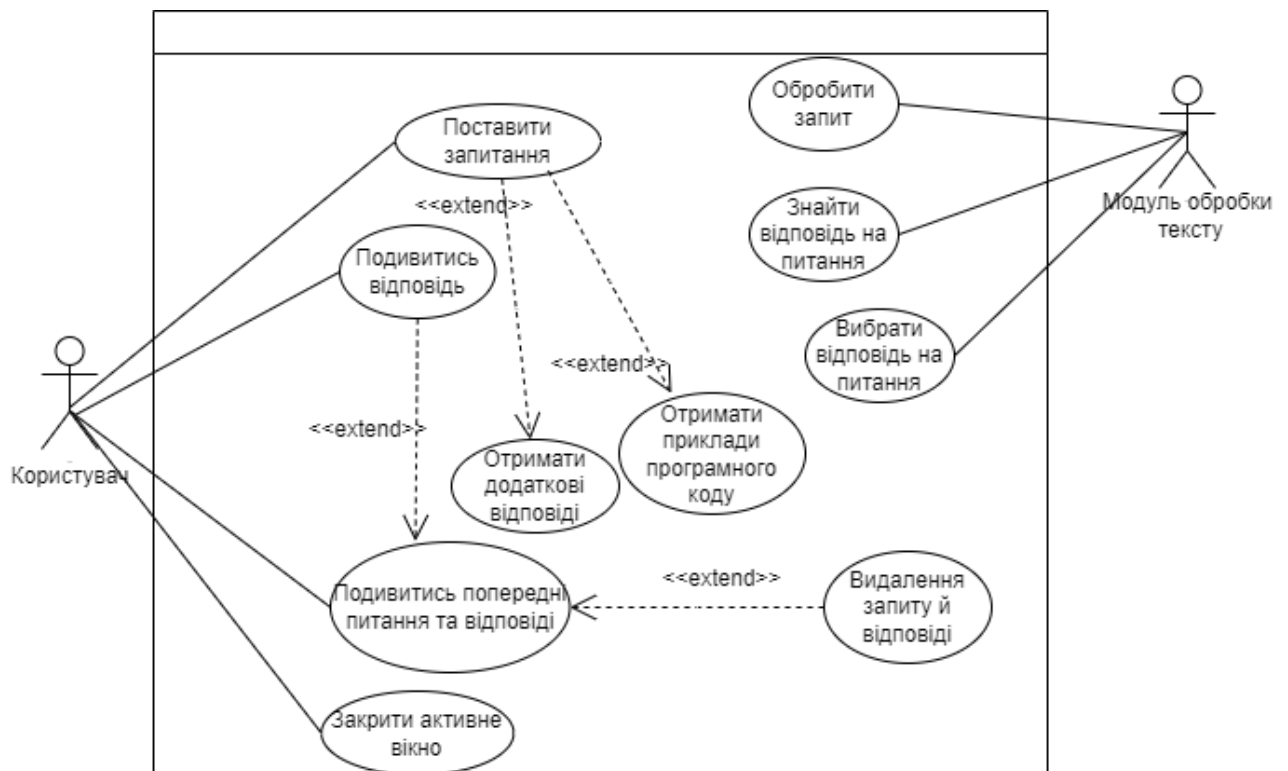


Рисунок 2.2 – Діаграма варіантів використання

Дана діаграма демонструє набір можливостей користувача:

- Поставити власне запитання, надрукувавши його на клавіатурі програмного пристрою. Дана опція розширюється двома варіантами: можливість отримати додаткові відповіді, що по змісту схожі з поставленим та отримати приклад програмного коду (чи декілька, в залежності від теми питання), який показує теорію на практиці.

- Можливість побачити попередні запитання та відповіді на них. Цей варіант не включатиме контекстні відповіді, що пропонуються користувачеві під час поставлення запитання. Розширюється можливістю видалити з бази даних відображену відповідь.

- Закрити активне вікно. Кожна окрема дія супроводжується відкриттям нового вікна, яке користувач будь-якої миті може закрити.

Також дана діаграма представляє перелік функцій, покладених на модуль обробки тексту. Серед них: обробити запит, знайти відповідь на запитання та видати дану відповідь користувачеві.

2.3 Розробка алгоритму обробки текстового запиту

Текстовий запит – це речення, яке вводить користувач з метою знаходження відповіді на незрозумілу тему. Зазвичай, в question-answering системах такі запити пишуться в запитальній формі, але бувають формулювання та виключення по типу «Покажи принцип роботи...», «Виведи на екран...» тощо. На рисунку 2.3 наведено алгоритм обробки текстового запиту.

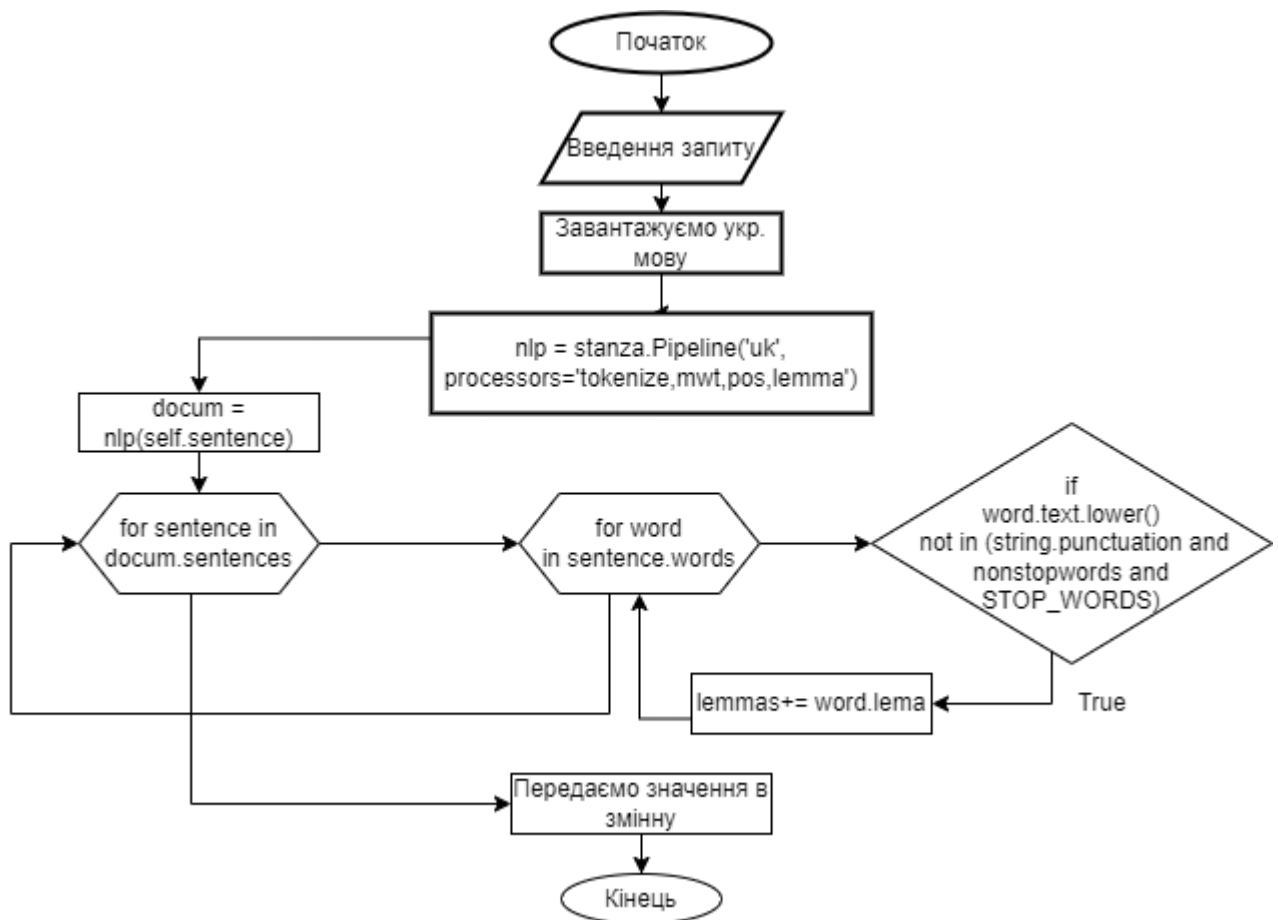


Рисунок 2.3 – Блок-схема обробки текстового запиту

Перш за все, встановлюємо параметри, за якими ми будемо працювати із текстом. Нам знадобляться `tokenize` та `lemma`, `mwt` – multi-word token для розширення кількості базових значень токена, та `pos` – part-of-speech, морфологічний аналіз слова в реченні. Додатково підключені слова-вийнятки, що являються ключовими словами в мові C++ (`if`, `else`, `cout`, `class`...).

Наступним кроком являється початок циклу. Весь текстовий запит розбивається на речення, в той час як речення розбивається на складові елементи, що відокремлені пробілом, тобто токенизація. Всі токени зводяться до малої літери, і перевіряється їх приналежність до обмежувальних списків: `string.punctuation` відповідає за пунктуаційні знаки, `nonstopwords` – список ключових слів мови C++ та `STOP_WORDS` як великий перелік всіх слів, які не приймають участі у розумінні речення.

Після завершення двовимірного циклу, на виході було отримано список tokenів, зведених до базової форми слова.

2.4 Розробка класів

Діаграма класів – UML інструмент, який відображає відношення класів, об'єктів між собою в програмі. Кожен блок класу складається з трьох елементів:

- назва класу - розташована у верхньому полі;
- атрибути класу - розташовані посередині, та відображає тип даних;
- методи класу - розташовані знизу, відображає функції класу.

На рисунку 2.4 наведено розроблену діаграму класів застосунку.

В програмному застосунку створено файл `mainWindow.py` з класом `MainWindow`. Цей файл виконуватиметься першим при запуску додатка. Він містить елементи інтерфейсу та стилістичні параметри під текстове поле `QLineEdit`, в яке користувач зможе ввести запитання, кнопка `buttonQue`, яка перенаправлятиме на сторінку з відповідями, кнопка `buttonPrevious`, що відкриватиме вікно `PreviousAnswer`. Також, дане вікно містить два чекбокси, які відповідають за відображення коду програми до питання, яке збирається

поставити користувач та можливість отримати відповіді на запитання, що схожі з поставленим.

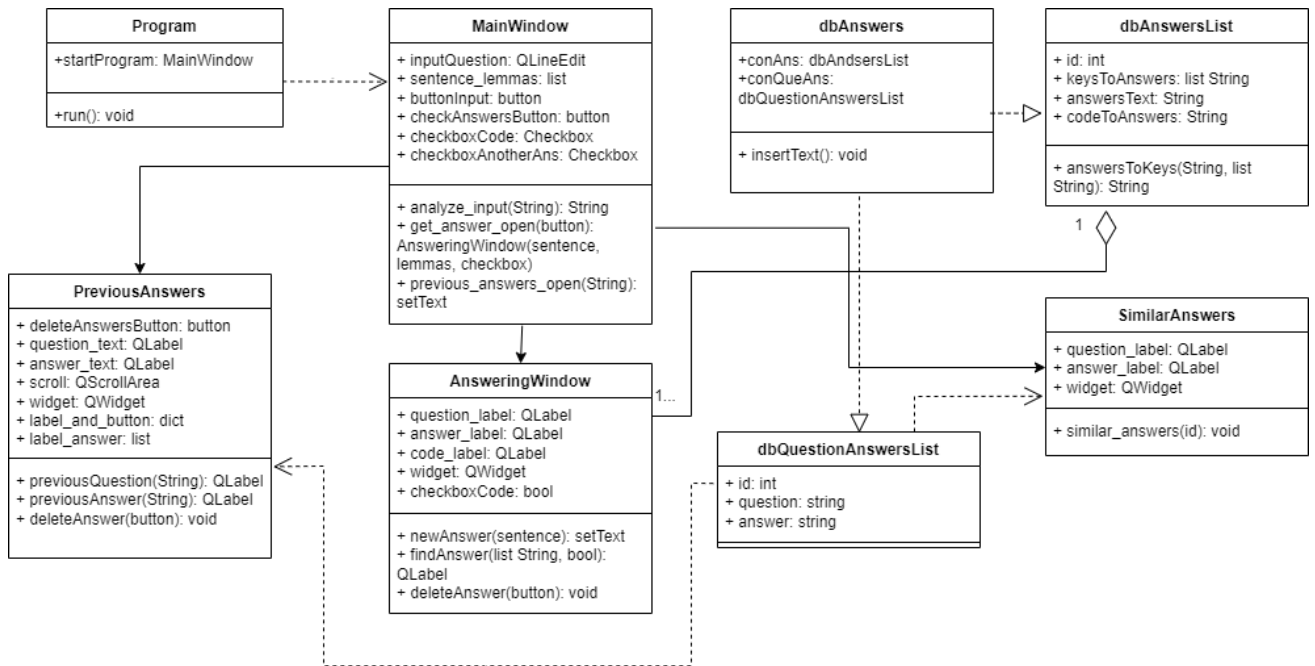


Рисунок 2.4 – Діаграма класів

Функціонал даного класу дозволяє відкривати три інших вікна: AnsweringWindow, SimilarAnswer та PreviousAnswers.

AnsweringWindow містить рядок question_text з питанням та answer_text з відповіддю на нього. Опціонально виводить code_text, якщо був натиснутий відповідний чекбокс. В середині цього класу методом findAnswer шукає відповідь з готової бази даних.

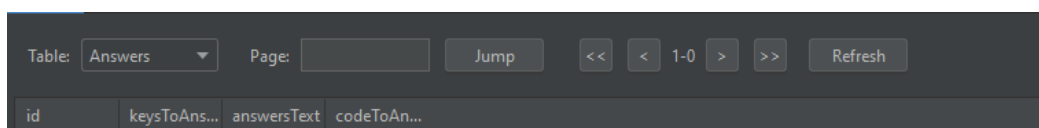
SimilarAnswers відкривається за умови натискання на чекбокс checkboxAnotherAns, і містить список питань, які по смислу схожі з поставленим запитанням.

PreviousAnswer відображає всі попередні відповіді списком, який включає в себе два QLabel та прив'язану до них кнопку. При натисканні delete_button, видаляються всі три об'єкта.

2.5 Розробка моделі з відповідями

Перш ніж писати код програми, потрібно створити модель, по якій система зможе орієнтуватись, на які питання видавати потрібну відповідь. Питання опрацьовувались шляхом пошуку на форумах статей, аналізом їх заголовків, та виписуванням точних формулювань.

Модель буде зберігатись у вигляді бази даних, використовуючи бібліотеку Python SQLite3, що містить стовпці з різними збереженими параметрами (рис.2.5).



The image shows a screenshot of a database management interface. At the top, there is a control bar with a dropdown menu set to 'Table: Answers', a 'Page:' field, a 'Jump' button, and navigation buttons: '<<', '<', '1-0', '>', and '>>'. A 'Refresh' button is also present. Below this is a table header with four columns: 'id', 'keysToAns...', 'answersText', and 'codeToAn...'. The table body is currently empty.

Рисунок 2.5 – Вигляд таблиці з відповідями

Структурні елементи моделі, які розділені на стовпці бази даних:

- ID. Кожен рядок таблиці матиме власний ID, по якому може надаватись доступна інформація в сусідніх клітинках, робитись перестановка змінних, додавати нові значення чи змінювати старі.

- keysToAnswer. Ключові слова, що стосуються закладеної відповіді. Вони виступають прапорами, по яким буде здійснюватися співставлення відповідності питання, що було написано користувачем та відповіддю, яку він має отримати.

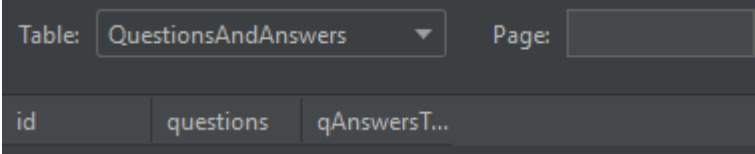
- answerText. Тексти відповідей, які були завчасно підготовлені з навчально-матеріальної бази, що була взята з підручників по програмуванню в Держаному університеті телекомунікацій, з кафедри Інженерії програмного забезпечення, підручник 2011-го року випуску за авторства О.Г. Трофименко «С++ теорія та практика», та підручник з цього ж університету року видачі 2011-го, авторства Ю.І. Грицюка та Т.Є. Рака «Програмування мовою С++».

Тексти розділені на розділи та підрозділи, відповідно чому було сформовано зміст відповідей.

- codeToText. Дана клітинка БД відповідає за збереження кодів програм, які демонструють текстовий опис алгоритмів, парадигм тощо, та короткий опис

(декілька слів), який тип програмного коду демонструється.

На рисунку 2.6 зображено стовпчики таблиці бази даних, яка містить: Id – порядковий номер рядка, Questions – список всіх запитань, що були написані користувачем, qAnswersText – список відповідей на запити.



| id | questions | qAnswersT... |
|----|-----------|--------------|
|----|-----------|--------------|

Рисунок 2.6 – База даних з питаннями та відповідями

2.6 Проектування та розробка інтерфейсу

Для візуального представлення розробленого додатку було використано бібліотеку Python PyQt6. Перш ніж розпочати працювати з даною бібліотекою, її необхідно встановити. Робиться це командою `pip install pyqt6`. Далі було реалізовано клас `MainWindow(QWidget)`, який успадковує клас `QWidget`, що дозволяє змінювати характеристики вікна, та встановлювати власні.

Перелік функцій для виставлення характеристик вікна:

- `setWindowTitle("Назва")` – змінює текст назви вікна.
- `setGeometry(a, b, c, d)` – приймає 4 параметри, що регулюють розмір вікна (c – висота, d – ширина) та його розташування відносно положення на моніторі (a – горизонтальний відступ від лівого краю, b – вертикальний відступ від верхнього краю).
- `setStyleSheet(str)` – приймає значення рядка, який може виставляти стиль сторінки за допомогою стилів CSS.

Далі було сплановано візуальне розташування необхідних функціональних частин. Серед них:

- Поле для вводу запитання.
- Кнопка, що при натисканні відкриватиме вікно з відповідями.

- Два чекбокси. Один пропонуватиме можливість отримати приклад програмного коду, інший – відкриватиме вікно з відповідями, що семантично схожі на поставлене запитання.

- Кнопка, що відкриватиме вікно з попередніми запитаннями та відповідями.

Після цього, було створено змінну `main_layout`, яка буде зберігати горизонтальний лейаут `QHBoxLayout()`, та `layout`, що розташує відповідні елементи у вертикальному порядку.

Наступним чином, текстове поле та кнопка для відповіді були розміщені поруч одне з одним, натомість інші елементи вертикально одне відносно одного.

3 ОПИС ФУНКЦІОНУВАННЯ ТА ТЕСТУВАННЯ СИСТЕМИ

3.1 Порядок роботи з системою

В розробленому додатку за замовчуванням відкривається одне вікно, яке має поле для вводу текстового запиту, та обов'язкову для натиснення кнопку «Отримати відповідь». Опціонально, користувач може вибрати два checkbox: один з них представляє програмний код, пов'язаний з відповіддю, інший відкриватиме вікно з подібними відповідями. На рисунку 3.1 зображено дане вікно та візуальне розміщення кожного з елементів.

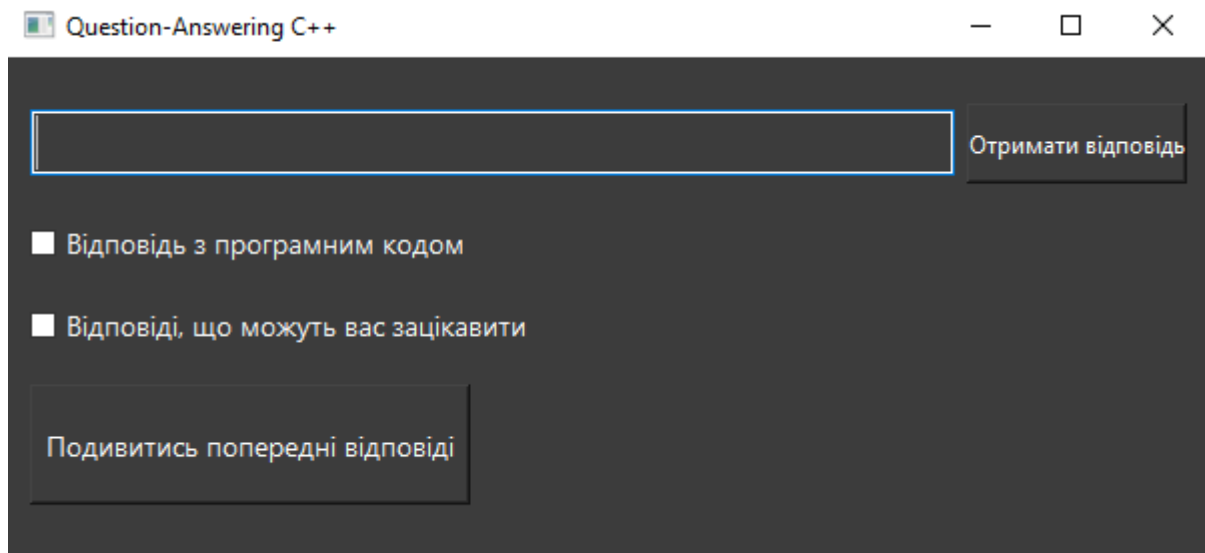


Рисунок 3.1 – Візуальне представлення інтерфейсу головного вікна

Для прикладу, користувач ввів запит «Як працює настанова if?», і не вибрав варіант з отриманням відповіді з демонстраційним програмним кодом. Цей запит обробляється, та на основі результатів цієї обробки вибирається відповідна відповідь. Рисунок 3.2 демонструє роботу подібного варіанту подій.

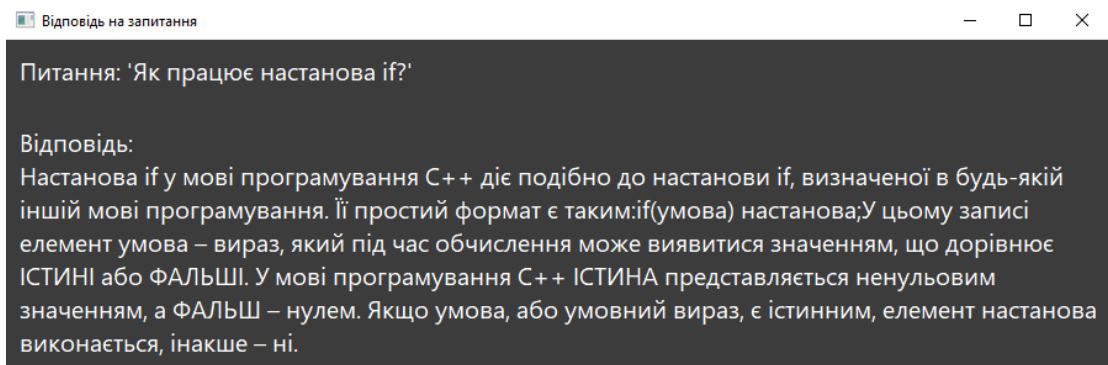


Рисунок 3.2 – Відповідь користувачеві без програмного коду

Якщо користувач вирішить отримати код програми, відкриється інший варіант програмного вікна, який зображений на рисунку 3.3. Програмний код може повторюватись при змінній відповіді, якщо питання стосуються одного й того ж розділу чи теми.

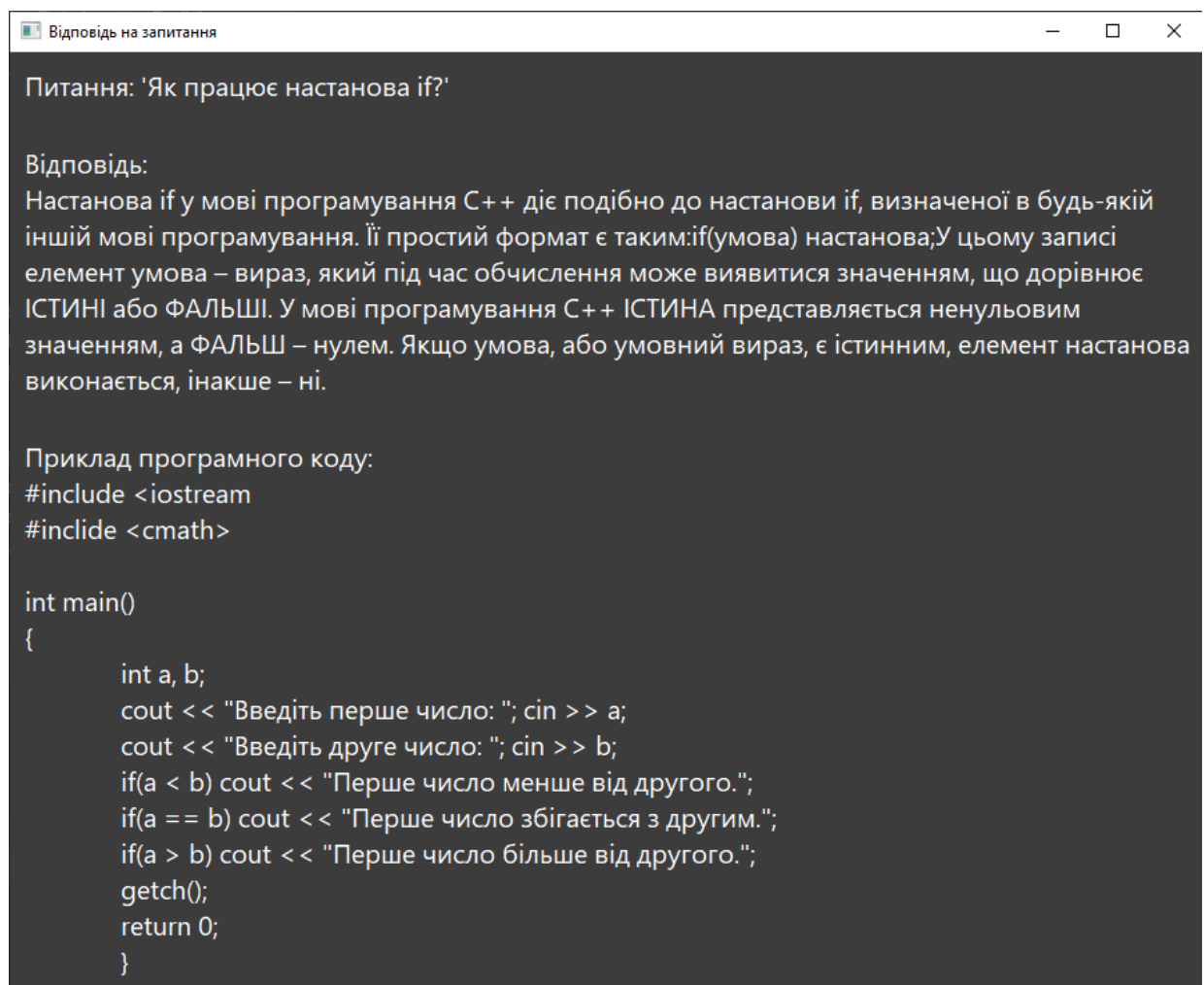


Рисунок 3.3 – Відповідь з програмним кодом

Далі наведено екранні форми роботи вікна з попередніми запитаннями. Це список питань та відповідей, які раніше були поставлені користувачем. На рисунку 3.4 відображено список з трьох запитань, на які користувач отримав відповіді. Рисунок 3.5 демонструє видалення запитання та відповідей з цього списку.

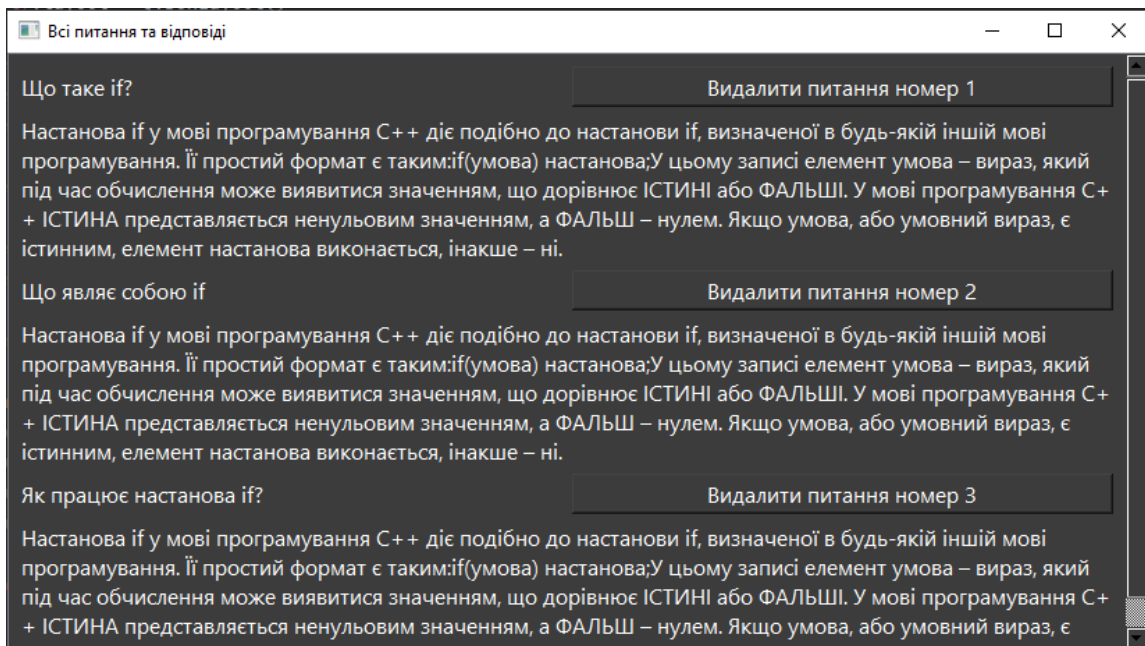


Рисунок 3.4 – питання та відповіді, що були поставлені раніше

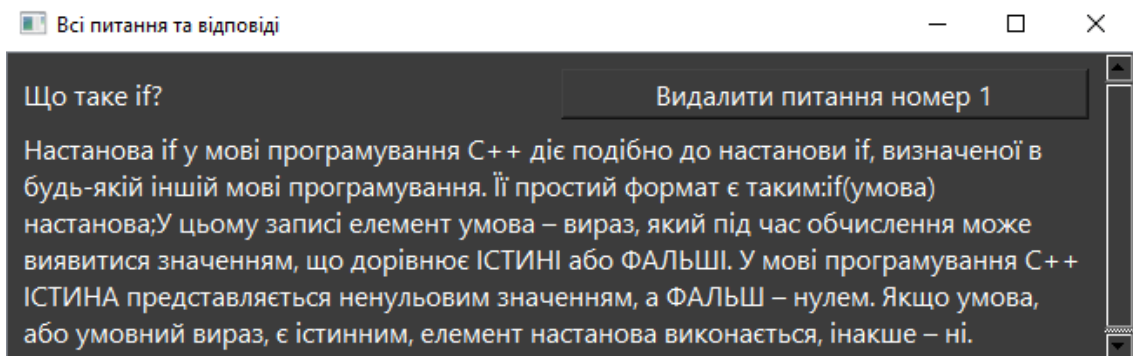


Рисунок 3.5 – питання та відповіді після видалення

Рисунок 3.6 демонструє вікно, яке відображає відповіді на суміжне запитання до запитання «оператор виведення».

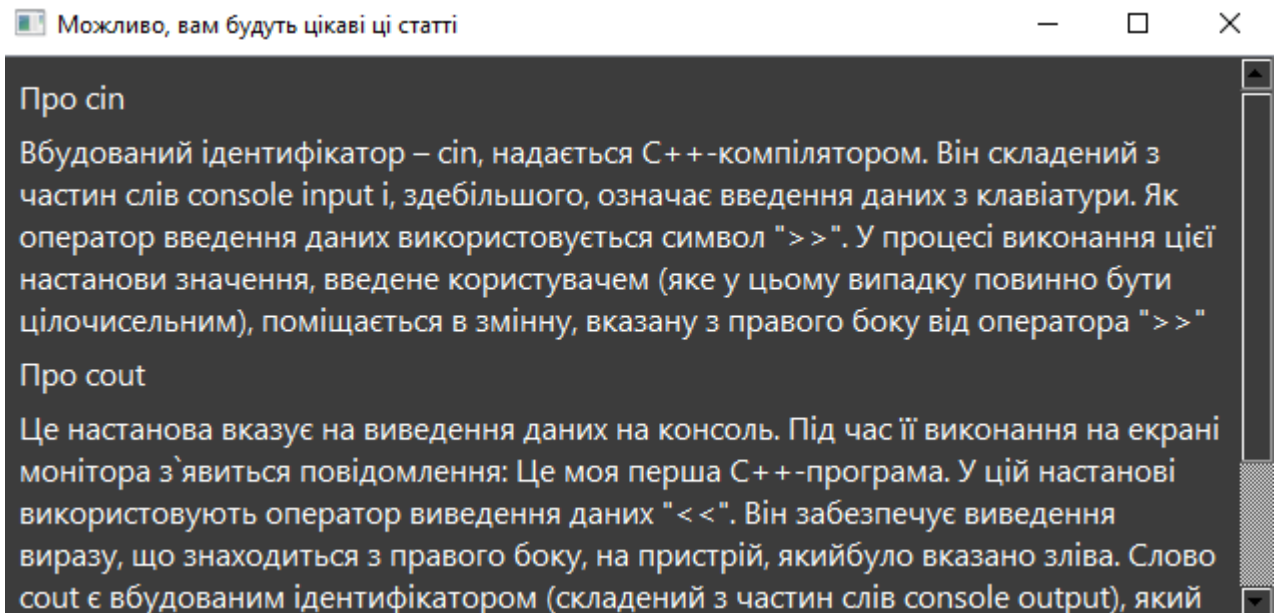


Рисунок 3.6 – Вікно суміжних відповідей

3.2 Тестування програмного забезпечення

Тестуванням програмного забезпечення називають процес встановлення відповідності вимог до втіленого застосунку та його функцій. Результатом тестування є максимально об'єктивна оцінка програмного продукту, з урахуванням всіх позитивних та негативних сторін.

Тестування можна описати як верифікацію та валідацію застосунку.

Верифікація – перевірка складових компонентів програми чи системи, з метою виявлення виконання цілей та задач виробничого процесу.

Валідація – знаходження відповідності потребам користувачів, сформованих вимог до системи та відповідних реалізацій в ПЗ.

Була сформована Таблиця 4.1 на основі тест-кейсів, яка демонструє перелік очікувань користувачів, очікуваний спосіб вирішення проблеми згідно вимог та результат тестування.

Всі тести було пройдено успішно, що підтверджує готовність системи до взаємодії з реальними користувачами.

Таблиця 4.1 – Результати тестування додатку

| № | Тест-кейс | Очікуваний результат | Результат тестування |
|---|--|--|----------------------|
| 1 | Натиснути кнопку «Отримати відповідь» декілька разів поспіль | Відкриватиметься лише 1 вікно з відповіддю | passed |
| 2 | Видалити всі відповіді у вікні «Попередні відповіді» | Залишиться порожнє вікно, яке при повторному відкритті буде містити відповіді | passed |
| 3 | Видалити всі відповіді у вікні «Попередні відповіді» | Візуально відповіді видаляться, проте вміст бази даних з питаннями та відповідями не зміниться | passed |
| 4 | Активувати чекбокс «Отримати код програми» та «Отримати схожі відповіді» одночасно | Відкриється вікно з відповіддю на запитання, разом з кодом програми, та додаткове вікно зі схожими відповідями | passed |
| 5 | Введення запитання з англійськими словами по типу if, else, cout | Включення даних токенів в список ключових слів для пошуку відповідей | passed |
| 6 | Ввести текстове запитання без розділових знаків | Відповідь надається коректно незалежно від наявності чи відсутності розділових знаків | passed |

ВИСНОВКИ

1. Проведено аналіз методів обробки запитів користувачів в Question-Answering системах на основі технологій Natural Language Processing. Визначено особливості роботи з текстами українською мовою.

2. Виконано огляд застосунків, які можуть бути використані для вивчення дисципліни «Програмування С++» в форматі «питання-відповідь». Визначено їх ключові переваги та недоліки, особливості формування відповідей.

3. Проведено аналіз навчально-методичних матеріалів дисципліни «Програмування С++». Для формування відповідей виконано сегментацію текстового корпусу дисципліни на фрагменти, які можуть виступати відповідями на типові запитання з дисципліни. Розроблено алгоритм для пошуку найбільш релевантних відповідей на запити студентів в текстовому корпусі дисципліни.

4. На основі огляду та налізу аналогів та з урахуванням особливостей формування відповідей на основі розміченого корпусу навчально-методичних матеріалів дисципліни «Програмування С++» було сформувано функціональні та нефункціональні вимоги до програмного забезпечення.

5. Спроектовано та розроблено програмне забезпечення, що реалізує технологію Question-Answering на базі матеріалів дисципліни «Програмування С++». Застосунок дозволяє ставити запитання природньою мовою, передивлятися та видаляти попередні відповіді, отримувати додаткові відповіді, що пов'язані з поставленим запитанням, та отримувати в якості відповіді фрагменти програмного коду.

6. Проведене тестування програмного забезпечення підтвердило релевантність між поставленими питаннями та відповідями, які генерує система.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Статистика середніх балів вступників [Електронний ресурс]. – 2023. – Режим доступу до ресурсу: <https://vstup.osvita.ua/y2022/r27/82/>
2. What is C++? [Електронний ресурс]. – 2023. – Режим доступу до ресурсу: <https://www.coursera.org/articles/what-is-c-plus-plus>.
3. Caballero M. A BRIEF SURVEY OF QUESTION ANSWERING SYSTEMS / Michael Caballero. // International Journal of Artificial Intelligence & Applications (IJAIA). – 2021. – С. 1–3.
4. Chat GPT [Електронний ресурс]. – 2023. – Режим доступу до ресурсу: <https://openai.com/blog/chatgpt>.
5. PyCharm Features [Електронний ресурс] // JetBrains. – 2023. – Режим доступу до ресурсу: <https://www.jetbrains.com/pycharm/features/>.
6. Rizel S. Why Python keeps growing, explained [Електронний ресурс] / Scarlett Rizel // GitHub Blog. – 2023. – Режим доступу до ресурсу: <https://github.blog/2023-03-02-why-python-keeps-growing-explained/>
7. Documentation v6.5.1 >> Introduction [Електронний ресурс]. – 2023. – Режим доступу до ресурсу: <https://www.riverbankcomputing.com/static/Docs/PyQt6/introduction.html>.
8. Utsav M. 10 Top NLP Algorithms [Електронний ресурс] / Mishra Utsav. – 2022. – Режим доступу до ресурсу: <https://www.analyticssteps.com/blogs/top-nlp-algorithms>.
9. PyQt5 vs PyQt6 What are the differences, and is it time to upgrade? [Електронний ресурс] // PythonGUIs. – 2023. – Режим доступу до ресурсу: <https://www.pythonguis.com/faq/pyqt5-vs-pyqt6/>.
10. About SQLite [Електронний ресурс]. – 2023. – Режим доступу до ресурсу: <https://www.sqlite.org/about.html>.
11. Stanza – A Python NLP Package for Many Human Languages [Електронний ресурс] // Stanza. – 2023. – Режим доступу до ресурсу: <https://stanfordnlp.github.io/stanza/>.

12. Available Models & Languages [Електронний ресурс]. – 2023. – Режим доступу до ресурсу: https://stanfordnlp.github.io/stanza/available_models.html.
13. C++ Tutorial [Електронний ресурс]. – 2023. – Режим доступу до ресурсу: <https://www.w3schools.com/cpp/default.asp>.
14. Vectorization Techniques in NLP [Guide] [Електронний ресурс]. – 2023. – Режим доступу до ресурсу: <https://neptune.ai/blog/vectorization-techniques-in-nlp-guide>.
15. Marshall C. Named Entity Recognition [Електронний ресурс] / Christopher Marshall. – 2019. – Режим доступу до ресурсу: <https://medium.com/mysuperaai/what-is-named-entity-recognition-ner-and-how-can-i-use-it-2b68cf6f545d>.
16. Organization of a text automatic summarization system based on fuzzy hierarchical neural network / А. Ю. Михайлюк, О. С. Михайлюк, Л. М. Огнівчук, В. П. Тарасенко // Спеціалізовані системи, мережі та інформаційні технології / А. Ю. Михайлюк, О. С. Михайлюк, Л. М. Огнівчук, В. П. Тарасенко.. – (УДК 004.9). – С. 53–61.

ДОДАТОК А

ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ



ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ
ТЕХНОЛОГІЙ
КАФЕДРА ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ



Розробка програмного забезпечення Question-Answering системи на основі навчально-методичних матеріалів дисципліни "Програмування C++" мовою Python

Виконав студент 4 курсу
групи ПД-41
Агашков Андрій Юрійович
Керівник роботи
К.т.н. доц., доцент кафедри ІПЗ
Золотухіна Оксана Анатоліївна

Київ – 2023

МЕТА, ОБ'ЄКТ ТА ПРЕДМЕТ ДОСЛІДЖЕННЯ

- **Мета** - спрощення процесу надання відповідей на питання студентів, що вивчають дисципліну «Програмування C++», за рахунок використання програмного забезпечення з технологією question-answering, реалізованого мовою Python.
- **Об'єкт** - процес надання відповідей на питання студентів, що вивчають дисципліну «Програмування C++».
- **Предмет** - програмне забезпечення question-answering системи на основі навчально-методичних матеріалів дисципліни «Програмування C++».

ЗАДАЧІ ДИПЛОМНОЇ РОБОТИ

1. Провести аналіз підходів до обробки користувацьких запитів в Question-Answering системах, в тому числі, запитів українською мовою.
2. Провести огляд застосунків, які можна використовувати для вивчення дисципліни «Програмування C++» в форматі «питання-відповідь».
3. Провести аналіз навчально-методичних матеріалів дисципліни «Програмування C++» та визначити особливості формування відповідей.
4. Сформувані функціональні та нефункціональні вимоги до програмного забезпечення.
5. Спроекувати та реалізувати програмне забезпечення Question-Answering системи, яка дозволяє формувати відповіді на питання студентів на основі навчально-методичних матеріалів дисципліни «Програмування C++».
6. Провести тестування програмного забезпечення.

3

АНАЛІЗ АНАЛОГІВ



| Найменування | W3School | SoloLearn | ChatGPT | Розроблений додаток |
|--|-----------|----------------------|--|---------------------|
| Назва засобів | | | | |
| Україномовна локалізація | - | - | + | + |
| Платформа | Браузер | Android, iOS | Браузер, Windows 10, iOS, Mac, Android | Windows 10 |
| Матеріал відповідно навчальної програми дисципліни «Програмування C++» | + | + | + | + |
| Можливість дивитись поставлені раніше запитання | - | - | + | + |
| Можливість поставити запитання українською мовою | - | - | + | + |
| Доступ без мережі інтернет | Відсутній | Обмежений функціонал | Тільки при попередньому завантаженні | Присутній |
| Можливість отримання відповідей, що сумісні з поставленими запитаннями | - | - | - | + |

4

ВИМОГИ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Функціональні вимоги

1. Можливість введення користувачем тексту запитання з клавіатури.
2. Наявність засобів попередньої обробки тексту запиту.
3. Формування текстової відповіді, релевантної ключовим словам, наявним в користувацькому запиті, на основі навчально-методичних матеріалів дисципліни «Програмування C++».
4. Можливість відображення у відповіді на питання програмного коду за темою.
5. Можливість пошуку відповіді за суміжними темами.
6. Можливість відображення переліку питань, які користувач ставив раніше, відповіді на них, можливість видалення цих запитань.

Нефункціональні вимоги

1. Локалізація українською мовою.
2. Можливість роботи без підключення до мережі інтернет.
3. Працездатність на платформі Windows 10.

5

ПРОГРАМНІ ЗАСОБИ РЕАЛІЗАЦІЇ

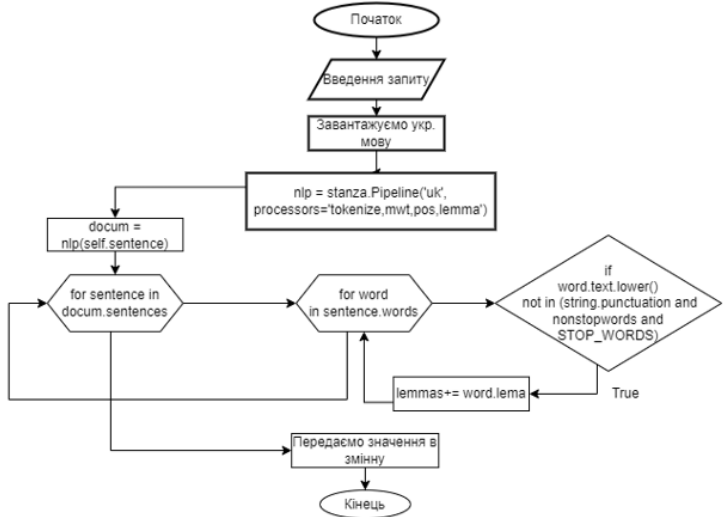


PyCharm



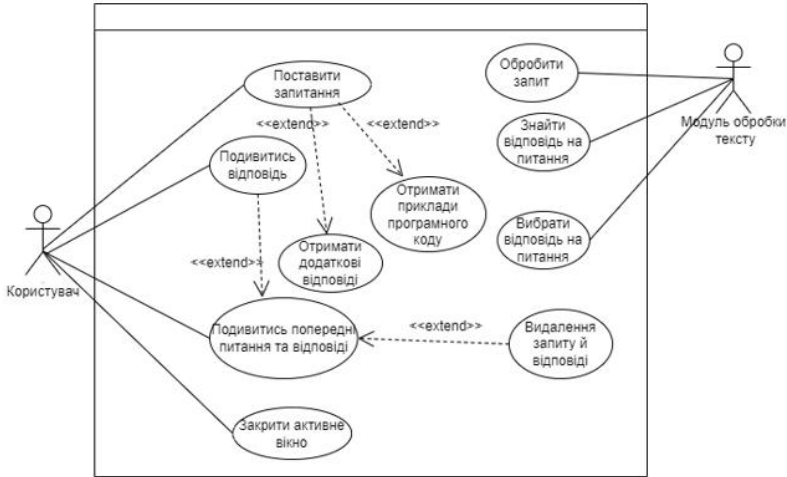
6

БЛОК-СХЕМА ОБРОБКИ ЗАПИТУ



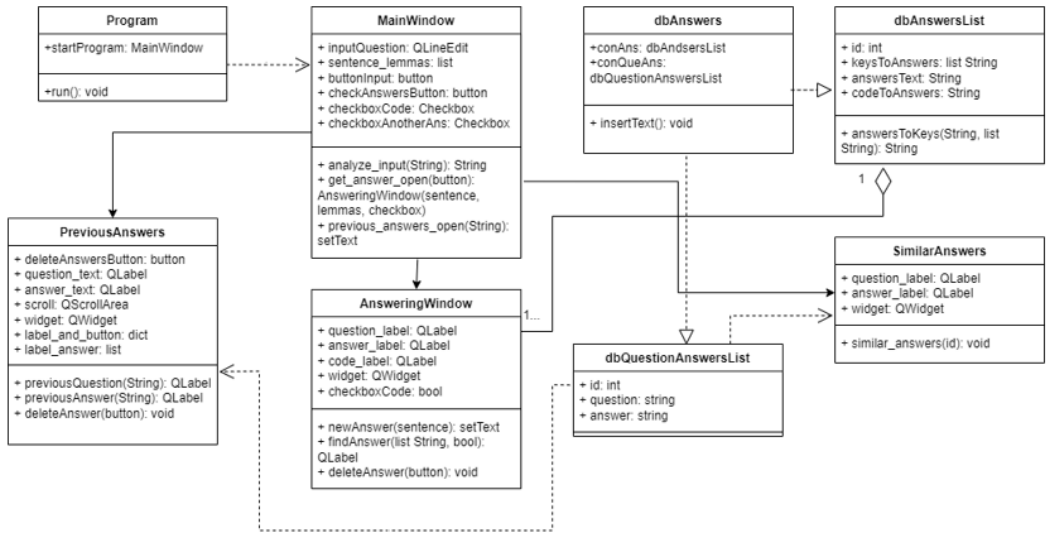
7

ДІАГРАМА ВАРІАНТІВ ВИКОРИСТАННЯ



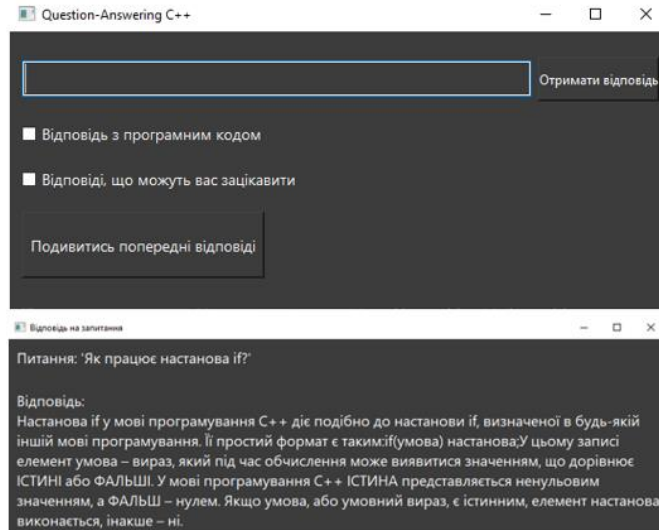
8

ДІАГРАМА КЛАСІВ



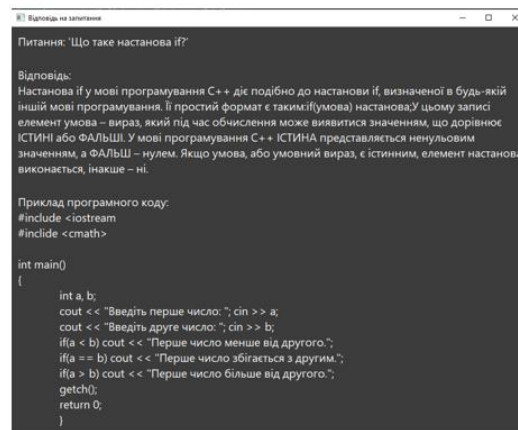
9

ЕКРАННІ ФОРМИ



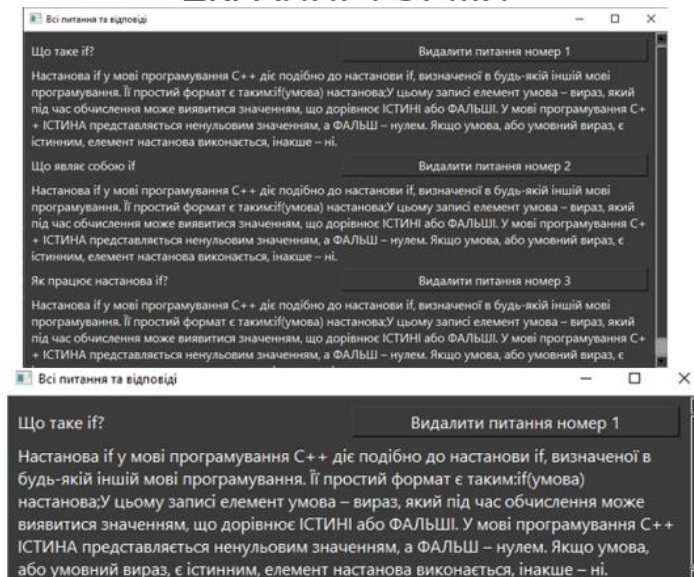
10

ЕКРАННІ ФОРМИ



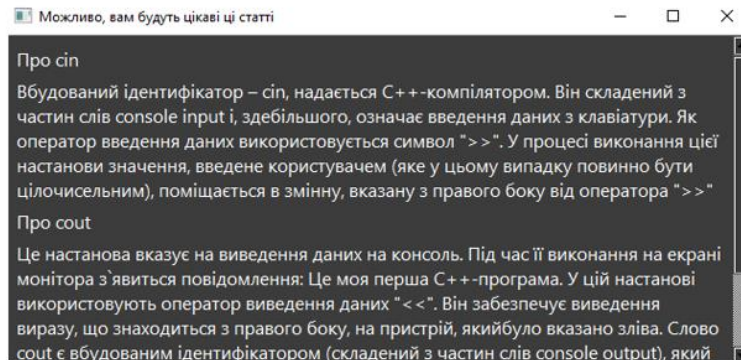
11

ЕКРАННІ ФОРМИ



12

ЕКРАННІ ФОРМИ



13

АПРОБАЦІЯ РЕЗУЛЬТАТІВ ДОСЛІДЖЕННЯ

1. Агашков А.Ю. Natural Language Processing (NLP) в процесях обробки запитів/ А. Ю. Агашков, О. А. Золотухіна // Сучасні аспекти діджиталізації та інформатизації в програмній та комп'ютерній інженерії. Збірник тез 01.06.2023-03.06.2023 ДУТ, м. Київ – К. ДУТ 2023 – подано до друку.
2. Агашков А.Ю. Machine Learning засобами Python при роботі з текстами/ А. Ю. Агашков, О. А. Золотухіна // Сучасні аспекти діджиталізації та інформатизації в програмній та комп'ютерній інженерії. Збірник тез 01.06.2023-03.06.2023 ДУТ, м. Київ – К. ДУТ 2023 – подано до друку.

14

ВИСНОВКИ

1. Проведено аналіз методів обробки запитів користувачів в Question-Answering системах на основі технологій Natural Language Processing. Визначено особливості роботи з текстами українською мовою.
2. Виконано огляд застосунків, які можуть бути використані для вивчення дисципліни «Програмування C++» в форматі «питання-відповідь». Визначено їх ключові переваги та недоліки, особливості формування відповідей.
3. Проведено аналіз навчально-методичних матеріалів дисципліни «Програмування C++». Для формування відповідей виконано сегментацію текстового корпусу дисципліни на фрагменти, які можуть виступати відповідями на типові запитання з дисципліни. Розроблено алгоритм для пошуку найбільш релевантних відповідей на запити студентів в текстовому корпусі дисципліни.
4. На основі огляду та аналізу аналогів та з урахуванням особливостей формування відповідей на основі розміченого корпусу навчально-методичних матеріалів дисципліни «Програмування C++» було сформувано функціональні та нефункціональні вимоги до програмного забезпечення.
5. Спроектовано та розроблено програмне забезпечення, що реалізує технологію Question-Answering на базі матеріалів дисципліни «Програмування C++». Застосунок дозволяє ставити запитання природньою мовою, передивлятися та видаляти попередні відповіді, отримувати додаткові відповіді, що пов'язані з поставленим запитанням, та отримувати в якості відповіді фрагменти програмного коду.
6. Проведене тестування програмного забезпечення підтвердило релевантність між поставленими питаннями та відповідями, які генерує система.

15

ДОДАТОК Б

ЛІСТИНГИ ПРОГРАМНИХ МОДУЛІВ

MainWindow.py

```

import string
import sys
import stanza
from PyQt6.QtWidgets import QApplication,
QWidget, QVBoxLayout, QHBoxLayout,
QPushButton, QLineEdit, QCheckBox
from spacy.lang.uk.stop_words import
STOP_WORDS

from previousAnswers import
PreviousAnswers
from answeringWindow import
AnsweringWindow
from similarAnswers import SimilarAnswers

class MainWindow(QWidget):

    def __init__(self):
        super().__init__()
        self.get_answers_window_data = None
        self.sentence = None
        self.sentence_lemmas = []
        self.checkbox_code_status = False
        self.checkbox_similar_answer_status =
False
        self.setWindowTitle("Question-
Answering C++")
        self.setGeometry(100, 100, 600, 300)
        self.setStyleSheet("background-color:
#3c3c3c;")

        self.main_layout = QHBoxLayout()
        self.layout = QVBoxLayout()

        self.buttonQue =
QPushButton("Отримати відповідь")
        self.buttonQue.setFixedSize(150, 40)
        self.buttonQue.setStyleSheet("color:
#FFFFFF; font-size: 16px;")

        self.question_input = QLineEdit()

        self.question_input.setStyleSheet("color:
#FFFFFF; font-size: 22px;")

        self.buttonPrevious =
QPushButton("Подивитись попередні
відповіді")
        self.buttonPrevious.setFixedSize(250,
60)

        self.buttonPrevious.setStyleSheet("color:
#FFFFFF; font-size: 16px;")

        self.checkboxCode =
QCheckBox("Відповідь з програмним
кодом")

        self.checkboxCode.setStyleSheet("color:
#FFFFFF; font-size: 16px;")

        self.checkboxAnotherAns =
QCheckBox("Відповіді, що можуть вас
зацікавити")

```

```

self.checkboxAnotherAns.setStyleSheet("color: #FFFFFF; font-size: 16px;")

self.main_layout.addWidget(self.question_input)

self.main_layout.addWidget(self.buttonQue)

        self.layout.addLayout(self.main_layout)

self.layout.addWidget(self.checkboxCode)

self.layout.addWidget(self.checkboxAnotherAns)

self.layout.addWidget(self.buttonPrevious)

        self.setLayout(self.layout)

self.buttonQue.clicked.connect(self.get_answer_open)

self.buttonPrevious.clicked.connect(self.previous_answers_open)

        self.previous_answers_window = None
        self.get_answers_window = None
        self.get_similar_window = None

self.show()

def analyze_input(self):
    self.sentence = self.question_input.text()
    stanza.download('uk')
    nlp = stanza.Pipeline('uk',

```

```

processors='tokenize,mwt,pos,lemma')

        nonstopwords = 'if, else, cout, cin,
include, iostream, class, Class, return, int,
double, float, char, string, arr, array, ' \
                'arr[], array[], arr[][] , array[][] ,
switch, break, continue, while, do, for, try,
catch, default, new, delete, long, short, goto,
enum, ' \
                'typedef, sizeof, malloc, free,
struct, conlo, unsigned, stdin, stdout, stderr,
printf, scanf, file, fopen, flose, fputc, fgetc,
fwrite, fread, fseek, fprintf, fscanf,' \
                'feof, ferror, rewind, remove,
mode, null, const, inline, export, public,
private, protected, namespace, std, main,
rand'

        docum = nlp(self.sentence)
        lemmas = [word.lemma for sentence in
docum.sentences for word in sentence.words
if
                word.text.lower() not in
(string.punctuation and nonstopwords and
STOP_WORDS)]
        self.sentence_lemmas = lemmas
        print(lemmas)

def get_answers_info(self):
    sentence = self.sentence
    sentence_lemmas =
self.sentence_lemmas
        checkbox_code_status =
self.checkbox_code_status
        return sentence, sentence_lemmas,
checkbox_code_status

def get_similar_main(self):

```

```

        sentence_lemmas =
self.sentence_lemmas
        return sentence_lemmas

def get_answer_open(self):
    self.analyze_input()
    if self.get_answers_window is None:
        self.get_answers_window =
AnsweringWindow()
        self.get_similar_window =
SimilarAnswers()
        sentence, sentence_lemmas,
checkbox_code_status =
self.get_answers_info()

elf.get_answers_window.receive_answer_inf
o(sentence, sentence_lemmas,
checkbox_code_status)
        self.get_answers_window.show()
        if
self.checkboxAnotherAns.isChecked():

self.get_similar_window.get_similar =
self.get_similar_main()
        self.get_similar_window.show()
    else:
        self.get_answers_window.show()

def previous_answers_open(self):
    if self.previous_answers_window is
None:
        self.previous_answers_window =
PreviousAnswers()

self.previous_answers_window.show()
    else:

```

```

self.previous_answers_window.show()

```

```

if __name__ == '__main__':
    app = QApplication(sys.argv)
    window = MainWindow()
    sys.exit(app.exec())

```

answeringWindow.py

```

import sys
import sqlite3
from PyQt6.QtWidgets import QApplication,
QWidget, QVBoxLayout, QLabel,
QSizePolicy

```

```

class AnsweringWindow(QWidget):
    def __init__(self):
        super().__init__()
        self.checkbox_code_status = None
        self.sentence_lemmas = None
        self.sentence = None
        self.code_label = None
        self.setWindowTitle("Відповідь на
запитання")
        self.setGeometry(700, 100, 600, 250)
        self.setStyleSheet("background-color:
#3c3c3c;")

        self.layout = QVBoxLayout()

        self.question_label = QLabel()
        self.question_label.setStyleSheet("color:
#FFFFFF; font-size: 20px")

self.question_label.setSizePolicy(QSizePolic
y.Policy.Fixed, QSizePolicy.Policy.Fixed)

self.layout.addWidget(self.question_label)

```



```

self.question_label.clear()

self.layout.addSpacing(25)
self.answer_label = QLabel()
self.answer_label.setWordWrap(True)
self.answer_label.setStyleSheet("color:
#FFFFFF; font-size: 20px")

self.answer_label.setSizePolicy(QSizePolicy.
Policy.Fixed, QSizePolicy.Policy.Fixed)

self.layout.addWidget(self.answer_label)
self.layout.addStretch(1)

self.receive_answer_info(self.sentence,
self.sentence_lemmas,
self.checkbox_code_status)

def receive_answer_info(self, sentence,
sentence_lemmas, checkbox_code_status):
    self.sentence = sentence
    self.sentence_lemmas =
sentence_lemmas
    self.checkbox_code_status =
checkboxbox_code_status

    self.find_answer(self.sentence_lemmas,
self.checkbox_code_status)

self.setLayout(self.layout)

self.new_answers(self.sentence)

self.show()

def get_sentence(self):
    return self.sentence

```

```

def get_sentence_lemmas(self):
    return self.sentence_lemmas

def get_checkbox_code_status(self):
    return self.checkbox_code_status

def new_answers(self, sentence):
    self.question_label.setText(f"Питання:
'{sentence}'")

    def find_answer(self, sentences_lemmas,
checkboxbox_codes_status):
        conn =
sqlite3.connect('dbAnswersList.db')
        conn1 =
sqlite3.connect('dbQuestionAnswersList.db')
        cursor = conn.cursor()
        cursor1 = conn1.cursor()
        answer_text = ""
        cursor.execute("SELECT COUNT(*)
FROM Answers")
        res = cursor.fetchone()[0]
        for ans_id in range(res):
            cursor.execute("SELECT
keysToAnswers, answersText,
codeToAnswers FROM Answers WHERE
id=?", (ans_id,))
            # ans_id += 1
            results = cursor.fetchall()
            for row in results:
                keys_value = row[0]
                answers_value = row[1]
                code_value = row[2]
                keys_list = keys_value.split()
                print(keys_list, '\n',
sentences_lemmas, '\n', answers_value)
                if (word.lower() for word in
sentences_lemmas) == (key.lower() for key

```

```

in keys_list):

self.answer_label.setText("Відповідь:\n"+an
swers_value)
        answer_text = answers_value
        if checkbox_codes_status:
            self.layout.addSpacing(25)
            self.code_label =
QLabel("Приклад програмного коду:\n" +
code_value)

self.code_label.setWordWrap(True)

self.code_label.setStyleSheet("color:
#FFFFFF; font-size: 20px")

self.code_label.setSizePolicy(QSizePolicy.Po
licy.Fixed, QSizePolicy.Policy.Fixed)

self.layout.addWidget(self.code_label)

                self.layout.addStretch(1)
                break
            sql = "INSERT INTO
QuestionsAndAnswers (questions,
qAnswersText) VALUES (?, ?)"
            row_id = ans_id
            question_text = self.sentence
            cursor1.execute(sql, (question_text,
answer_text))

            conn1.commit()
            conn.close()
            conn1.close()

if __name__ == '__main__':
    app = QApplication(sys.argv)
    window = AnsweringWindow()

```

```

        sentence = window.get_sentence()
        sentence_lemmas =
window.get_sentence_lemmas()
        checkbox_code_status =
window.get_checkbox_code_status()
        window.receive_answer_info(sentence,
sentence_lemmas, checkbox_code_status)
        sys.exit(app.exec())

```

previousAnswers.py

```

import sys
import sqlite3
from PyQt6.QtWidgets import (QWidget,
QLabel, QPushButton, QScrollArea,
QApplication,
                                QHBoxLayout,
QVBoxLayout, QMainWindow)
from PyQt6.QtCore import Qt
from PyQt6 import QtWidgets

class PreviousAnswers(QMainWindow):

    def __init__(self):
        super().__init__()
        self.scroll = QScrollArea()
        self.widget = QWidget()
        self.layout = QVBoxLayout()

        self.setWindowTitle("Всі питання та
відповіді")
        self.setGeometry(100, 450, 600, 250)
        self.setStyleSheet("background-color:
#3c3c3c;")

        self.widget.setLayout(self.layout)

        self.label_and_button = {}

```

```

self.label_answer = []

self.insert_text()

def insert_text(self):
    conn1 =
sqlite3.connect('dbQuestionAnswersList.db')
    cursor1 = conn1.cursor()
    cursor1.execute("SELECT COUNT(*)
FROM QuestionsAndAnswers")
    num_answers = cursor1.fetchone()[0]
    k = 0
    for i in range(num_answers):
        cursor1.execute("SELECT questions,
qAnswersText FROM
QuestionsAndAnswers")
        result = cursor1.fetchall()
        h_layout = QHBoxLayout()
        question_text = QLabel(result[k][0])
        question_text.setStyleSheet("color:
#FFFFFF; font-size: 16px;")
        answer_text = QLabel(result[k][1])
        answer_text.setStyleSheet("color:
#FFFFFF; font-size: 16px;")
        answer_text.setWordWrap(True)
        delete_button =
QPushButton(f"Видалити питання номер
{k+1}")
        delete_button.setStyleSheet("color:
#FFFFFF; font-size: 16px;")

        self.label_and_button[delete_button]
= (question_text, answer_text)
        h_layout.addWidget(question_text)
        h_layout.addWidget(delete_button)

self.layout.addLayout(h_layout)

```

```

self.label_answer.append(answer_text)
self.layout.addWidget(answer_text)

delete_button.clicked.connect(lambda
checked, btn=delete_button:
self.delete_answer(btn))
    k += 1

self.scroll.setVerticalScrollBarPolicy(Qt.Scro
llBarPolicy.ScrollBarAlwaysOn)

self.scroll.setHorizontalScrollBarPolicy(Qt.S
crollBarPolicy.ScrollBarAlwaysOff)
self.scroll.setWidgetResizable(True)
self.scroll.setWidget(self.widget)
self.setCentralWidget(self.scroll)

self.show()

def delete_answer(self, delete_button):
    if delete_button in
self.label_and_button:
        question_text, answer_text =
self.label_and_button.pop(delete_button)

        question_text.setParent(None)
        delete_button.setParent(None)
        answer_text.setParent(None)

        question_text.deleteLater()
        delete_button.deleteLater()
        answer_text.deleteLater()

if __name__ == '__main__':
    app = QtWidgets.QApplication(sys.argv)

```

```
prevAns = PreviousAnswers()
prevAns.show()
sys.exit(app.exec())
```

SimilarAnswers.py

```
import sys
import sqlite3
from PyQt6.QtWidgets import (QWidget,
                              QLabel, QPushButton, QScrollArea,
                              QApplication,
                              QHBoxLayout,
                              QVBoxLayout, QMainWindow)
from PyQt6.QtCore import Qt
from PyQt6 import QtWidgets

class SimilarAnswers(QMainWindow):

    def __init__(self):
        super().__init__()
        self.sentence_lemmas = None
        self.scroll = QScrollArea()
        self.widget = QWidget()
        self.layout = QVBoxLayout()

        self.get_similar(self.sentence_lemmas)

        self.setWindowTitle("Можливо, вам
будуть цікаві ці статті")
        self.setGeometry(100, 450, 600, 250)
        self.setStyleSheet("background-color:
#3c3c3c;")

        self.widget.setLayout(self.layout)

        self.similar_list = ["if else switch case
default goto break", "cin cout printf scanf",
```

```
"for while do goto",
                        "class object main", "int
float double long short, null, unsigned,
const",
                        "string char sizeof rand",
"arr char array new delete rand", "include
iostream", "break continue",
                        "try catch default",
" class struct private public
protected typedef define inline",
                        "namespace std stdin
stdout stderr", "main return getch",
                        "file fopen flose fputc fgetc
fwrite fread fseek fprintf fscanf feof ferror
rewind remove mode",
                        ]
        self.label_and_button = { }
        self.label_answer = []

        self.insert_text()

    def get_similar(self, sentence_lemmas):
        self.sentence_lemmas =
sentence_lemmas

    def check_similar(self):
        sentence_lem = self.sentence_lemmas
        for word1 in sentence_lem:
            for word2 in self.similar_list:
                if word1 in word2:
                    return word2
        return False

    def insert_text(self):
        keys = self.check_similar()
        conn =
sqlite3.connect('dbAnswersList.db')
        cursor = conn.cursor()
```

```

cursor.execute("")
k = 0
keys = keys.split()
id_remember = []
for i in range(len(keys)):
    cursor.execute("SELECT id,
keysToAnswers, answersText FROM
Answers")
    result = cursor.fetchall()
    for res in result:
        id_ans = res[0]
        keys_ans = res[1]
        answr = res[2]
        if keys[i] in keys_ans:
            question_text = QLabel("Iipo "
+ keys[k])

question_text.setStyleSheet("color:
#FFFFFF; font-size: 16px;")
            print(res[2])
            answer_text = QLabel(answr)

answer_text.setStyleSheet("color: #FFFFFF;
font-size: 16px;")
            answer_text.setWordWrap(True)

self.layout.addWidget(question_text)

self.layout.addWidget(answer_text)

        k += 1

self.scroll.setVerticalScrollBarPolicy(Qt.Scro
llBarPolicy.ScrollBarAlwaysOn)

self.scroll.setHorizontalScrollBarPolicy(Qt.S

```

```

crollBarPolicy.ScrollBarAlwaysOff)
        self.scroll.setWidgetResizable(True)
        self.scroll.setWidget(self.widget)
        self.setCentralWidget(self.scroll)

        self.show()

if __name__ == '__main__':
    app = QtWidgets.QApplication(sys.argv)

    prevAns = SimilarAnswers()
    prevAns.show()
    sys.exit(app.exec())

dbAnswers.py
import sqlite3
import sys

import stanza
import string
from spacy.lang.uk import STOP_WORDS

class dbToAnswerPage:
    def __init__(self):
        super().__init__()
        self.conAns =
sqlite3.connect('dbAnswersList.db')
        self.conQueAns =
sqlite3.connect('dbQuestionAnswersList.db')

        self.cursorAns = self.conAns.cursor()
        self.cursorQueAns =
self.conQueAns.cursor()

        self.cursorQueAns.execute("CREATE
TABLE IF NOT EXISTS
QuestionsAndAnswers (

```

```

        id INTEGER PRIMARY
KEY,
        questions TEXT,
        qAnswersText TEXT
    )")

    self.cursorAns.execute("""CREATE
TABLE IF NOT EXISTS Answers (
        id INTEGER PRIMARY
KEY,
        keysToAnswers TEXT,
        answersText TEXT
        codeToAnswers TEXT
    )")

# self.insertText()

self.conAns.commit()
self.conQueAns.commit()

self.conAns.close()
self.conQueAns.close()

```

dbAnswers.py

```

import sqlite3
import stanza
import string
from spacy.lang.uk import STOP_WORDS
from mainWindow import MainWindow

# cursorAns.execute("""CREATE TABLE IF
NOT EXISTS Answers (
#         id INTEGER PRIMARY
KEY,
#         keysToAnswers TEXT,
#         answersText TEXT,
#         codeToAnswers TEXT
#     )")

```

```

class dbToAnswerPage:
    def __init__(self):
        super().__init__()
        self.sentence, self.sentence_lemmas,
self.checkbox_code_status =
MainWindow.get_answers_info()
        self.conAns =
sqlite3.connect('dbAnswersList.db')
        self.conQueAns =
sqlite3.connect('dbQuestionAnswersList.db')

        self.cursorAns = self.conAns.cursor()
        self.cursorQueAns =
self.conQueAns.cursor()

        self.insertText()

        self.conAns.commit()
        self.conQueAns.commit()

        self.conAns.close()
        self.conQueAns.close()

    def insertText(self):
        sentence = self.sentence
        stanza.download('uk')
        nlp = stanza.Pipeline('uk',
processors='tokenize,mwt,pos,lemma')

        nonstopwords = 'if, else, cout, cin,
include, iostream, class, Class, return, int,
double, float, char, string, arr, array, '\
        'arr[], array[], arr[][],
array[][]], switch, break, continue, while, do,
for, try, catch, default, new, delete, long,
short, goto, enum, '\

```

```
'typedef, sizeof, malloc, free,
struct, conlo, unsigned, stdin, stdout, stderr,
printf, scanf, file, fopen, flose, fputc, fgetc,
fwrite, fread, fseek, fprintf, fscanf,' \
'feof, ferror, rewind, remove,
mode, null, const, inline, export, public,
private, protected, namespace, std, main,
rand'
```

```
docum = nlp(sentence)
lemmas = [word.lemma for sentence in
docum.sentences for word in sentence.words
if
word.text.lower() not in
(string.punctuation and nonstopwords and
STOP_WORDS)]
keys_to_answers = lemmas
answer = "Цикл for повторює вказану
настанову задану кількість разів.
Настановаfor у мові програмування C++
діє практично так само, як настанова for,
визначена в таких" \
" мовах програмування, як Java
C#, Pascal і Visual Basic. Її прос-тий
формат є таким:for(ініціалізація; умова;
інкремент) настанова;У цьому записі
елемент ініціалізація " \
"є настанова присвоєння, яка
вста-новлює керівній змінній циклу
початкове значення. Ця змінна діє як
лічиль-ник, який керує роботою циклу.
Елемент умова є виразом, у якому " \
"тестуєтьсязначення керівної
змінної циклу. Результат цього тестування
визначає, вико-нається цикл for ще раз чи
ні. Елемент інкремент – вираз, який
визначає, як " \
"змінюється значення керівної
```

змінної циклу після кожної ітерації. Цикл
forвиконуватиметься доти, доки
обчислення елемента умова дає істинний
результат. " \

```
"Як тільки умова стане
помилковою, виконання програми
продовжиться з настанови, що
знаходиться наступною за циклом for."
code = "#include <iostream> //
Потокове введення-виведення\nusing
namespace std; // Використання
стандартного простору імен\nint
main()\n{\n\tint pm;" \
"\n\tfor(pm=1; pm<=100;
pm=pm+1) cout << pm << " "; \n\tgetch();
\n\treturn 0;\n}"
self.cursorAns.executemany("INSERT
INTO Answers (keysToAnswers,
answersText, codeToAnswer) VALUES ( ?,
?, ?)", (keys_to_answers, answer, code))
self.conAns.commit()
self.conAns.close()
```

```
# cursorQueAns.execute("CREATE TABLE
IF NOT EXISTS QuestionsAndAnswers (
#          id INTEGER PRIMARY
KEY,
#          questions TEXT,
#          qAnswersText TEXT
#          )")
```