Вінницький національний технічний університет
Факультет інформаційних технологій та комп'ютерної інженерії
Кафедра обчислювальної техніки

БАКАЛАВРСЬКА ДИПЛОМНА РОБОТА
на тему:
Система автоматизованого створення словника з англійської мови
ПОЯСНЮВАЛЬНА ЗАПИСКА

Виконав студент 4 курсу, групи 1КІ-18б
спеціальності 123 — Комп'ютерна інженерія
Трошенок О. О.

Керівник к.т.н., доц. каф. ОТ
Снігур А. В.
"19" 06 2022 р.

Рецензент к.т.н., доц. каф. ЗІ
Куперштейн Л. М.
"20" 06 2022 р.

Допущено до захисту
д.т.н., проф. Азаров О.Д.
"21" червня 2022 р.

ВНТУ 2022

ВІННИЦЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ

Факультет інформаційних технологій та комп'ютерної інженерії
Кафедра обчислювальної техніки
Освітній рівень — бакалавр
Спеціальність — 123 Комп'ютерна інженерія

ЗАТВЕРДЖУЮ

Завідувач кафедри обчислювальної техніки

д.т.н., проф. Азаров О.Д.

"08" 02 2022 р.

З А В Д А Н Н Я

НА БАКАЛАВРСЬКУ ДИПЛОМНУ РОБОТУ

студенту Трошенку Олександру Олексійовичу

1 Тема роботи «Система автоматизованого створення словника з англійської мови» керівник роботи Снігур Анатолій Васильович к.т.н., доцент, затверджено наказом вищого навчального закладу від «24» березня 2022 року №66

2 Строк подання студентом роботи 15 червня 2022.

3 Вихідні дані до роботи: програмний додаток, текстовий файл.

4 Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити): вступ, аналітичний огляд існуючих підходів та програмних засобів для створення або використання словників, особливості створення програмного забезпечення для автоматизованого створення словника, розробка програмного забезпечення для створення автоматизованого словника з англійської мови, тестування додатку, інструкція користувача, висновки.

5 Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень): технічне завдання, структурна схема, модель роботи, блок-схема алгоритму.

6 Консультанти розділів роботи приведені в таблиці 1.

Таблиця 1— Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1-4	Снігур Анатолій Васильович к.т.н., доцент		

7 Дата видачі завдання 12.03.2022 .

8 Календарний план виконання БДР приведений в таблиці 2.

Таблиця 2 — Календарний план

№ з/п	Назва етапів БДР	Строк виконання	Підпис
1	Постановка задачі	13.03.2022 р.	
2	Огляд існуючих рішень	15.03.2022 р.	
3	Розробка структурної схеми програми	16.03.2022 р.	
4	Розробка моделі програми	18.03.2022 р.	
5	Вибір ПЗ для моделювання	20.03.2022 р.	
6	Моделювання роботи системи	23.03.2022 р.	
7	Оформлення пояснювальної записки та ілюстративного матеріалу	02.04.2022 р.	
8	Виконання бакалаврської дипломної роботи	20.04.2022 р.	
9	Перевірка якості виконання бакалаврської дипломної роботи та усунення недоліків	01.05.2022 р.	
10	Підписи супроводжувальних документів у керівника, рецензента, нормоконтролера	15.05.2022 р.	
11	Перевірка «антиплагіат»	16.05.2022 р.	
12	Попередній захист	18.05.2022 р.	

Студент Трошенок Олександр Олексійович

Керівник к.т.н., доц. Снігур Анатолій Васильович

АНОТАЦІЯ

Пояснювальна записка містить 84 сторінки, 51 рисунок, 3 таблиці та 21 посилання.

У бакалаврській дипломній роботі досліджується система автоматизованого створення словника з англійської мови.

Головною метою створення системи автоматизованого створення словника з англійської мови є розробка програми, що буде створювати словник із тексту користувача максимально швидко та просто, мінімально залучаючи користувача до роботи із самим перекладом та записом у файл.

Ключові слова: словник, англійська мова, мова програмування, автоматизований, переклад, java.

ABSTRACT

The explanatory note contains 84 pages, 51 figures, 3 tables and 21c references.

The system of automated creation of a dictionary in English is studied in the bachelor's thesis.

The main purpose of creating an automated dictionary system in English is to develop a program that will create a dictionary from the user's text as quickly and easily as possible, minimally involving the user to work with the translation and writing to the file.

Key words: dictionary, English language, programming language, automated, translation, java.

ЗМІСТ

ВСТУП

У всі часи надзвичайно важливою проблемою для людства була можливість спілкуватися із населенням усієї планети заради обміну та отримання інформації, нових знань, торгівлі між собою. Із цього виникає ще одна проблема: необхідність розуміти одне одного, оскільки кожен народ має свою мову. Результатом стала поява словників, які містять у собі слова з двох або більше мов, у вигляді таблиці. Для зручності такі слова розташовані в алфавітному порядку для полегшення пошуку. Проте, оскільки першими словниками були паперові версії у вигляді книг, виникали декілька незручностей, таких як: необхідність купувати словники, при необхідності носити їх із собою.

Після появи комп'ютерів, з'явився компроміс у вигляді електронних словників. Такі словники являються базою даних із величезною кількістю слів, які можна оновлювати, додаючи до бази нові слова та значення. Також такі словники можуть мати набагато більшу кількість мов, використовуючи лише пам'ять пристрою, без збільшення габаритів словника. Проте, такі словники не могли стати поширеними по всьому світу без мережі Інтернет.

Із появою Інтернету збільшилися й можливості у створенні словників. Так, з'явилася можливість вільно завантажувати їх, без необхідності поширювати їх за допомогою переносних пристроїв. Також з'явилися онлайн-словники, які є такою ж базою даних, проте не вимагають знаходження програми на пристрої. Розвиток таких словників значно полегшив роботу між різними країнами світу, дозволивши кожному отримувати переклад потрібних слів швидко та просто.

Кількість опублікованих багатомовних словників за останні кілька років зросла, але вони не так популярні, як двомовні словники. Щороку завдяки вдосконаленню програмного забезпечення з'являється велика кількість електронних словників, але гарантувати правильність і правильність текстів може лише грамотний перекладач, який вміє працювати з текстами складної тематики [1].

Актуальність розробкипрограмних засобівпостійно зростає, оскільки електронні пристрої, серед яких, в основному, комп'ютер або ж смартфон,

набувають все більшого й більшого поширення. Так, якщо буквально десятиліття тому мобільні телефони використовувалися, у більшості випадків, як засіб для зв’язку, то зараз — це повноцінний засіб для обробки та зберігання інформації із величезною кількістю функцій. Саме тому поширення різноманітних програмних засобів на будь-який випадок є актуальною проблемою.

Функціонал для розроблюваного програмного забезпечення буде використовувати мову програмування Java. Java — це об'єктно-орієнтована мова програмування, яка бере свою історію з популярної мови C++. Однак, на відміну від останньої, Java є інтерпретованою мовою програмування. Написані на ній програми можуть працювати на різних платформи, незалежно від тої, на якій написана сама програма [2].

Інтерфейс програми буде виконаний на базі JavaFX. За допомогою JavaFX ви можете створювати багато типів додатків. Як правило, це мережеві програми, які розгортаються на кількох платформах і відображають інформацію у високопродуктивному сучасному інтерфейсі користувача, який містить аудіо, відео, графіку та анімацію [3].

Інтерфейс програми має бути зручним для користувача та зрозумілим. Для цього в користувача буде можливість викликати вікно допомоги із поясненням усіх функцій інтерфейсу програми. Для роботи із програмним засобам користувачу необхідно буде відкривати файли формату PDFабо TXT, або ж вписувати слова самостійно. При натисненні клавіші перекладу буде виводитися окремо слова на англійській мові, а також їхній переклад у сусідній колонці. Для користувача буде можливість записувати результат виконання програми у файл для подальшого використання із різними можливостями: для перекладу слів при створенні великої бази словника, а також для навчання та покращення навичок у перекладі англійської мови, що також є надзвичайно корисним у сучасному світі.

Об’єктомдослідження є процеси, що відбуваються в автоматизованих електронних словників, які використовують компанії або користувачі.

Предмет дослідженняє методи та засоби розробки системи для автоматизованого створення словників із англійської мови.

Мета роботи: аналіз сучасних засобів та розробка програмного додатку для створення словників із англійської мови.

Для досягнення поставленої мети необхідно реалізувати наступні **задачі**:

— проаналізувати методи та засоби для створення програмних застосунків;
— обрати програмне середовище для розробки програмного забезпечення;
— проаналізуватизасоби роботи із готовими бібліотеками;
— проаналізуватирізні можливості для створення електронних словників;
— розробити блок-схему алгоритму застосунку;
— налаштувати інтерфейс програмного засобу;
— розробити функціонал програми за розробленим інтерфейсом;
— протестувати програмний засіб, шляхом моделювання роботи користувача.

Методи дослідження. Дослідження, що були виконані в даній роботі базуються на використанні технологій розробки програмного забезпечення.

Апробація результатів бакалаврської роботи: розроблені тези для доповіді на LIнауково-технічній конференції Вінницького національного технічного університету.

Практичне значення отриманих результатів: можливість автоматизованого заповнення словника з англійської мови із текстових документів формату PDFабо TXT.

Матеріали роботи доповідались та опубліковувались [4]:

А. В. Снігур О. О. Трошенок / Проектування підсистеми автоматизованого створення словника з англійської мови // Тези доповіді. LI Науково-технічна конференція факультету інформаційних технологій та комп'ютерної інженерії. Вінниця 2022 р. Режим доступу: https://conferences.vntu.edu.ua/index.php/all-fitki/all-fitki-2022/paper/view/15378/12943

1 АНАЛІТИЧНИЙ ОГЛЯД ІСНУЮЧИХ ПІДХОДІВ ТА ПРОГРАМНИХ ЗАСОБІВДЛЯ СТВОРЕННЯ АБО ВИКОРИСТАННЯ СЛОВНИКІВ

1.1 Типи словників з англійської мовита ПЗ для їх створення

Існує декілька видів словників: електронний, паперовий та онлайн. Для паперового словника відповідають декілька переваг. Наприклад, немає необхідності в доступі до Інтернету, а також у наявності електроенергії або ж пристроїв, які її споживають. Однак, у такому випадку існує безліч недоліків. Такі словники в середньому вміщають у собі приблизно сто — сто п’ятдесят тисяч слів. Така кількість слів невідмінно переростає у величезну книгу, яку важко носити із собою та яка займає дуже багато місця (рисунок 1.1). Також такі словники хоч і вміщують дуже багато слів, проте їх не можна поповнювати новими значеннями слів, а також оновлювати їх, оскільки англійські слова часто виходять із вживання та замінюються новими. Також недоліком є ціна. Такі книжки потрібно купляти кожен раз, як тільки з’являється необхідність у новій версії книги. Окрім цього, старі словники потрібно десь зберігати, а також пошук слів здійснюється не так швидко, як в інших варіантах, навіть не дивлячись на те, що вони розташовують слова в алфавітному порядку. Такі словники мають лиши двосторонній переклад, тому їх важко використовувати для багатьох мов. Останнім недоліком є їхній вихід із моди, тому нові версії виходять рідше. Саме це робить паперові словники досить незручним варіантом, серед наявних зараз.Паперові словники створюються на фабриках на друкарських машинках, що також незручно, оскільки вимагає великої кількості обладнання для друку для забезпечення масштабного виробництва, при цьому, також накладаються логістичні проблеми із доставкою.

Рисунок 1.1 —Зображення паперового словника

Другим варіантом, який прийшов на зміну паперовим, є онлайн словник. Такі словники не вимагають постійного перенесення книги із собою, а також постійно оновлювати її користувачу. Однак, вони вимагають постійного доступу до Інтернету, який може бути не всюди. А також необхідність перенесення пристрою, який буде давати доступ до мережі. У сучасному світі майже у всіх є смартфони, які можуть забезпечувати доступ до Інтернету, тому така проблема є не такою значною, як необхідність переносити книгу. Проте, можливі перебої як у мережі, так і в доступі до сайту, що може накласти значні проблеми при необхідності отримати переклад тут і зараз. Такі словники часто оновлюються, вони безкоштовні та знаходяться у вільному доступі. У них зручний пошук завдяки інтерфейсу, який здійснюється за секунди. Недоліком також є те, що їх не можна поповнювати самому, а також створити власний словник, у якому будуть необхідні для роботи слова. Онлайн-словники по своїй суті являють собою величезну базу даних із словами та їхнім перекладом, при цьому, реалізований зручний інтерфейс для користувача. Такий спосіб розробки вимагає наявності команди, яка буде відповідати за мережеву розробку, створення бази даних, розробку інтерфейсу тощо. Проте, після їхнього створення необхідним залишиться лише підтримка додатку та оновлення бази даних. Такісловники зображено на рисунку 1.2.

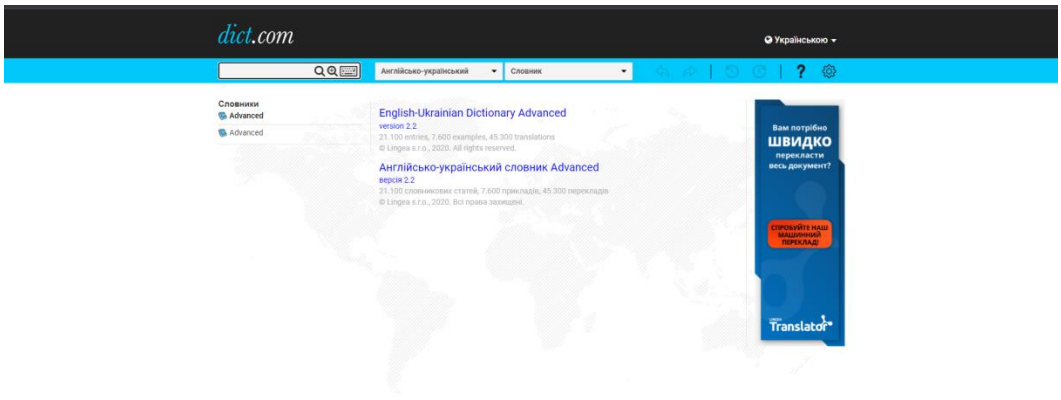

Рисунок 1.2 —Онлайн словник

Також існують й онлайн перекладачі, які надзвичайно схожі на попередній словник. Вони також вимагають доступ до інтернету та наявність пристрою, через

який буде здійснюватися робота із перекладом. Проте, вони частіше всього використовуються для перекладу одного або декількох слів, без великого об'єму, або ж великої частини тексту. Вони мають відмінність від словників у тому, що в них не так часто приводяться приклади використання слів, або ж декілька тлумачень таких слів. Однак, зараз такої різниці майже немає, оскільки такі словники постійно оновлюються. Найпопулярнішим перекладачем є Google Translate (рисунок 1.3), який масово використовуються завдяки своїй зручності та постійному оновленні слів. Також у такому перекладачу існує можливість перекладу одразу на безліч мов, тому він і набув такої масовості. Проте, їх не можна поповнювати самому, а також вони вразливі до перебоїв із мережею як зі сторони користувача, так і з сторони сервісу. Розробляються такі системи схожим чином, як і онлайн-словники, проте вони вимагають наявність алгоритму для підстановки правильного положення слів, їхнього значення, відмінювання тощо. Здебільшого, такі системи розробляються великими командами спеціалістів.

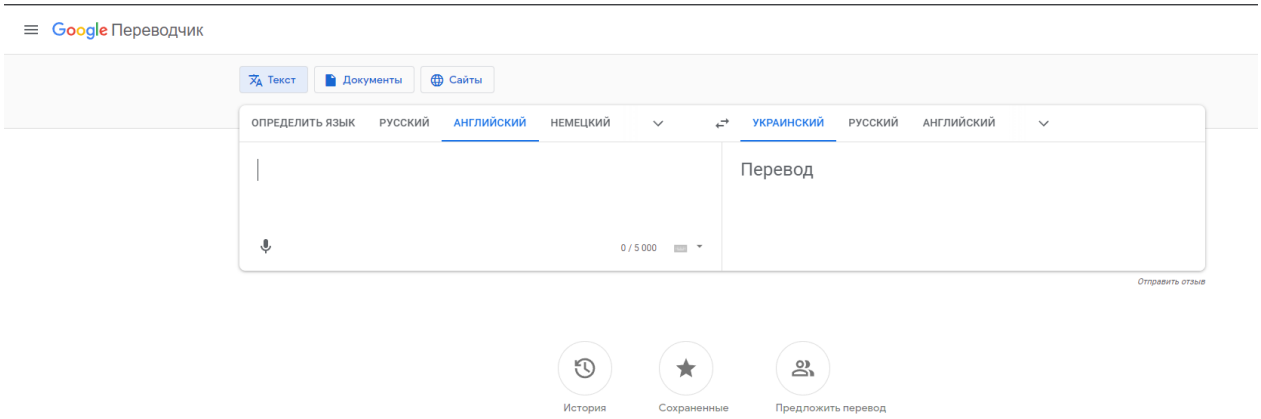

Рисунок 1.3 —Онлайн перекладач Google Translate

Останнім варіантом словників є електронні. Вони можуть використовуватися без доступу до інтернету, а також у них немає перебоїв у доступі до сервісу як під час технічних проблем, так і під час оновлення обладнання. Проте вони вимагають вимагають наявність пристрою, через який буде здійснюватися доступ до програми, а також займають місце в пам'яті комп'ютера, хоч і зараз це не така значна

проблема, оскільки вони не вимагають багато місця, а сучасні комп’ютери мають досить пам’яті. Такі словники можуть бути як безкоштовними, так і платними на основі постійної купівлі, так і підтримуватися за рахунок підписки. Приклад електронного словника зображено на рисунку 1.4. Головною перевагою таких словників є можливість завантажувати готові словники, або ж створювати свої. До них також можна додавати свої слова. Такі словники мають широкий функціонал для пошуку слів, а також можуть налаштовуватися під користувача, що робить їх надзвичайно зручними у використанні як для власного використання, так і для користування на великих підприємствах чи закладах освіти. Електронні словники схожі із онлайн-словниками, проте вони знаходяться безпосередньо на пристрої користувача, та є великою базою даних слів із інтерфейсом для їхнього виведення, що розроляється на різних мовах програмування, які будуть зручні для програмістів. Такі словники потрібно оновлювати для поповнення бази даних, а також виправлення системних порушень, тому потребують періодичного доступу до мережі Інтернет.

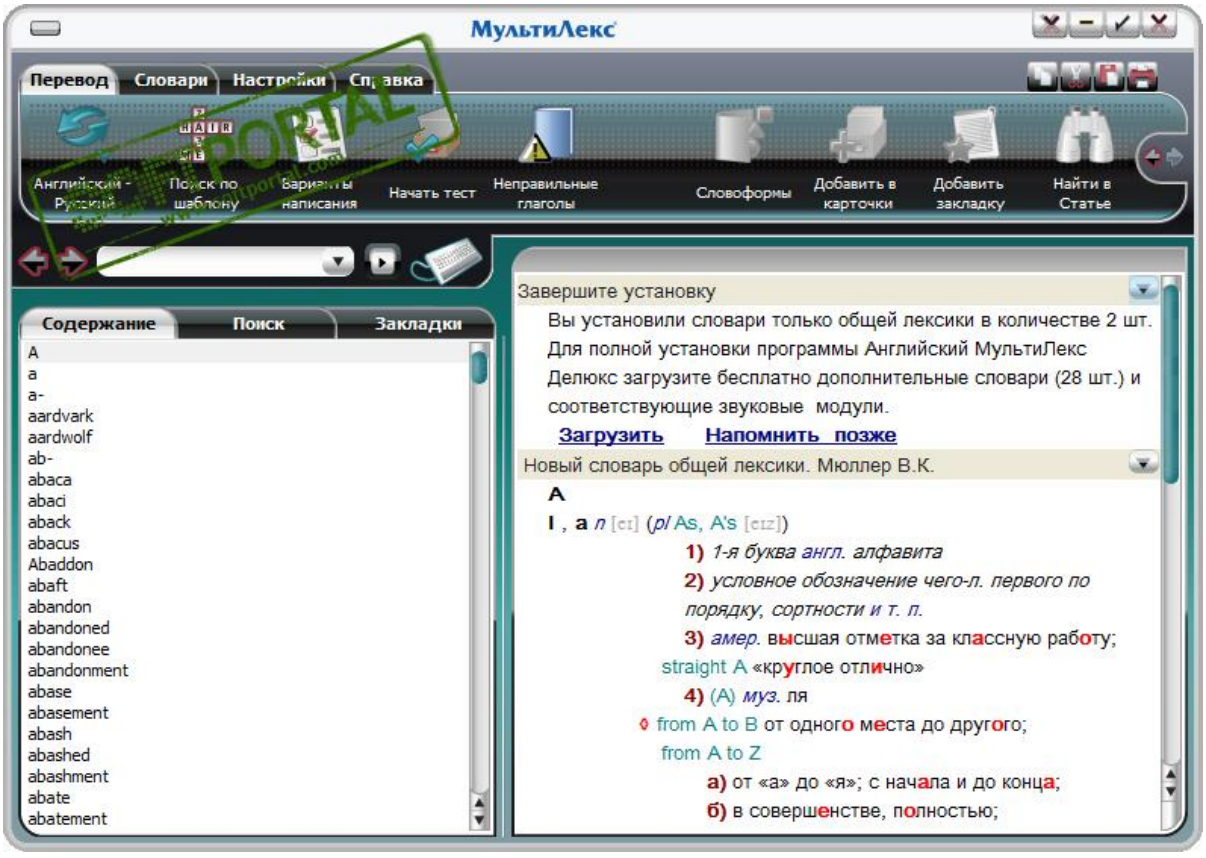

Рисунок 1.4 —Приклад електронного словника

1.2Основні параметри створюваних словників

Головними вимогами для словників є наявність великої кількості слів, щоб надати користувачам широкий спектр можливостей до перекладу. Словники мають постійно поповнюватися як новими словами, так і новими значеннями вже існуючих слів. Інтерфейс як програми, так і онлайн сервісу має бути зручним та зрозумілим для користувача, щоб полегшити роботу із ним, а також поширення словника серед широкої аудиторії.

Словник має бути доступний майже завжди та бути стійким до перебоїв із мережею Інтернет, або ж у роботі із онлайн сервісом. У випадку Google Translate для платформи Android є можливість завантажити базу даних зі словами для доступу до перекладу в оффлайн режимі. У випадку із електронною версією програми, вона має постійно оновлюватися для підтримки різних платформ, а також до оновлень цих самих платформ, для яких розроблюється дане програмне забезпечення.

1.3 Порівняльний аналіз існуючихсистем, що створюють або використовують словники з англійської мови

Порівняння програм для створення чи використання словників із англійської мови, наведено у таблиці 1.1.

Актуальність вивчення іноземної мови напряму пов’язане із створенням відповідних програмних словників. На сьогоднішній день є відносно велика кількість програм-словників, які відрізняються між собою здебільшого зручністю інтерфейсу для користувача. У той же час створення словника з окремих файлів або з навчальних матеріалів для вивчення іноземної мови може потребувати розробки нового програмного забезпечення, що має відповідні функції та надає ще більше зручності користувачу при створенні та користуванні таким словником. Для цього проаналізуємо існуючі програми-словники та надамо рекомендації для створення

більш зручних таких словників, що можуть формуватися з окремих файлів та відповідного навчального матеріалу.

Таблиця 1.1—Порівняльна характеристика аналогів із програмою, що розробляється

Характеристика	Duolingo	English Galaxy	Ходячий словник	DictMaster	BX Language acquisition	Розроблюване ПЗ
Запис слів із документів формату PDF або TXT	-	-	-	-	-	+
Створення користувацького словника	-	+	+	+	+	+
Можливість запису слів із клавіатури	-	-	-	-	+	+
Простота використання	+	-	+	-	-	+

На сьогоднішній день існує певна кількість програм для створення словників та роботи із ними. При цьому словники можуть складатися особисто користувачем вручну, заповнюватися автоматично модулем системи вивчення іноземної мови як словник індивідуального вивчення, а також формувати словник, наприклад, з документу Word.

Так, зокрема, система «Duolingo», при вивченні слів іноземної мови дозволяє формувати список із невивчених слів —індивідуальний внутрішній словник для подальшого повторення, хоч і не доступний для користувача напряму. Інтерфейс програми зображено на рисунку 1.5. Теми для вивчення формуються на основі тесту, який користувач проходить перед початком роботи із програмою. Далі, після визначення результатів, формуються теми для вивчення, які будуються у вигляді дерева із наростанням складності слів. Слова також об’єднуються за тематикою,

наприклад «тварини», «люди», «сім’я» тощо. Для мотивації користувача введено рейтинг для підтримки конкуренції, яка формується основі ігрового досвіду. Тобто, чим більше тем проходить користувач, тим вище він у рейтингу. Також враховується стабільність вивчення. Такі рейтинги дійсно допомагають підтримувати заохочення користувача до навчання, щоб увести вивчення у звичку. Необхідно зазначити, що в додатку, окрім англійської, існує також інші існуючі мови, вивчення яких проходить у такій самій системі [4].

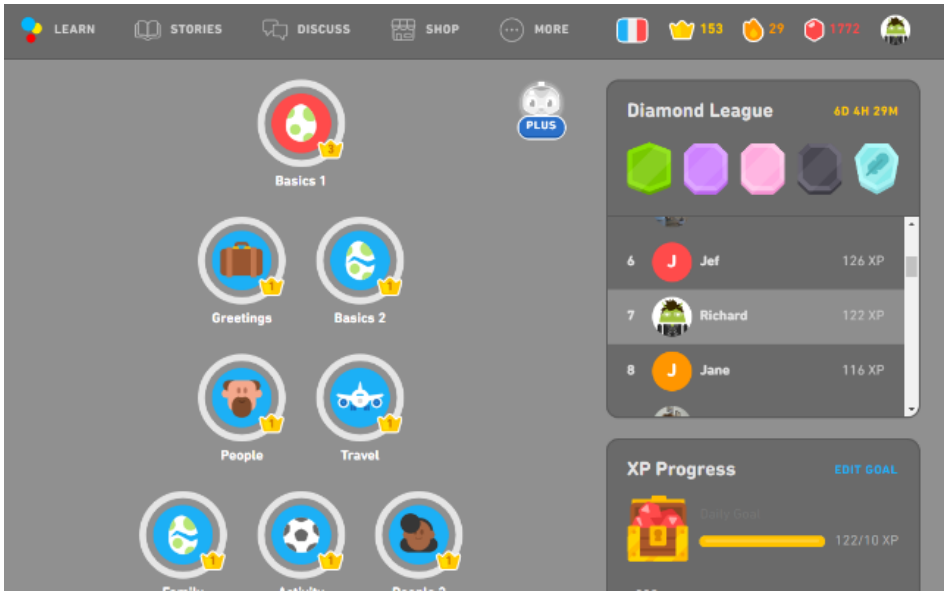

Рисунок 1.5 — Інтерфейс програми «Duolingo»

Більш розширені можливості по створенню словника надає програма «English Galaxy». Вона дозволяє створювати користувацький словник, куди будуть додаватися слова, які обере користувач, із перекладом та транскрипцією, а також із можливістю прослуховування правильного звучання. Ця програма теж створена у якості додатку для навчання користувачів англійської мови. Інтерфейс програми зображено на рисунку 1.6. У цьому додатку також є можливість учити мову за допомогою різних тем, які формуються на основі рівню знання англійської мови. На відміну від попередньої програми, цей додаток розширює самостійність вивчення, оскільки дозволяє самому записувати слова для вивчення, а також у ній існує безліч збірників із темами, які слугують підручниками, тільки в електронному форматі.

Також у базі програми є книжки на англійській мові, які доступні для користувача. У них він може прослуховувати слова, запам’ятовуючи їхнє звучання, а також здійснювати функції перекладу невідомих слів. У програмі існують тести, які визначають рівень мови після проходження навчання. Також при навчанні використовуються зображення слів у відповідній тематиці для покращення запам’ятовування, забезпечуючи тим самим стимуляцію асоціативної пам’яті, окрім слухової та зорової [5].

У якості прикладу можна навести програмне забезпечення «Ходячий словник». Ця програма не має функції перекладача, але дозволяє змагатися з іншими користувачами в знанні слів і створювати власний словник. Це гарний приклад додатку, який дозволяє покращувати знання в мові шляхом обміну досвідом із іншими користувачами в мережі Інтернет. Інтерфейс програми зображено на рисунку 1.7 [6].

Рисунок 1.6 — Інтерфейс програми «English Galaxy»

Рисунок 1.7— Інтерфейс програми «Ходячий словник»

Створювати словники та редагувати їх дозволяє програма «DictMaster». Цей додаток надає користувачу велику кількість функцій у формуванні словника. Наприклад, формування пустого словника, для подальшого його наповнення словами, які користувач вважає за потрібне записати. Окрім цього, дане програмне забезпечення дозволяє завантажувати готові словники з документів формату txt та word для їхнього доповнення та редагування. У програмі є також налаштування роботи під себе. Інтерфейс програмного забезпечення зображено на рисунку 1.8 [7].

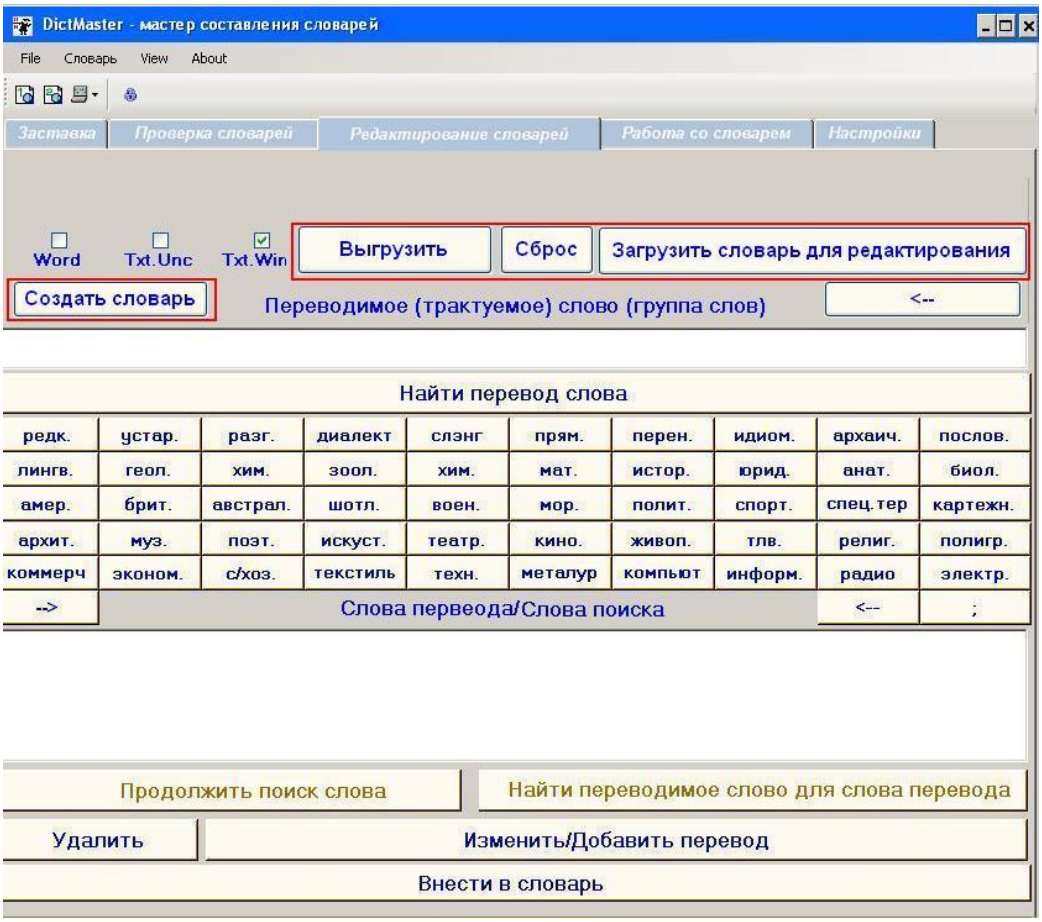

Рисунок 1.8— Інтерфейс програми «Dictmaster»

Програма «BX Language acquisition» дозволяє створювати свої словники вручну за допомогою редактора або автоматично на основі частотного аналізу тексту. Ви можете розділяти слова у словнику за категоріями, прив'язувати приклади використання слів, встановити порядок вивчення словника. BX Language acquisition — умовно-безкоштовна програма-оболонка, призначена для заучування написання та вимови іноземних слів. Заучування лексичних одиниць (ЛО) у програмі ґрунтується на накопиченні статистики відповідей на завдання програми. За кожне правильно виконане завдання нараховується певна кількість балів. За кожну помилку знімаються бали. У кожній наступній вправі переважають завдання з найменшим балом, кілька нових завдань, а також завдання на повторення вивченого матеріалу. При наборі певної кількості балів завдання вважається умовно вивченим. Умовно вивчені слова пропонуються для повторення за методикою Еббінгауза.

Користувачі мають можливість самостійно налаштувати систему балів та повторення вивченого матеріалу. Інтерфейс програми зображено на рисунку 1.9 [8].

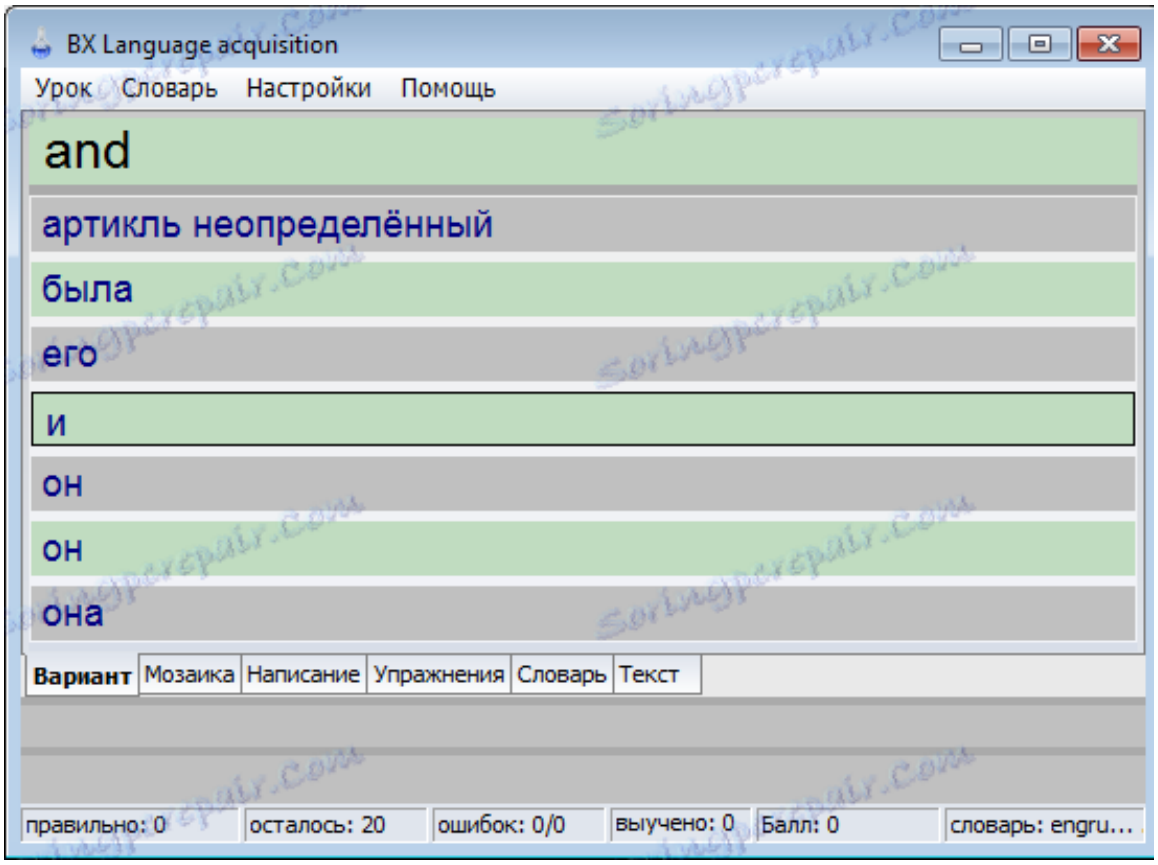

Рисунок 1.9 — Інтерфейс програми «BX Language acquisition»

2 ОСОБЛИВОСТІ СТВОРЕННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ АВТОМАТИЗОВАНОГО СТВОРЕННЯ СЛОВНИКА

2.1Основні вимоги до створення електронних словників

Електронний (автоматичний) словник — це словник у спеціальному машинному форматі, що функціонує як частина програмного забезпечення комп'ютера. Сьогодні широко розповсюджуються електронні версії найрізноманітніших словників. На відміну від традиційних словників, електронний словник поряд з текстом та графічними зображеннями може містити весь спектр медіа-об'єктів, включаючи відео або анімаційні фрагменти, звук, музику та інше.

Важливою особливістю електронного словника є гіпертекстовий пристрій. Посилання, введені в слова, фрази або малюнки, дозволяють користувачеві вибрати текст або малюнок та негайно вивести на екран пов'язані з ним відомості та мультимедійні матеріали. Взаємини між компонентами словникової статті є лінійними. Словникова стаття має чітку логічну структуру із ієрархічними зв'язками між елементами. Кожна інформаційна категорія займає тут суворо фіксоване місце — так звану «зону». Користувач, виявляючи інтерес до тієї чи іншої інформації, запитує певний параметр та отримує доступ до окремих фрагментів статті. Відповідно до запиту активізується лише окрема зона, тому немає необхідності переглядати всю статтю. Отже, творці електронного словника можуть передбачити досить багато словникових входів, що дозволяє користувачеві легко і швидко отримувати будь-яку необхідну йому інформацію, а проблеми алфавітного розташування словникових статей для масового користувача немає принципового значення.

Електронні словники мають значні переваги перед паперовими, що відбивається на швидкому зростанні ринку. Електронні словники можуть обійти ключове протиріччя книжкових словників: чим більше інформації надає словник і чим розвиненіші його наукові інструменти, тим складніше ним користуватися. Тому класичні словники поділяються на дві категорії. Перший більш популярний, зручніший, але також дуже простий. Другий – це детальні академічні публікації, які

завжди дозволяють швидко отримати потрібну інформацію. Зараз, такі словники можуть не тільки перевищувати обсяги слів паперових словників, а й мають функцію пошуку слів, що дозволяє знаходити слова за дуже короткий час [9].

При створенні словників із іноземних мов враховуються різні фактори. Наприклад, необхідність постійно поповнювати базу слів новими, а також додавати нові значення до вже існуючих слів. У них має бути зручний та зрозумілий інтерфейс для того, щоб користувач завжди знав, для чого йому та чи інша функція, а також мати можливість вивести на екран програми допомогу, де можна буде знайти необхідні функції та їхнє пояснення.

Також така програма має бути доступною для великої кількісті пристроїв та мати підтримку різних платформ для поширення серед користувачів різних категорій. Додаток має мати низькі вимоги до платформи, щоб поширити швидкодію програми й позбавити програму від перебоїв та підвисань.

Програма має бути кросплатформенною для забезпечення роботи як на комп’ютерах у системі Windows, так для Linux систем і смартфонів на платформі Android. Також можливе створення окремих програм для різних систем, хоч це і не зручно та вимагає значної кількості роботи як у розробці, так і в підтримці наявних додатків.

Для електронних словників буде плюсом можливість працювати із документами, як і у випадку зчитування тексту, так і в можливості запису. Буде перевагою й розбиття слів за категоріями для полегшення пошуку слів користувачем у ручну, а також запису нових слів. Також це буде корисним для людей, які забажають вивчати мову в тому числі і за допомогою словника. Словники мають мати змогу перекладати й велику кількість слів, оскільки можуть використовуватися у виробництві.

2.2 Особливості використання електронних словників

Головними варіантами користування електронними словниками є можливість використовувати їх на виробництвах, як великих, так і малих, а також їхнє

використання із цілями навчання дитини, або ж дорослого. Крім цього, такі словники можуть використовуватися в якості бази даних для іншого застосунку або веб-сайту. Розглянемо детальніше кожний варіант використання.

Електронні словники дуже корисні для навчання дітей, дорослих іноземній мові. Мови можуть містити безліч слів та постійно оновлюватися, не вимагаючи від користувача постійно купувати новий паперовий варіант. Для батьків це зручно, оскільки вони можуть самостійно заповнювати та редагувати слова в словнику, обираючи теми для вивчення на основі знань їхньої дитини та особистого досвіду. Самі ж дорослі також можуть використовувати такі словники, заповнюючи їх за власним бажанням. Такі словники мають величезну перевагу для вчителів та викладачів, оскільки дозволяють поширювати їх для великої аудиторії, оскільки їм необхідно лише налаштувати розсилання для учнів, контролювати їхні знання та самостійно обирати теми. Для шкіл, університетів або ж приватних закладів освіти це також корисно, адже такі словники можуть поширюватися на основі підписки, часто із можливістю окремого договору для закладів, або ж взагалі безкоштовно, що дозволяє зекономити величезну кількість коштів на закупівлі та зберіганні великої кількості книг. Така особливість також важлива для учнів, оскільки дозволяє зменшити необхідну кількість книг, які треба носити до закладів освіти.

Для приватних підприємств, що працюють із іноземними мовами та перекладами, електронні словники також мають перевагу над паперовими. Вони можуть міститися на серверах, що дозволяє користуватися ними всім працівникам комп’ютерної мережі. Такі словники захищені від можливих збоїв Інтернету, дозволяючи працювати навіть за відсутності мережі. Електронні словники мають зручний пошук, який забезпечується або ж програмою, у якій вони знаходяться, або ж самими засобами операційної системи. Також, при наявності спеціального відділу або окремого працівника, відповідального за цю роботу, словники можуть формуватися також із врахуванням особливостей роботи підприємства, заповнюючи їх необхідними слова, які можуть бути професіональними, або ж сленговими.

Для створення програмних засобів, або ж веб-застосунків електронні словники також можуть використовуватися та бути найкращим варіантом. Так, такі словники можуть слугувати базами даних для програм, які будуть використовувати переклад. Наприклад, додатки для перекладу тексту, який можна використовувати офлайн. Вони будуть формувати текст для виводу на основі зчитування слів із електронної бази та формування речень із них. Також такі бази даних можна використовувати для комп'ютерних ігор, де речі, фрази чи імена персонажів можуть перекладатися, беручи слова із електронного словника, хоч такий переклад важко назвати професіональним. Для веб-застосунків такі словники також можуть бути корисними. Наприклад, для пошуку якоїсь інформації. Так, користувач може ввести в пошук запит на одній мові, а сайт виведе результати, враховуючи також й інші мови.

Для полегшення розуміння задачі, створимо структурну схему програми, яка дозволить розбити завдання на модулі, які необхідно буде розробити та об'єднати в одну робочу програму. Дана схема зображена на рисунку 2.1.

Рисунок 2.1 — Структурна схема додатку

2.3 Аналіз особливостей використання розроблюваного додатку

Особливістю розроблювального застосунку є можливість користувачу самому створювати словник із англійської мови. Так, програма буде самостійно перекладати слова користувача та записувати їх у текстовий словник, який він зможе використовувати за власними бажаннями, наприклад, для навчання себе або когось іншого, у випадку роботи викладачем або одним із батьків дитини. Також програма

може використовуватися для заповнення необхідних слів для підприємств або закладів.

Користувач зможе перекладати слова, які будуть вводитися вручну з клавіатури або ж сформовані із тексту різних форматів. Це особливо зручно для навчання учнів, оскільки для них часто задають переклад текстів, які в сучасному світі та поширення електронних підручників. Такий додаток зробить словник із заданих слів, дозволяючи полегшити їхнє вивчення. Після запам’ятовування слів можна створити новий словник із вибором найважчих для вивчення. Також це зручно для викладачів, так як вони можуть формувати слова для вивчення по темах, та подальшого поширення між учнями. Сформовані словники можна також роздрукувати та роздати учням у паперовому варіанті.

Для підприємств така програма також буде корисна, оскільки дозволить формувати словники для використання із врахуванням особливостей роботи. Це також корисно для підготовки персоналу, роздаючи їм матеріал для вивчення, оскільки для кожного підприємства є документація та технічні особливості, а із поширенням використання англійської мови це також буде корисним.

Сформуємо модель використання додатку, що буде моделювати роботу користувача для правильної постанови задачі та розробки програми. Схема зображена на рисунку 2.2.

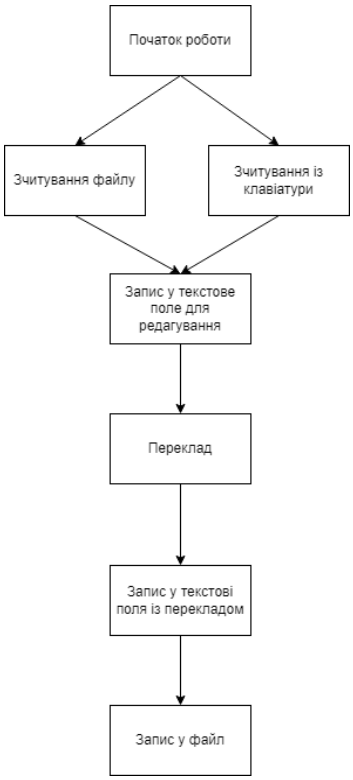

Рисунок 2.2 — Модель використання додатку

2.4Інтегроване середовище розробки IntelliJ Idea

IntelliJ IDEA —це інтегроване середовище розробки (IDE) для мов JVM, яке призначене для максимізації продуктивності розробників. Воно виконує рутинні та повторювані задачі для вас, забезпечуючи коректне завершення коду, статичний аналіз коду та рефакторинг і дозволяє зосередитись на позитивному боці розробки програмного забезпечення, роблячи його не лише продуктивним, а й зручним досвідом. IntelliJIdeaпідтримує розробку додатків на таких мовах програмування, як: Java, Kotlin, Scala, Groovy. Також, за допомогою плагінів можна додати підтримку більшість найпопулярніших мов програмування, які існують на даний момент. При цьому, кількість нових плагінів для підтримуваних мов постійно збільшується [10].

IntelliJIdeaпоширюється розробниками у трьох версіях: IntelliJ IDEA Community Edition, IntelliJIDEAUltimate, а також IntelliJ IDEA Edu. Community-tyEditionпоширюється безкоштовно та має всі необхідні для розробника програмного забезпечення, а також студента або ж людини, яка навчається самостійно. Ця версія використовується для розробки як для JVM, так і для

Androidпрограм. У таблиці 3.1 наведено системні вимоги щодо використання даного інтегрованого середовища. Ultimateверсія поширюється на основі платної підписки із врахуванням знижки за другий та третій роки користування. Ця версія середовища розробки містить у собі всі наявні функції безкоштовної версії, а також, окрім них, підтримку величезної кількості фреймворків для серверної та front-end розробки, застосунків для серверів, а також інтеграцію із базами даних та безліч інструментів. Версія Educлугує як середовище для вивчення та тренування навиків роботи із мовою Javaй самим середовищем розробки. Дана версія містить вбудовані уроки по Java, спеціальні інтерактивні завдання, а також інструменти для вчителів задля розробки власних курсів навчання учнів. Ця версія також поширюється на безкоштовній основі [11]. Також необхідно розуміти, що дана програма має свої системні вимоги для правильної роботи без підвисань, вилетів та помилок. Також при недостатній потужності системи або відсутності необхідних компонентів, середовище може не працювати взагалі.

Таблиця 2.1—Офіційні системні вимоги для IntelliJIdea

Вимога	Windows	Linux	macOS
Операційна система	Windows 8 або новіші версії для 64-бітних версій	Дистрибутив із підтримкою Gnome, KDE або UnityDE	macOS 10.14 або новіші
Оперативна пам'ять	Мінімально 2 ГБ вільної оперативної пам’яті, рекомендовано 8 ГБ загальної пам’яті		
Процесор	Будь-який сучасний процесор		
Дисковий простір	2.5 ГБ вільного простору із додатковим 1 ГБ для кешів. Для кращої роботи SSD із мінімум 5 ГБ вільного простору		
Роздільна здатність монітору	Мінімальна: 1024×768, рекомендована: 1920×1080		

Версія JDK	Вам не потрібно встановлювати Java для запуску IntelliJ IDEA, оскільки JetBrains Runtime входить в комплект із IDE (на основі JRE 11). Однак, для розробки Java-додатків потрібен окремий JDK.

Після завантаження програмного забезпечення та його налаштування, яке можна виконати як у прискореному режимі із вибором стандартних налаштувань, або ж із вибором із великої кількості тем та налаштувань, якими можна побудувати середовище для власної зручності. Після налаштування нам відкриється вікно із можливістю створити новий проект, або ж відкрити вже готовий. Після натиснення на клавішу «Новий проект», відкриється вікно налаштування, яке зображене на рисунку 2.3.

Перед нами буде вибір між різними мовами програмування, можливістю вибрати версію JavaDevelopmentKittощо. Після натиснення на клавішу Create, перед нами відкриється саме середовище розробки із програмним кодом, приклад якогось зображено на рисунку 2.4.

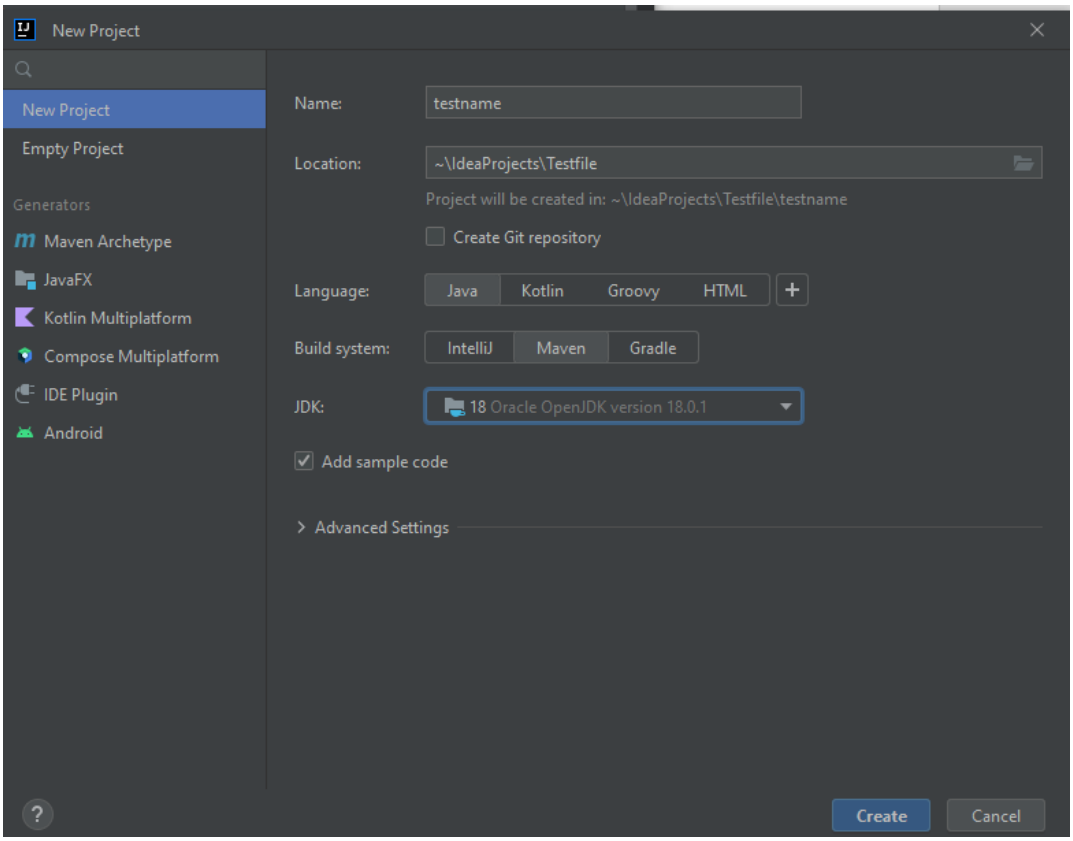

Рисунок 2.3 —Вікно створення нового проекту

Рисунок 2.4 —Приклад створеного нового проекту

Після створення проекту ми можемо переходити до роботи. Також IntellijIdeaпропонує безліч зручностей для розробники, серед яких є гарячі клавіші, підказки для введення коду, підсвічення помилок, а також дуже зручна функція введення великих шаблонних блоків коду за допомогою скороченого введення абревіатур, як показано на рисунку 2.5.3 лівого боку буде введення скороченого введення, а з правого — результат.

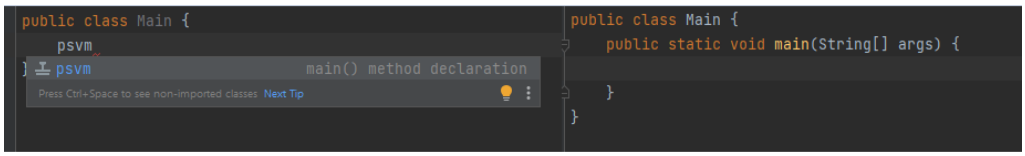

Рисунок 2.5 —Приклад скороченого введення

У розроблювальному додатку буде використовуватися JavaFX. JavaFX—це назва сімейства технологій для розробки візуально-насичених додатків для різних пристроїв. APIJavaFXмають радикально інший спосіб обробки графіки, відомий як Immediate mode, зміщення фокусу від піксельного режиму (як у бібліотеці Java2D, що використовується Swing), доструктурованого підходу, який робить анімацію чистіше і легше. Для ImmediatemodeAPI є процедурним. Щоразу, коли малюється новий кадр, програма безпосередньо видає команди малювання. Графічна бібліотека не зберігає сцени між кадрами. Натомість, програма відстежує саму сцену [12]. В основі JavaFX є нова мова програмування JavaFX Script, побудована з нуля для

моделювання та анімації мультимедійних програм. JavaFX Scriptскомпільований і об'єктно-орієнтований, з синтаксисом, що не залежить від Java, але здатним працювати з файлами Java класу. Водночас JavaFX Script (мова) та JavaFX (API та інструменти) створюютьсучасний, потужний та зручний спосіб створення програмного забезпечення [13].

Для початку роботи із JavaFXпри виборі нового проекту необхідно обрати JavaFXу лівій частині відкритого вікна, як зображено на рисунку 2.6. Там ми можемо обрати на основі якої системи буде побудований проект, мову програмування, назву проекту, його розміщення в провіднику, версію JDKтощо.

Далі, після створення, відкривається сам проект із чотирма різних файли, які необхідні для розробки програмного застосунку. Вони показані на рисунку 2.7.

Файл pom (ProjectObjectModel) —цеспеціальний файл XML, який завжди зберігається в базовій директорії проекту. Цей файл містить інформацію про проект та різні деталі конфігурації, які використовуються для створення проекту. Він також містить різні завдання та плагіни [14]. Далі, у файлі hello-view.xmlзнаходиться розмітка та конфігурація інтерфейсу програми таких елементів, як: клавіші, блоки тексту, або ж TextField чи TextArea, налаштування меню, CheckBoxтощо. При конфігурації ми можемо обирати ширину та висоту елементу, його підпис, позицію на інтерфейсі, а також IDдії при натисканні або взаємодії із елементом. Наступним файлом є контролер, який на рисунку 2.7 підписаний як HelloController. Він має розширення .javata містить у собі головні налаштування проекті, такі як: функції елементів, дії при натисканні клавіш, об’явлення змінних та їхня обробка. Останнім необхідним нам елементом є HelloApplication, який для завантаження та виведення самої програми та її інтерфейсу. Саме він міститиме функцію main, із якої буде відбуватися запуск програми. Після цього можна приступати до роботи над проектом.

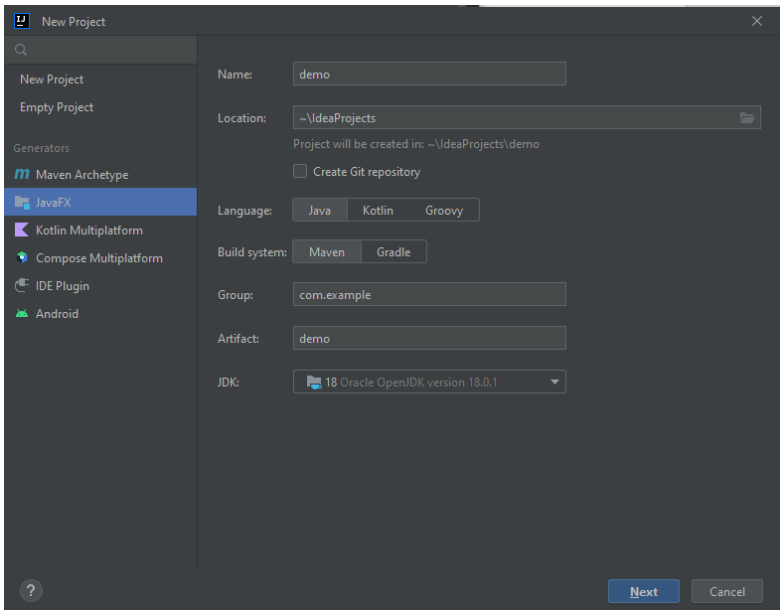

Рисунок 2.6 —Створення проекту на базі JavaFX

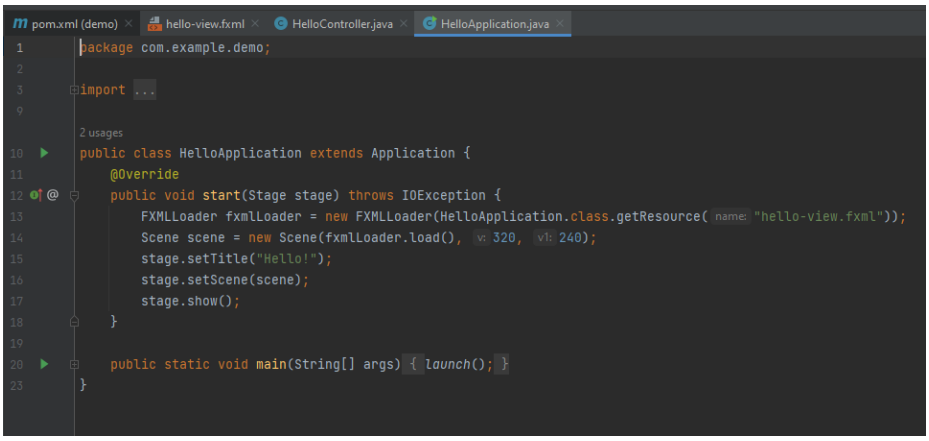

Рисунок 2.7 —Результат створення проекту

3 РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ СТВОРЕННЯ АВТОМАТИЗОВАНОГО СЛОВНИКА З АНГЛІЙСЬКОЇ МОВИ

3.1Налаштування інтерфейсу. JavaFXSceneBuilder

При розробці інтерфейсу необхідно враховувати декілька важливих моментів. Так, інтерфейс має бути максимально зручним та зрозумілим для користувача, який може бути як досвідчений користувачем програмних забезпечень різних типів, через що не буде стикатися із проблемами в користуванні, або ж користувач, який не є досвідченим користувачем, через що може стикатися із низкою проблем. Для

досвідченого користувача інтерфейс має бути інформативним та із максимальною доступністю до функцій програми, задля підвищення продуктивності та швидкості роботи. Для користувача, який не так часто користується комп'ютером, або ж йому потрібна лише певна програма, інтерфейс має бути максимально зрозумілим, щоб він без зайвих пошуків міг знайти необхідну йому функцію. При можливості, можна надати клавішу допомоги, яка стисло пояснить користувачу необхідні функції програмного забезпечення без необхідності шукати їх у довідці користувача.

Можливості JavaFXдозволяють розробити такий інтерфейс, де не буде нічого зайвого із збереженням повного функціоналу. Хоча й, при бажанні, можна сконструювати програму із урахуванням будь-яких бажань розробника щодо дизайну. Проте, стосовно самої проектування програми зі сторони програміста, налаштування .xmlфайлу може бути не зручним, якщо розробник не має необхідного досвіду. На рисунку 3.1 показаний приклад інтерфейсу для нового проекту.

Рисунок 3.1 —Приклад створеного нового hello-view.xml

Так, досить незручно налаштовувати об'єкти інтерфейсу за допомогою простого коду. Для перевірки розташування та подальшого вирівнювання елементів треба буде потратити досить багато часу. Саме для цього існує програмне забезпечення для полегшення роботи. Для виконання завдання використовується програмне забезпечення SceneBuilder. JavaFXSceneBuilder—це інструмент для візуальної розробки, що дозволяє користувачам швидко створювати інтерфейси програм JavaFX без кодування. Для користувачівє функція для

перетягуваннянаявних елементів інтерфейсу в область програми, а також функціяналаштування та конфігурації їхніх властивостей. Користувачі можутьзмінювати стиль елементівта записувати код FXML для макету, який вони створюють та автоматично генерують у фоновому режимі. Результатом є FXML-файл, який потім може бути об'єднаний з проектом Java шляхом прив'язки інтерфейсу користувача до логіки програми [15].

На рисунку 3.2 зображений приклад використання SceneBuilder, що інтегрований безпосередньо в саме середовище розробки IntellijIdea.

Проте, максимально зручний функціонал наявний саме при використанні окремої програми. На рисунку 3.3 зображене головне меню програмного забезпечення, де можна створити новий файл xml, або ж відкрити вже готовий. У програмі є пошук, для полегшення роботи в середовищі, елементи легко пересуваються та правильно вирівнюються за допомогою вбудованих інструментів. При цьому, усі зміни автоматично заповнюють програмний код, полегшуючи роботу розробника. На рисунку 3.8 показаний приклад роботи вже в самій програмі.На ньому видно надзвичайно великий функціонал програми, який дозволяє повністю налаштовувати інтерфейс, прописуючи код від встановлення положення самих елементів, до їхньої конфігурації із заданням функцій та властивостей.

Саме через зручність та легкість у розробці була обрана ця програма. На рисунку 3.4 зображено готовий інтерфейс програми, що побудований на основі необхідних вимог до таких програмних засобів.

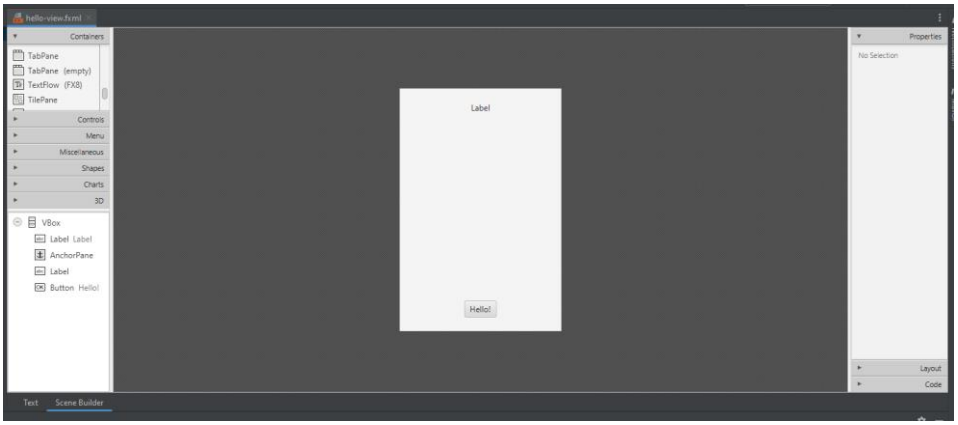

Рисунок 3.2—Scene Builder у середовищі Intellij Idea

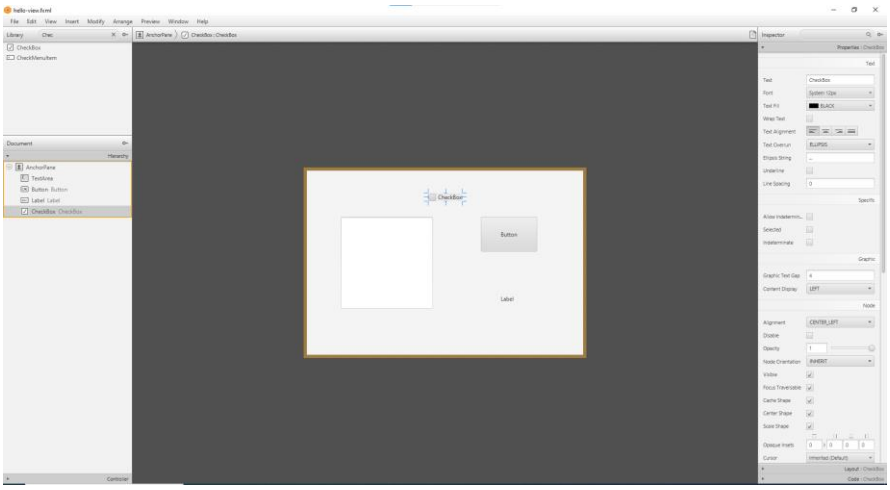

Рисунок 3.3—Робота в програмі SceneBuilder

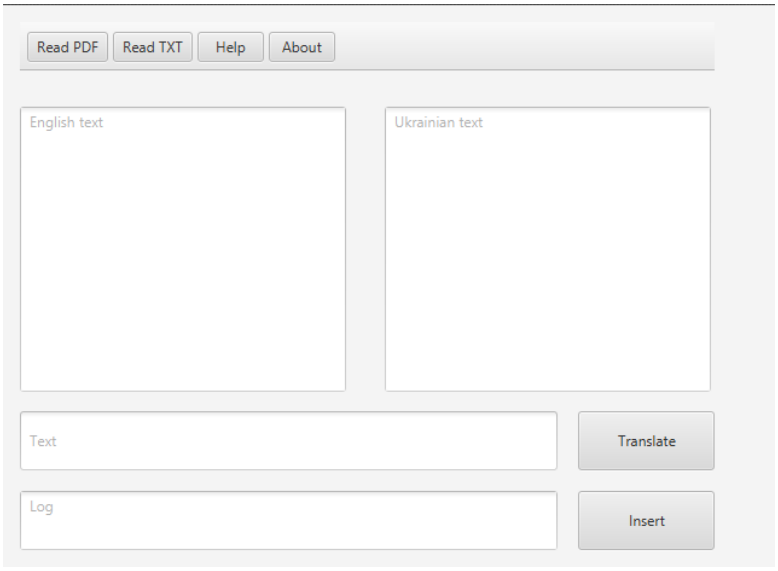

Рисунок 3.4—Інтерфейс розроблюваного програмного засобу

На рисунку 3.5 зображений вигляд змісту файлу hello-view.xml, де видно результат роботи програмного забезпечення SceneBuilder.

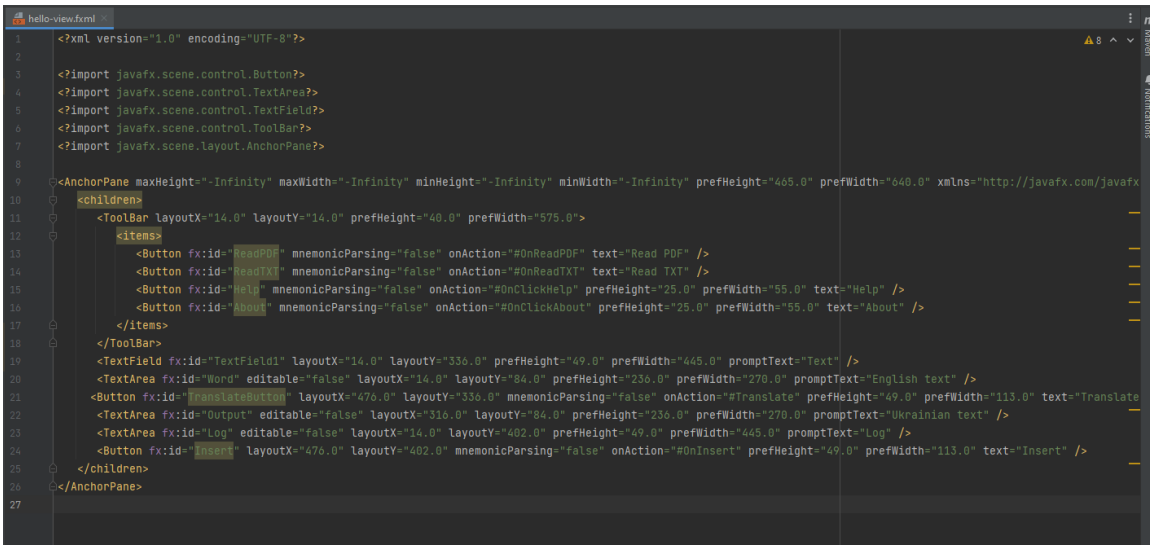

Рисунок 3.5—Зміст файлу hello-view.xml

На рисунку видно, що для кожного елементу приписується свій id для використання вже в налаштуванні логіки програми. Також написано позиціювання об’єктів на екрані програми, при цьому була уникнута необхідність ручного задавання координат. Властивість mnemonicParsingвикористовується в написанні функціоналу для використання користувачем гарячих клавіш. За замовчуванням властивості задається false. Для функціональних елементів, із якими користувач буде працювати безпосередньо, задана властивість onAction. Для розроблюваного програмного засобу вона буде використовуватися для налаштування клавіш та їхньої роботи. Конфігурація логіки для таких елементів буде прописана у файлі Controller, де вони й будуть викликатися за допомогою їхнього idта підпису дії. Також для зручності у текстових полях була прописана підказка у властивості promptText, яка буде зникати при появі користувацького тексту в них. При налаштуванні програми в програмному засобі також автоматично під’єднуються необхідні для роботи бібліотеки.

Розроблювальне програмне забезпечення є максимально зручним у використанні, із підписом усім клавіш та текстових полів. В інтерфейсі було враховано можливість роботи із ним як досвідченому, так і новому користувачу. Більш детальне пояснення функціоналу програми із розписом роботи та

призначення кожного об’єкту буде наведено в четвертому розділі в інструкції користувача.

3.2Під’єднання бібліотек та налаштування файлу pom.xml

Для полегшення роботи в написанні програм використовуються сторонні бібліотеки. Вони дозволяють використовувати розроблений раніше програмний код у різних програмах. Таким чином, програміст може не розробляти частину коду своєї програми, а скористатися тим, що входить до складу бібліотек.

У мові програмування Java коди бібліотек є функціями, розміщеними у файлах, які скомпільовані в об'єктні файли, а ті, у свою чергу, об'єднані в бібліотеки. В одній бібліотеці поєднуються функції, що вирішують певний тип завдань. Наприклад, є бібліотека математичних функцій.

Кожна бібліотека має мати свій заголовний файл, в якому повинні бути описані прототипи (оголошення) всіх функцій, що містяться в цій бібліотеці. За допомогою заголовних файлів ви "повідомляєте" вашого програмного коду, які бібліотечні функції є та як їх використовувати.

При компіляції програми бібліотеки підключаються лінковником, який викликається gcc. Якщо програмі потрібні лише стандартні бібліотеки, то додаткових параметрів передавати лінковнику не треба (є винятки). Він знає, де стандартні бібліотеки знаходяться, і підключить їх автоматично. В інших випадках при компіляції програми потрібно вказати ім'я бібліотеки та її місцезнаходження.

Бібліотеки бувають двох видів —статичні та динамічні. Код перших при компіляції повністю входить до складу файлу, що виконується, що дозволяє легко переносити програму. Код динамічних бібліотек не входить до файлу, а містить лише посилання на бібліотеку. Якщо динамічна бібліотека буде видалена або переміщена в інше місце, програма не працюватиме. З іншого боку, використання динамічних бібліотек дозволяє скоротити розмір файлу, що виконується. Також, якщо в пам'яті знаходиться дві програми, що використовують однакову динамічну бібліотеку, то бібліотека буде завантажена в пам'ять лише один раз [16].

Для роботи із проектом буде використовуватися Maven. Maven —це інструмент для управління та збирання проектів. Він полегшує життя розробнику на всіх стадіях роботи: від створення структури проекту та підключення необхідних бібліотек до розгортання продукту на сервері. При роботі з будь-яким фреймворком доведеться використовувати Maven [17].

Зручним є те, що для завантаження репозиторіїв бібліотек не потрібно буде шукати їх в Інтернеті. Таким чином, нам достатньо написати посилання на репозиторій у файлі pom.xml, як зображено на рисунку 3.6.

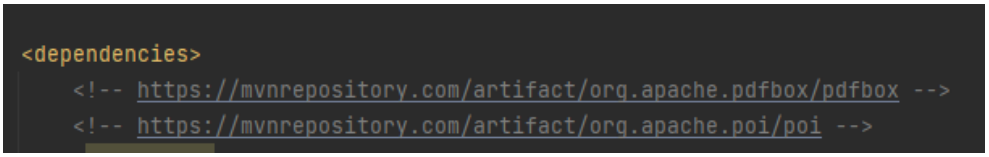

Рисунок 3.6—Завантаження репозиторіїв

Після цього нам потрібно прописати залежність для тих завдань, які необхідно буде виконати за допомогою даної бібліотеки. Приклад запису залежності показано на рисунку 3.7.

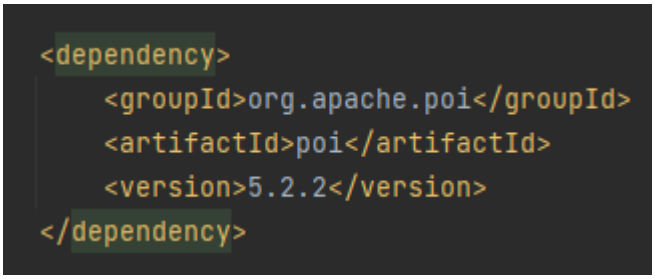

Рисунок 3.7—Написання залежності

Після написання всіх залежностей необхідно завантажити репозиторії та перевантажити проект. Здійснити це можна натиснувши правою клавішею миші на файл pom.xml. Його можна знайти із лівого боку на панелі файлів проекту. Після появи вікна із налаштуваннями треба перейти в самий низ і навести на поле Maven. У відкритому вкладенні обираємо Reloadproject, чекаємо деякий час, за який

відбувається завантаження репозиторіїв та перезавантаження проекту. Як перевантажити проект наочно показано на рисунку 3.8.

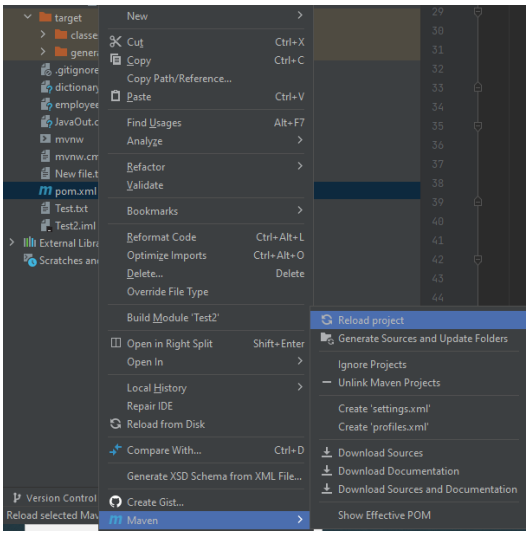

Рисунок 3.8—Перевантаження проекту для завантаження репозиторію

Після виконання всіх необхідних налаштувань у файлі pom.xml, перейдемо до файлу HelloApplication, де потрібно буде приєднати необхідні нам бібліотеки. Робиться це просто, за допомогою команди import, як зображено на рисунку 3.9. Додавання бібліотек потрібно прописувати в самому верху програми, перед усіма класами.

```
package com.example.test2;

import javafx.event.ActionEvent;
import javafx.fxml.FXML;
import javafx.fxml.Initializable;
import javafx.scene.Group;
import javafx.scene.Scene;
import javafx.scene.control.*;
import javafx.scene.image.Image;
import javafx.scene.image.ImageView;
import org.apache.pdfbox.pdmodel.PDDocument;
import org.apache.pdfbox.text.PDFTextStripper;
import javafx.stage.FileChooser;
import javafx.stage.Stage;
```

Рисунок 3.9—Імпорт бібліотек для використання

Для використання готових бібліотек у середовищі IntellijIdea, які не потребують завантаження сторонніх репозиторіїв, можна одразу прописувати необхідні нам строки коду, а після підсвічення натискати комбінацію клавіш alt + shift + enter. Після цього середовище розробки самостійно приєднає необхідну бібліотеку без необхідності його пошуку. Як це можна зробити показано на рисунку 3.10.

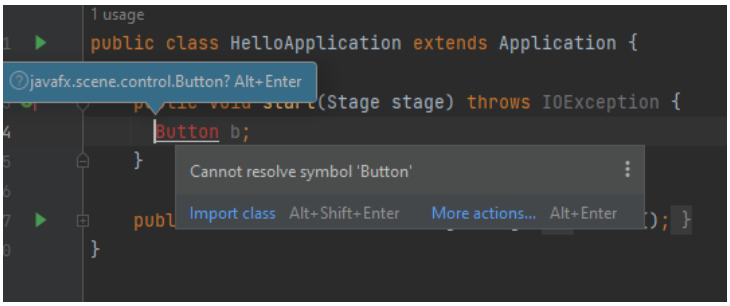

Рисунок 3.10—Простий спосіб додавання шаблонних бібліотек

У розробці програми буде використано безкоштовну версію бібліотеки ApachePOI. Apache POI —це бібліотека Java з відкритим вихідним кодом, наданий Apache, це досить обширна бібліотека, яка допомагає вам працювати з документами Microsoft, як Word, Excel, Power point тощо.

На останок, для завершення налаштування наших бібліотек, у файлі module-info.javaпотрібно прописати команду для використання цих самих бібліотек. Робиться це за допомогою команди requires, як зображено на рисунку 3.11.

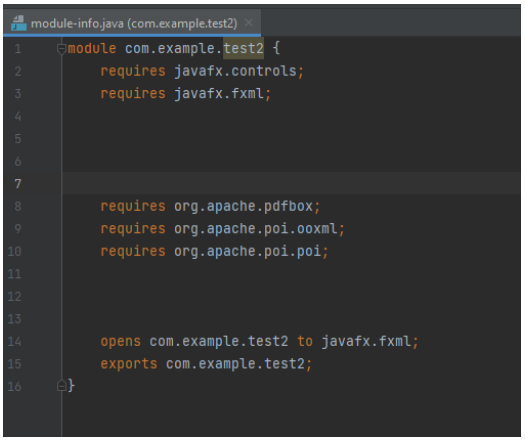

Рисунок 3.11— Завершення налаштування бібліотек для сторонніх репозиторіїв

3.3Побудова блок-схеми тарозробка програмного забезпечення

При розробці великих програмних засобів розробники часто починають із побудови блок-схеми алгоритму роботи програми. Така блок-схема —це просте викладення роботи програми та її дій при певній дії користувача. Зазвичай, такі схеми є своєрідною таблицею переходів, результатами яких може «так», або «ні». Також використовують позначення 1 або 0. Це хоч і не необхідний етап розробки програми, проте він дозволяє розділити проектування на етапи, або ж блоки, які полегшують розуміння логіки програми та дозволяє записати її дії в простому викладенні завдання, що також зручно для поставлення задачі для розробника. Блок-схема розроблюваного програмного забезпечення зображена на рисунку 3.12.

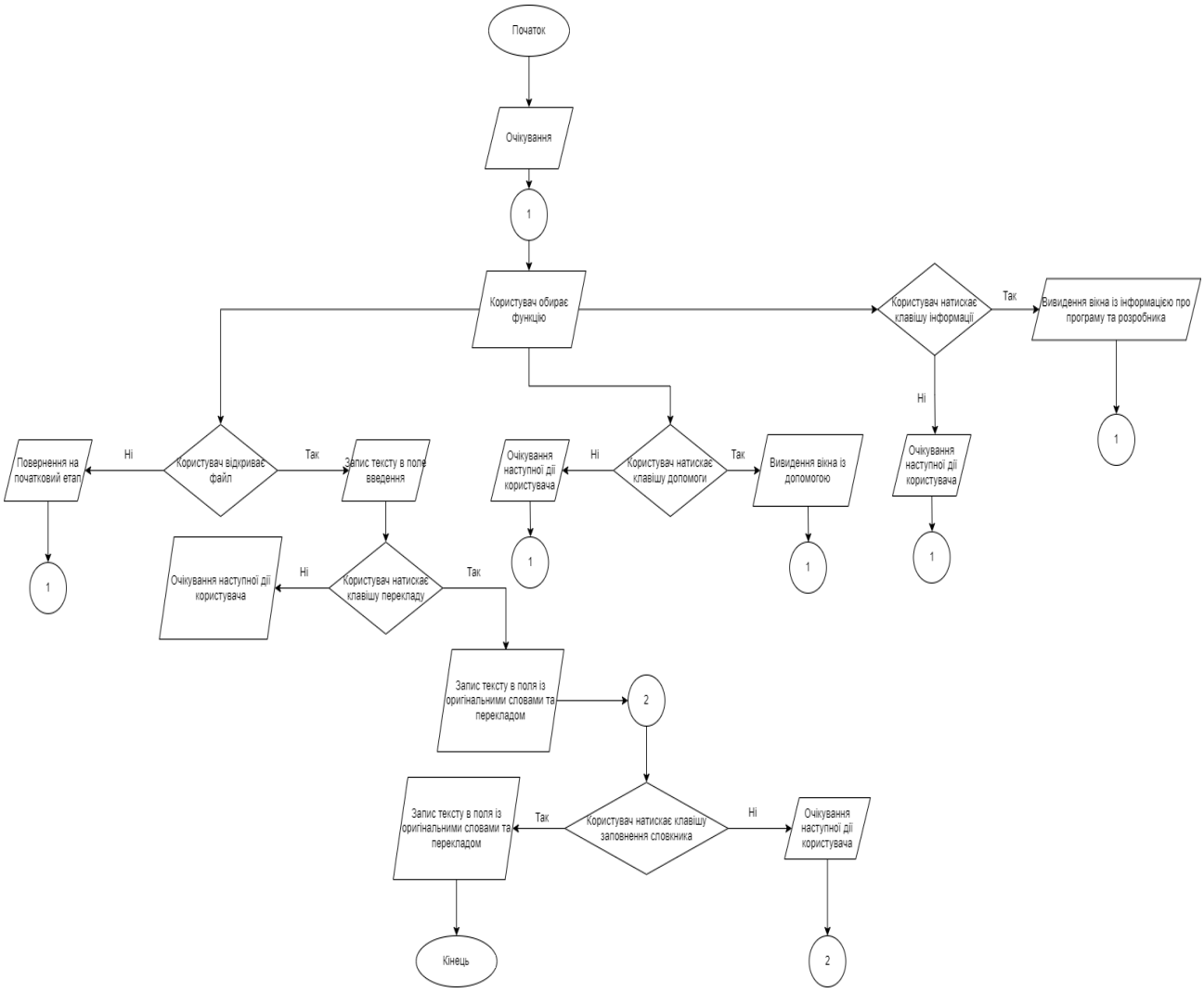

Рисунок 3.12—Блок-схема роботи алгоритму

При поставленій задачі можна приступати до безпосередньої розробки програми. Розпочати можна із файлу HelloApplication, у якому в розроблювальному програмному забезпеченні майже не буде змін після створення шаблонного файлу. На рисунку 3.13 зображено частину його вмісту, повний лістинг буде знаходитися в додатках.

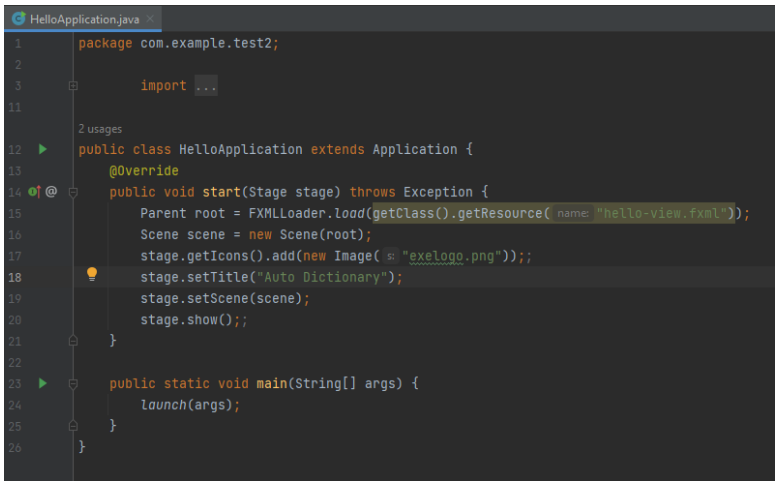

Рисунок 3.13—Зображення вмісту файлу HelloApplication.java

У даному файлі у верхній частині записується імпорт необхідних нам бібліотек. Далі, у методі startвідбувається завантаження нашого файлу інтерфейсу hello-view.xml. Запис Sceneслугує контейнером для всіх графічних елементів. Для покращення візуального вигляду можна задати іконку для програми, а також її заголовок. У методі mainвідбувається безпосередній запуск програмного засобу та виведення інтерфейсу.

Налаштування основної логіки програми відбувається у файлі HelloController.java. Він є найбільшим та містить опис усіх основних функцій програми, конфігурація клавіш, задання змінних тощо. Запис файлу, як і зазвичай, починається із завантаження бібліотек, що будуть містити необхідні нам функції. Далі, в основному класі HelloControllerми задаємо змінні для елементів інтерфейсу (рисунок 3.14), для їхньої конфігурації та використання. Їхні назви обов’язково мають співпадати із idy файлі hello-view.xml, інакше буде повертатися помилка. Ці

змінні є публічними, тому ми можемо використовувати їх у будь-якій частині нашої програми для запису в них тексту, дій при натисненні клавіш, виведення інформації або ж інших функцій, які будуть необхідні в розробці програми.

Далі реалізуємо функцію для перекладу translate(рисунок 3.15), де буде використано інтеграцію розроблюваного програмного забезпечення із засобами GoogleScripts за допомогою API. Більш докладно про цю частину програми буде написано в розділі 3.4.

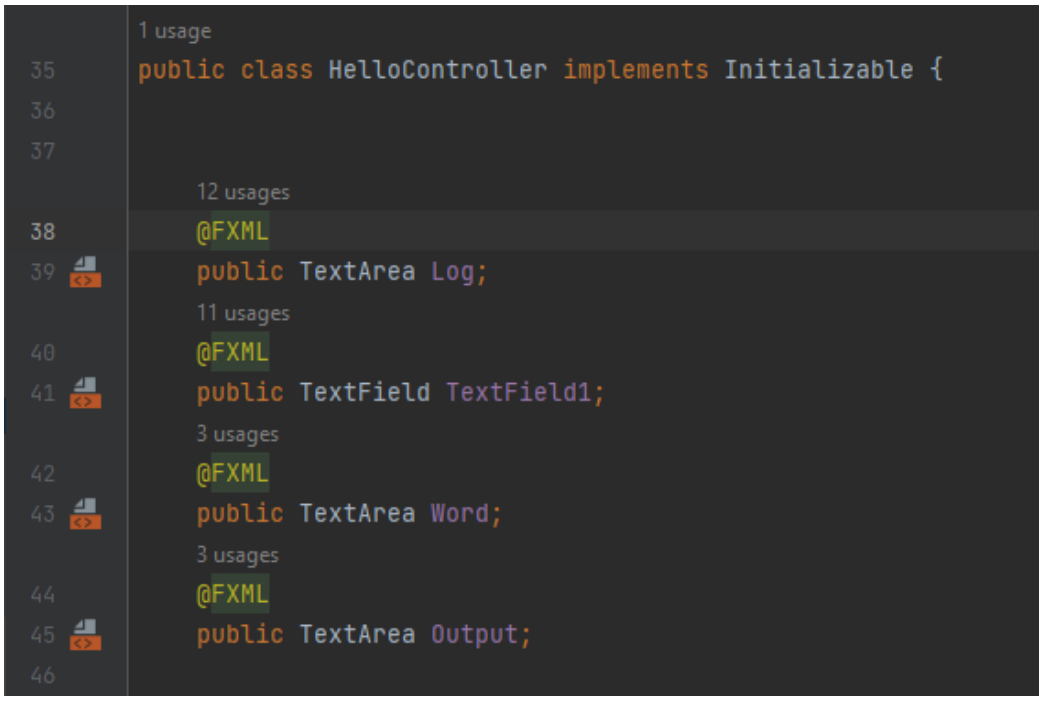

Рисунок 3.14—Задання змінних інтерфейсу

Рисунок 3.15—Реалізація використання GoogleScripts

Реалізовуємо зчитування файлу типу pdfдля нашого програмного забезпечення. Частина коду зображена на рисунку 3.16.

При натисненні клавіші ReadPDF спочатку задається пусте поле, у якому можна вводити текст або ж заносити текст із файлів. Саме у цьому методі будуть використані засоби бібліотеки ApachePOI, яка полегшує роботу із файлами, у тому числі й із pdf. Далі ми задаємо вивід нового вікна, де користувачу буде надана можливість обрати файл для зчитування інформації із нього. Далі ми створюємо документ уже в самій програмі, який буде завантажуватися із того самого файлу, який обрав користувач. У блоці ifми перевіряємо, чи обраний користувачем файл не є зашифрованим. Якщо ні, ми створюємо змінну для витягнення інформації із документу, та потім передаємо текст у змінну типу Stringдля його зберігання. Далі ми редагуємо текст із заміною символів для зручного виводу та записуємо його в поле TextField1, де текст і буде виведено для користувача. Якщо файл зашифровано, у полі виводу Logбуде показано текст помилки. Після написання основного блоку коду необхідно прописати команду закриття створеного документу.

```
1 usage
@FXML
public void OnReadPDF(ActionEvent actionEvent) throws IOException{
    TextField1.setText("");
 Stage stage = new Stage();
    FileChooser fil_chooser = new FileChooser();
    File file = fil_chooser.showOpenDialog(stage);
    if (file != null) {

        Log.setText("PDF file is selected");
    } //PDF Chooser


    PDDocument document = PDDocument.load(file);
      if (!document.isEncrypted()) {
          PDFTextStripper stripper = new PDFTextStripper();
          String text34 = stripper.getText(document);
         TextField1.setText(text34);
          String texx = TextField1.getText();
          String result = texx.replace( target: ".",  replacement: " ");

          TextField1.setText(result);
      }
      document.close();
```

Рисунок 3.16—Реалізація зчитування PDF-файлу

Наступним методом буде реалізація відкриття та запису файлу формату txt. У ньому ми виконуємо схожі із попереднім блоком дії, де очищуємо поле виводу та даємо користувачу вибір файлу для зчитування. Частина коду показана на рисунку 3.17. Для реалізації цього методу достатньо засобів IntellijIdea. Ми зчитуємо текст файлу та записуємо його в буфер читача, а потім, за допомогою циклу whilezаписуємо його зміст у поле виводу тексту, перевіряючи, поки буфер не стане пустим, після чого зупиняємо роботу циклу. Далі таким самим методом, як у попередньому методі редагуємо текст та передаємо його у вивід.

```
1 usage
@FXML
public void OnReadTXT(ActionEvent actionEvent) throws IOException {

    TextField1.setText("");
    Stage stage = new Stage();
    FileChooser fil_chooser = new FileChooser();
    File file = fil_chooser.showOpenDialog(stage);
    if (file != null) {

        Log.setText("TXT file is selected");
    }
    //TXT Chooser
    BufferedReader br = null;
    String c = ", ";

    try {
        br = new BufferedReader(new FileReader(file));
        String line;
        while((line = br.readLine() )!= null){

            TextField1.appendText(line);

            Log.setText("Success");
        }
        String texx = TextField1.getText();
        String result = texx.replace( target: ".",  replacement: " ");

        TextField1.setText(result);
```

Рисунок 3.17—Частина коду реалізації відкриття txtфайлу

У наступному кроці створення нашого програмного засобу ми задаємо вивід зображення із поясненням роботи програми. Виведення буде реалізовано за допомогою нового вікна, із своїм налаштування ширини та висоти сцени, підписом

та іконкою. Спочатку ми створюємо новий об’єкт класу Stage, що дозволить нам вивести нове вікно поверх старого. Далі ми створюємо змінну типу image, у яку передаємо наше зображення із допомогою користувачу. Також ми задаємо іконку для покращення візуального вигляду. Після налаштування висоти та ширини об’єкту, ми створюємо групу об’єктів та нову сцену, яку передаємо в stage. При правильній роботі блоку в поле Logпередається запис «success». Частина лістингу зображена на рисунку 3.18.

```
1 usage
@FXML
public void OnClickHelp(ActionEvent actionEvent) throws IOException{
    Stage stage = new Stage();
Image image = new Image( s: "help.png");
stage.getIcons().add(new Image( s: "glogo.png"));

ImageView imageView = new ImageView(image);

//position
imageView.setX(50);
imageView.setY(25);

//height width
imageView.setFitHeight(1009);
imageView.setFitWidth(900);
```

Рисунок 3.18—Частина методу для виведення допомоги

Наступник кроком буде реалізація виведення інформації про програму та розробника, код якого зображений на рисунку 3.19. Це буде просте повідомлення типу Alertіз виведенням блоку тексту. Такий спосіб запису зручний для виведення короткої інформації, наприклад, помилки в роботі програми.

```
1 usage
@FXML
public void OnClickAbout(ActionEvent actionEvent) {

    Alert alert = new Alert(Alert.AlertType.INFORMATION, s: "Ця програма була створена Олександром Трошенком " +
            "1КІ-186. Це програма для автоматизованого створення словника з англійської мови.");
    alert.setHeaderText(null);
    alert.setTitle("About");


    alert.show();
    Log.setText("Success");
}
```

Рисунок 3.19—Реалізація виведення повідомлення

Далі конфігуруємо натискання клавіші Translate, при якому буде здійснено переклад тексту, який було зчитано або введено безпосередньо користувач. Частина методу показано на рисунку 3.20. У цій частині ми отримуємо текст із поля Text-Field1, та редагуємо його, замінюючи різні символи. Після цього ми передаємо оригінальні англійські слова, що будуть розділені крапками, у поле Word. Переклад буде здійснюватися за допомогою функції translate, конфігурацію якої було описано раніше. У змінну типу Stringбуде передано відредагований текст. Далі записуються мови, із якими буде працювати програма. Запис відбувається за допомогою кодів мов, які можна знайти в Інтернеті. Так, для першої мови, англійської, із якої будуть перекладатися слова, код —це «en», а для української мови —мови, на яку буде здійснюватися переклад, код —це «uk». Через проблеми із кодуванням, символ апострофу буде виводитися некоректно певною послідовністю символів, через що буде реалізовано заміну цих символів на апостроф.

```
@FXML
public void Translate(ActionEvent actionEvent) throws IOException{


String text = TextField1.getText();
  String result = text.replace( target: ",", replacement: "").replace( target: "?", replacement: "")
          .replace( target: "!", replacement: "").replace( target: ":", replacement: "").replace( target: ";", replacement: "")
         .replace( target: "-", replacement: "")
           .replace( target: "=", replacement: "").replace( target: " ", replacement: ". ").replaceAll( regex: "&#39;", replacement: "'");
    Word.setText(result);

   String res = (translate( langFrom: "en", langTo: "uk", result) );
   String fin =res.replace( target: "&#39;", replacement: "'");
   Output.setText(fin);

   Log.setText("Success");
```

Рисунок 3.20—Блок методу перекладу

Останнім методом у програмі буде реалізація самого запису двох блоків тексту в словник у файл формату txt. Частина коду зображена на рисунку 3.21. Спочатку ми створюємо фасйл, або ж виводимо в Log, що файл уже існує. Далі ми

створюємо об’єкт для запису інформації у файл. Після цього записуємо дані, що лежать у полях із англійським та український текстом у змінні типу String. Оскільки таким чином слова запишуться у форматі рядка та будуть неправильно виводитися, переписуємо строки в масиви, користуючись інструментом split, який буде розділяти слова. Далі знаходимо довжину масиву для створення циклу типу for, який буде записувати слова в словник, записуючи два масиви в один ряд із розділенням.

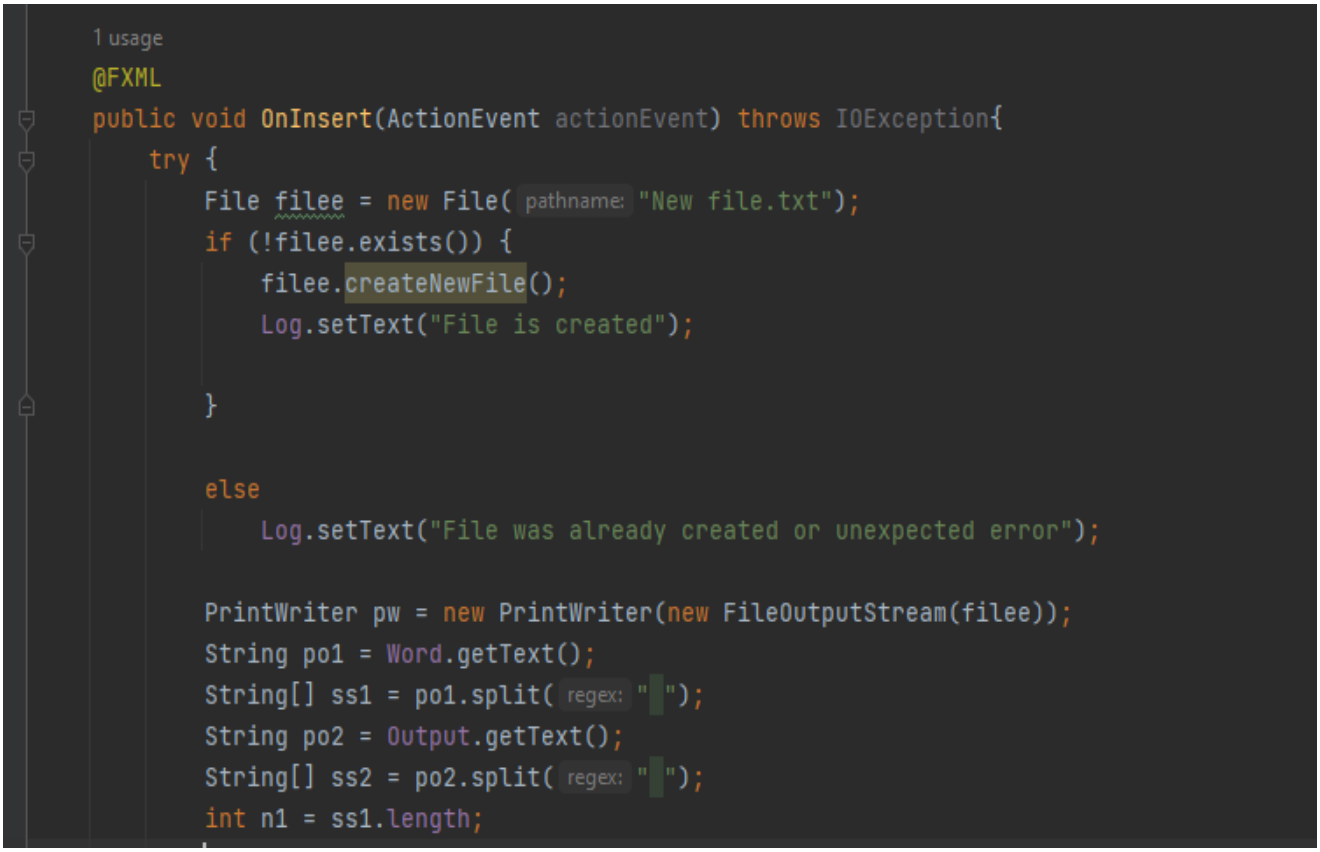

Рисунок 3.21—Реалізація заповнення словника

3.4Реалізація функції перекладу за допомогою Google Apps Script API

За допомогою Google Apps Script API буде здійснюватися функція перекладу, оскільки це досить зручний інструмент для реалізації обміну даними між пристроєм та мережею. API —це механізми, які дозволяють двом програмним компонентам взаємодіяти одне з одним, використовуючи набір визначень та протоколів. Наприклад, система метеослужби містить щоденні дані про погоду. Програма

погоди на телефоні спілкується з цією системою через API і показує щоденні оновлення погоди на телефоні.API —Application Programming Interface, що означає програмний інтерфейс програми. У контексті API слово "додаток" відноситься до будь-якого програмного забезпечення з певною функцією. Інтерфейс можна розглядати як сервісний контракт між двома програмами. Цей договір визначає, як вони взаємодіють один з одним, використовуючи запити та відповіді. Документація API містить інформацію про те, як розробники повинні структурувати ці запити та відповіді. Архітектура API зазвичай пояснюється з погляду клієнта та сервера. Програма, яка надсилає запит, називається клієнтом, а програма, що надсилає відповідь, називається сервером. Отже, у прикладі з погодою база даних служби — сервер, а мобільний додаток —це клієнт [18].

Google Apps Script API дозволяє програмовано створювати, змінювати та розгортати проекти сценаріїв програм —дії, які в іншому випадку вимагають використання редактора сценаріїв програм. Ваші програми можуть використовувати API для керування проектами сценаріїв, створення та розгортання нових версій сценаріїв, а також моніторингу виконання сценаріїв.API Apps Script також замінює та розширює API виконання Apps Script. Ви можете використовувати Apps Script API для віддаленого виконання функцій Apps Script, як і з Execute API [19].

Для створення скрипту необхідно відкрити відповідну сторінку та створити новий проект. Далі, ми записуємо код, який поширюється у вільному доступі в Інтернеті. Код зображено на рисунку 3.22. У ньому ми задаємо функцію для отриманнякодування тексту, мови вхідного тексту та мови перекладу. Далі, налаштовуємо переклад тексту за допомогою вбудованих засобів Googleта повертаємо результат у вигляді тексту.

Для можливості використання даного APIу розроблюваній програмі необхідно вивантажити даних скрипт як веб-додаток. Для цього необхідно перейти в поле Публікація та натиснути на Вивантажити як веб-застосунок. Далі, у новому полі, потрібно обрати тип доступу «усім, навіть анонімним», без цього програма не зможе отримувати дані із API. Після натиснення клавіші Вивантажити відкривається вікно,

де буде посилання на наш скрипт. Саме його ми будемо використовувати в блоці перекладу в IntellijIdea. Інструкція із вивантаження зображена на рисунку 3.23

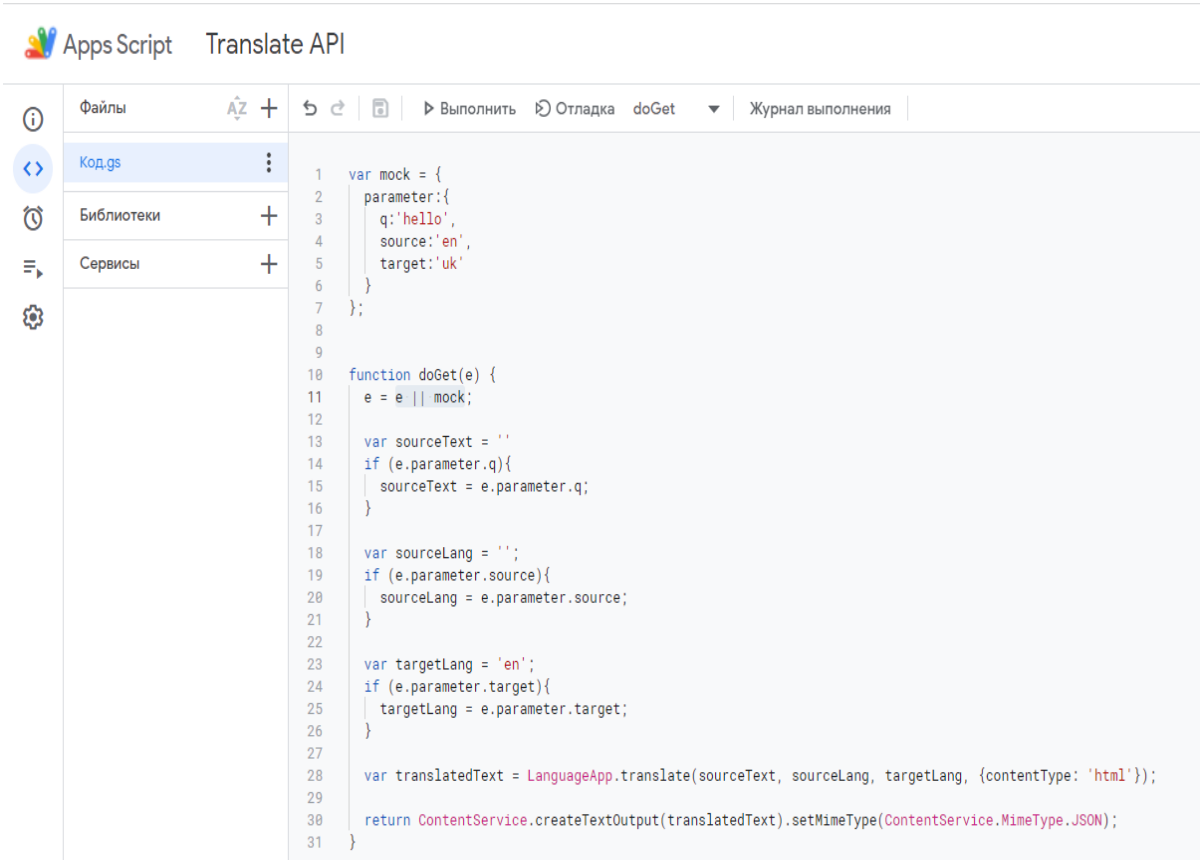

Рисунок 3.22—Вигляд GoogleScriptsAPI

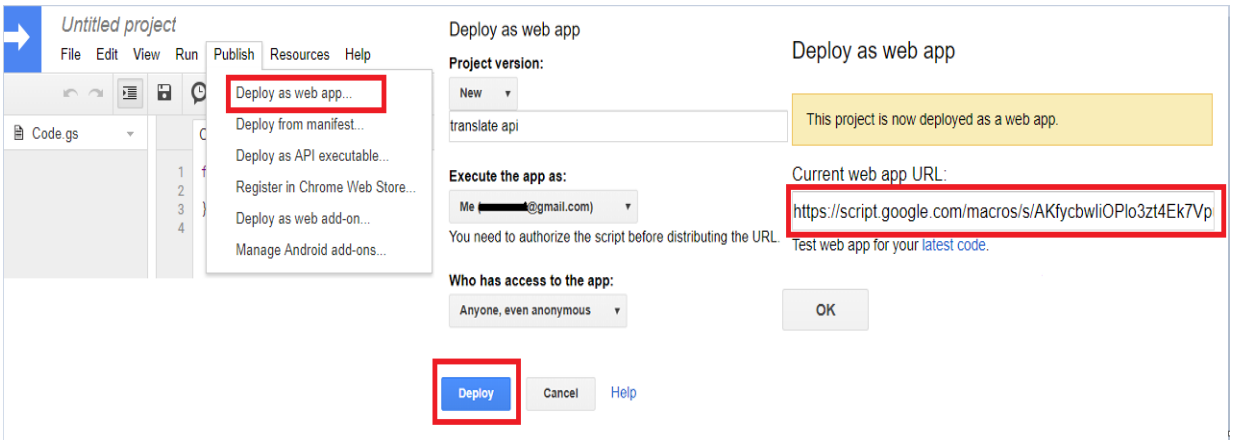

Рисунок 3.23—Публікація скрипту як веб-застосунок

Повертаючись до розробки функції translate, розглянемо його більш детально. Нам потрібно отримати посилання на наш скрипт, для цього запишемо його в змінну

типу String, де, окрім посилання, буде тип кодування тексту, мова оригіналу та призначення. Далі, встановлюємо з’єднання із API та отримуємо відповідь, яку записуємо в буфер. Після цього зчитуємо його, й записуємо в змінну. На останок, повертаємо змінну із відповіддю API, яка і є нашим перекладом тексту.

4ТЕСТУВАННЯ ДОДАТКУ. ІНСТРУКЦІЯ КОРИСТУВАЧА

4.1 Тестування роботи програмного застосунку

Прикладне тестування —це діяльність, яка часто виконується майже кожним тестером програмного забезпечення у його кар'єрі. Ці два слова надзвичайно широкі за практичними аспектами. Будь-то невелике програмне забезпечення, як калькулятор з лише основними арифметичними операціями або жвелика система на підприємстві, є три категорії додатків:прикладні, мережеві, мобільні. Для настільних додатків тестування має враховувати інтерфейс користувача, бізнес-логіку, бази даних, звіти, ролі та права, цілісність, функціональність, продуктивність, безпеку, апаратну та програмну сумісність та потік даних. Для веб-застосунків тестери повинні приділяти достатню увагу продуктивності, навантаженню та безпеці програми.Інші основні типи тестування, що охоплюють веб-тестування додатків — це функціональне тестування, крос-браузерне тестування, бета-тестування, регресійне тестування, тестування сумісності, пробне тестування, тестування сумісності та багатомовної підтримки та стресове тестування.Для мобільних додатків основними типами тестування, які мають бути зроблені, є тестування інтерфейсу користувача, засноване на правилах тестування, регресія, функціональне тестування та тестування безпеки. Так AUT (додаток на стадії тестування) — це або настільне програмне забезпечення, або веб-сайт чи мобільний додаток [20].

При запуску додатку користувач побачить головне вікно програми, де будуть зображені основні функції. Він може відкрити файл типу txtабо pdf, відкрити вікно допомоги або вікно інформації про додаток. Також він може вручну записати слова для словника і запустити функцію перекладу. На останок він може вставити текст у словник формату txt. Усі ці функції мають працювати без проблем із роботою. Розглянемо функцію виведення допомоги користувачу. Вона працює через натиснення клавіші Helpта викликом нового вікна із зображенням. Робота функції показана на рисунку 4.1.

Перевіримо роботу функції виведення інформації про програму за допомогою клавіші About. Перевірка робота зображена на рисунку 4.2.

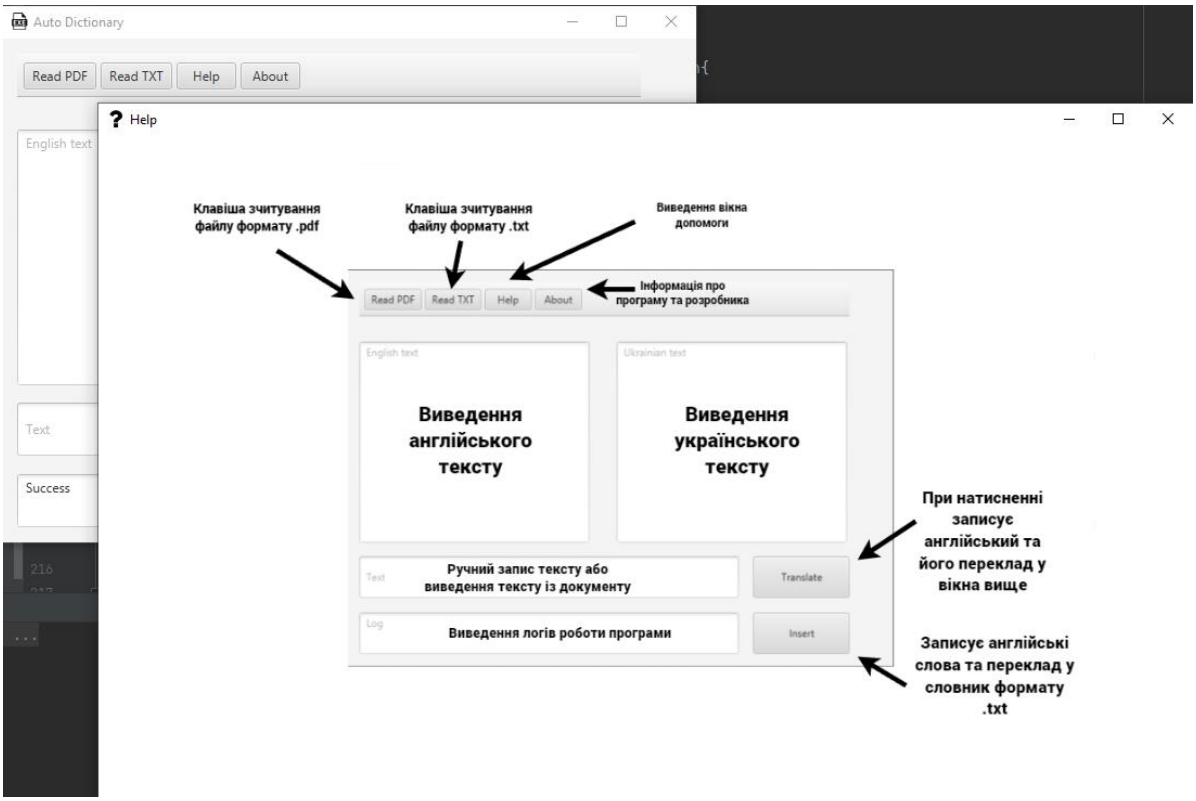

Рисунок 4.1 —Робота клавіші Help

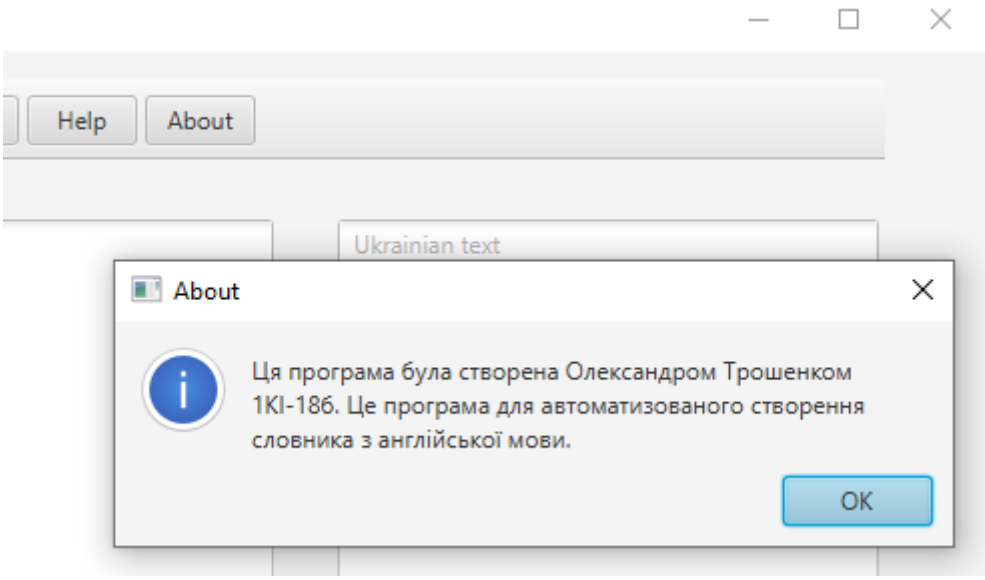

Рисунок 4.2 —Робота клавіші About

Далі, користувачу надається функція відкрити pdfфайл, зчитати інформацію із нього та вставити її в текстове поле. При натисканні на клавішу ReadPDFповинне відкритися вікно із провідником та можливістю вибрати в ньому файл. На

рисунку 4.3 зображено функцію роботи пошуку файлу. Далі, програма повинна записати текст із файлу в поле. Зчитування показане на рисунку 4.4.

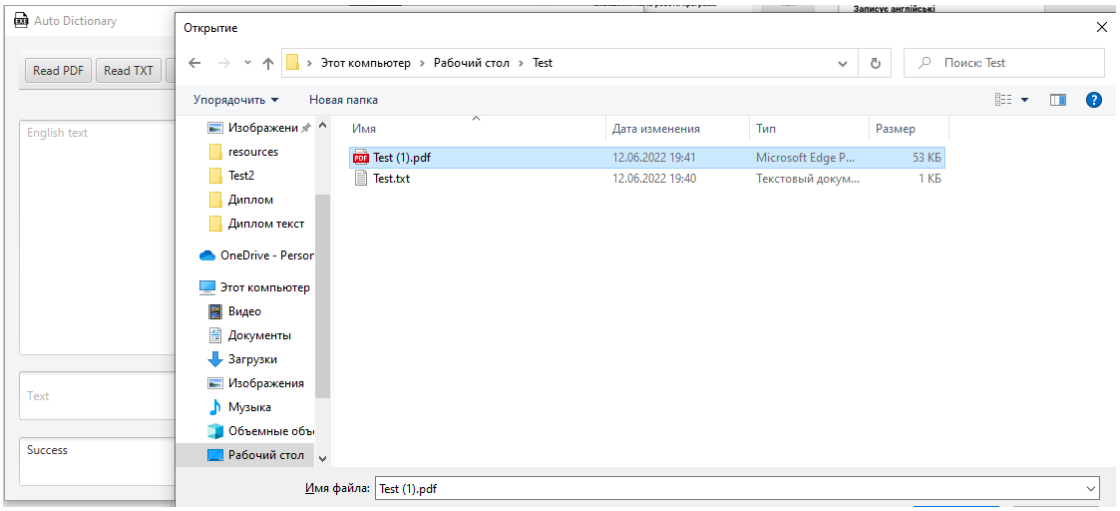

Рисунок 4.3 —Відкриття PDF файлу

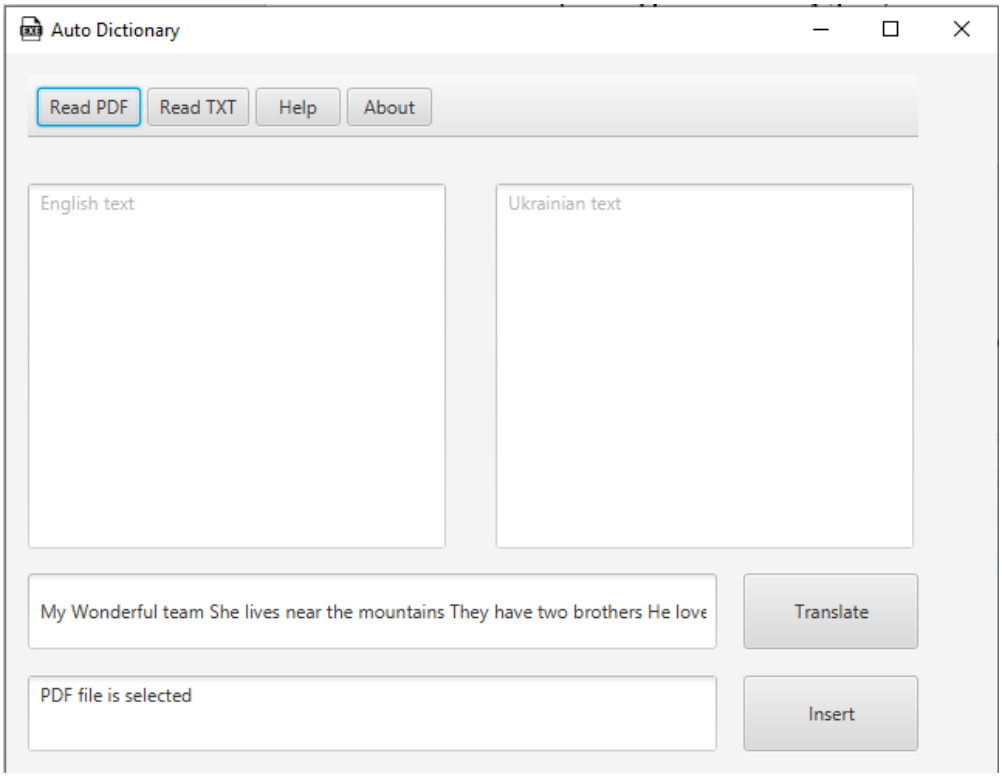

Рисунок 4.4 —Запис тексту із файлу PDFу поле вводу

Наступною функцією, яку може викликати користувач є відкриття файлу .txtчерез клавішу ReadTXT. Програма має виконувати ті самі функції, що й

зчитування .pdfфайлу. Перевіримо функцію відкриття файлу, яка буде зображена на рисунку 4.5, і функцію запису, що зображена на рисунку 4.6. Програма правильно відкриває файл та записує в текстове поле інформацію, що знаходиться у файлі.

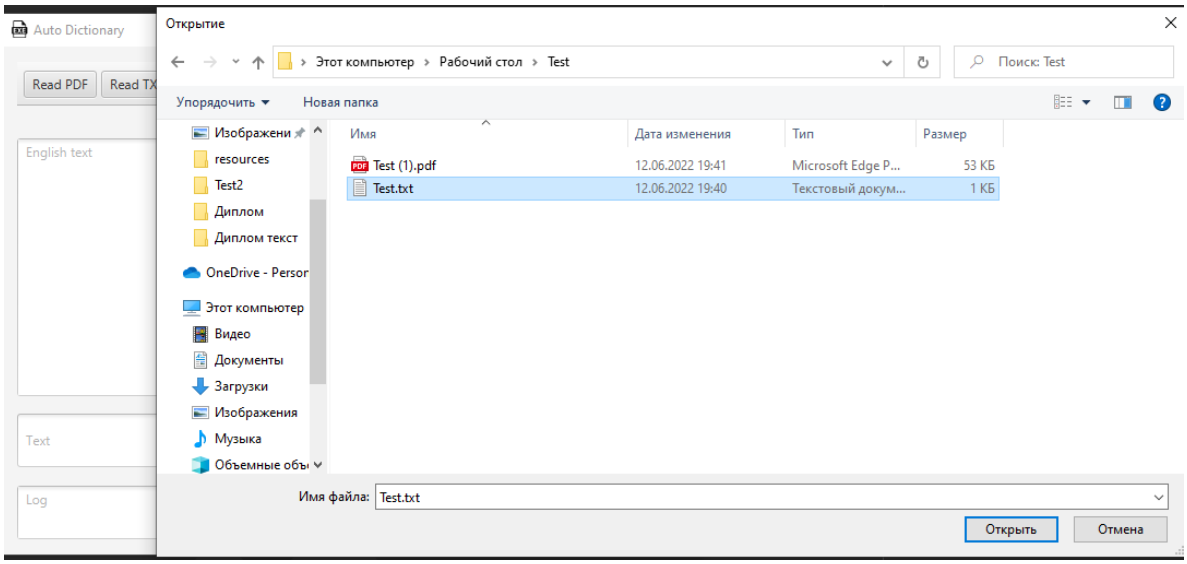

Рисунок 4.5 —Відкриття файлу формату .txt

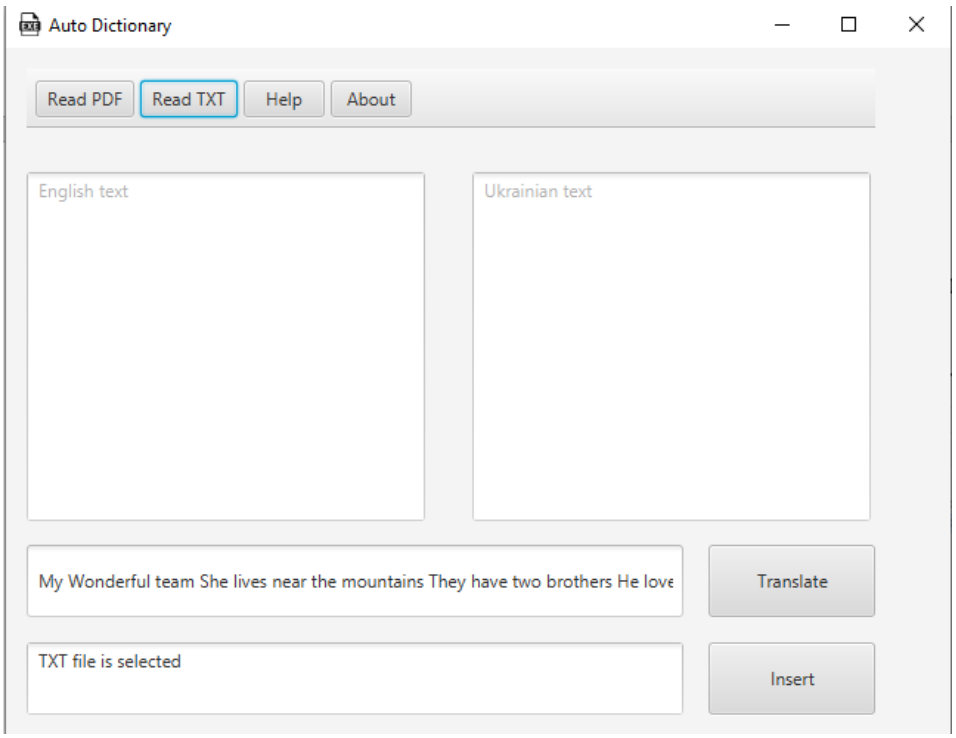

Рисунок 4.6 —Запис інформації у файлі в текстове поле

Також користувач може використати функцію запису слів вручну з клавіатури. Для цього запишемо деякий текст у текстове поле. Робота функції зображена на рисунку 4.7.

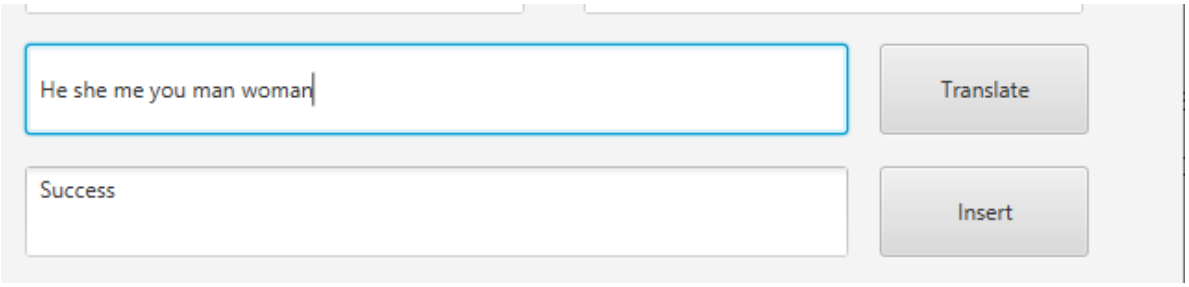

Рисунок 4.7 —Робота функції ручного введення тексту

При створенні словника необхідно буде перекласти дані, що знаходяться в текстовому полі та занести англійські та українські слова у відповідні поля виводу. Дана функція прив’язана за клавішею Translate, її робота показана на рисунку 4.8.

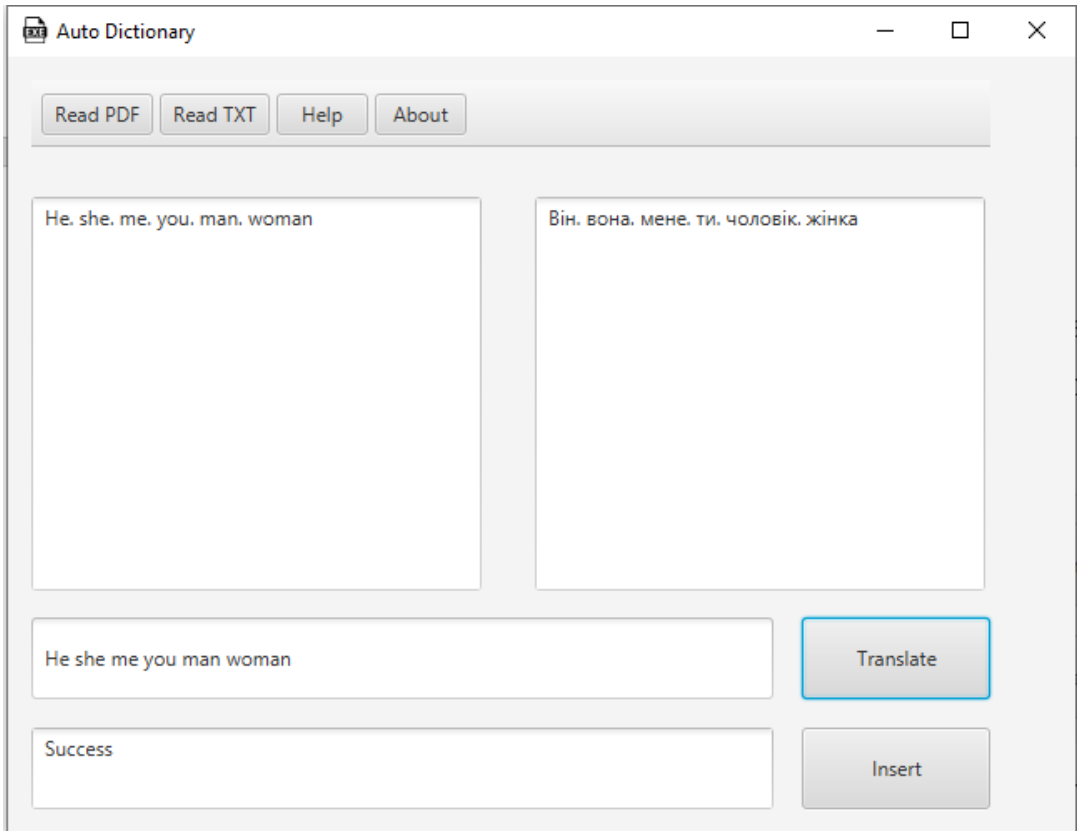

Рисунок 4.8 —Робота функції перекладу

Останньою функцією є створення словника у файлі формату txt. Для цього існує клавіша Insert, яка записує слова оригіналу та перекладу в колони в текстовому файлі. Перевіряємо функціонал, натиснувши клавішу вставки та перевіривши файл. Заповнення файлу словами, які були введенні на рисунку 4.8, буде показано на рисунку 4.9.

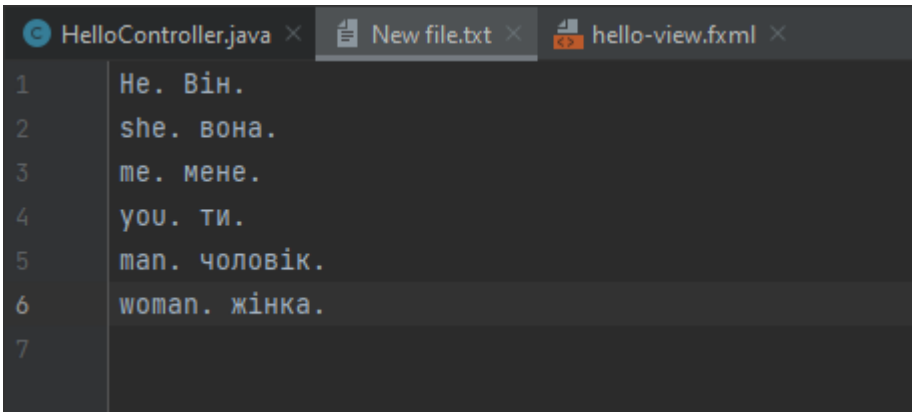

Рисунок 4.9 —Запис файлу в текстовий словник

Як зображено на рисунках, усі функції програми працюють правильно, що дозволяє користувачу без проблем використовувати програмне забезпечення у своїх цілях.

4.2. Інструкція користувача

Дане програмне забезпечення виконує функції створення користувацького словника шляхом зчитування файлів або ручного запису слів, перекладом їх на українську мову та записом результату в текстовий файл. Вона може використовуватися користувачем у якості створення бази слів, які необхідні в особистій роботі або роботі закладу чи підприємства, а також у цілях навчання іноземній мові шляхом записування незнайомих слів або прочитаних текстів, формуючи таким чином словник — посібник. Інтерфейс програми виконаний у мінімалістичному стилі, що дозволяє не навантажувати систему та запускати додаток на всіх типах комп’ютерів, а також зрозумілий у користуванні. Початок

роботи починається зі стартового вікна, який і є основний у програмі. Інтерфейс зображено на рисунку 4.10.

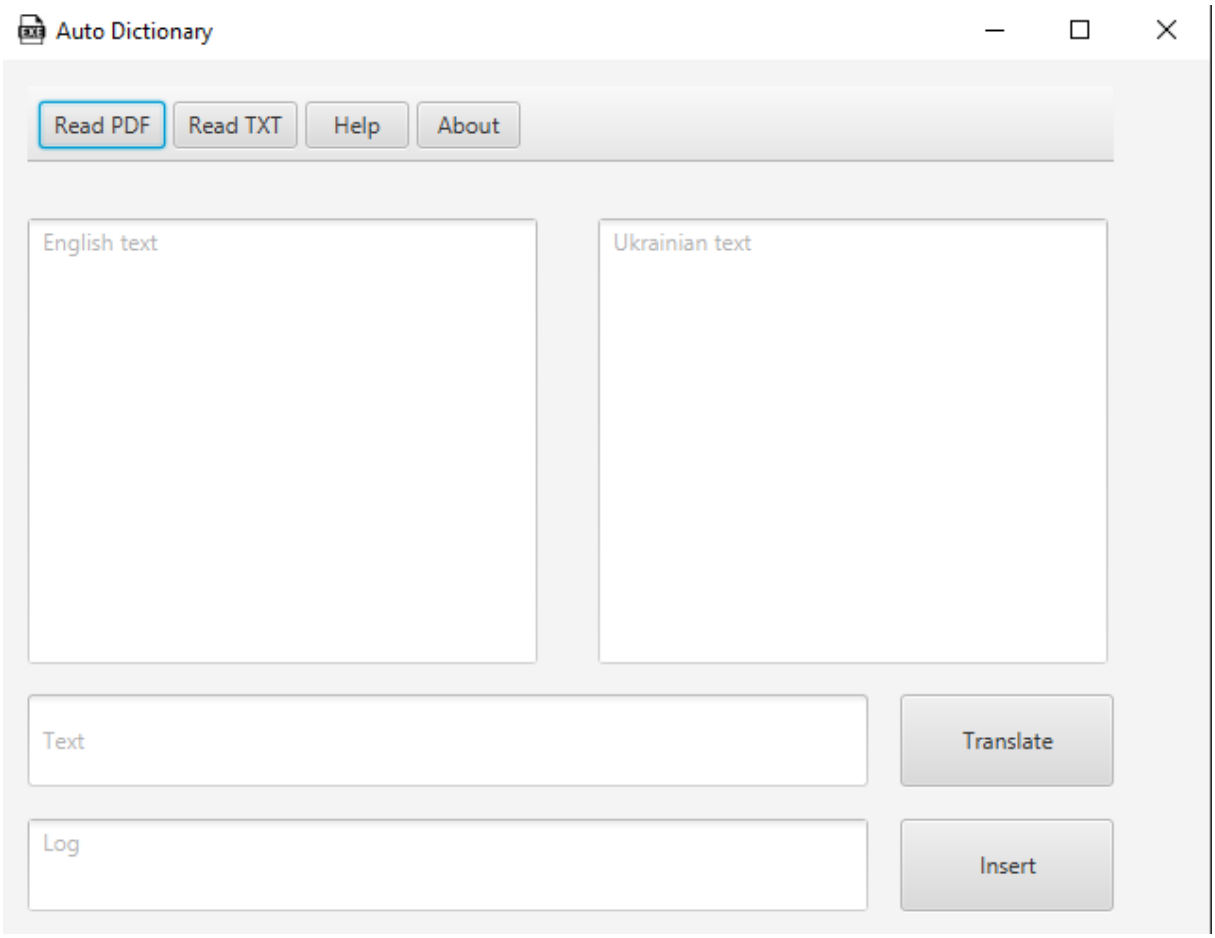

Рисунок 4.10 —Інтерфейс програмного застосунку

Для зручності користувача всі клавіші підписані, а також додані підказки на текстових полях для розуміння їхнього призначення. Даний текст пропадає під час запису в нього інформації. При бажанні записати текст у словник вручну, необхідно ввести слова або речення в текстове поле із підписом Text. Приклад введення зображено на рисунку 4.11.

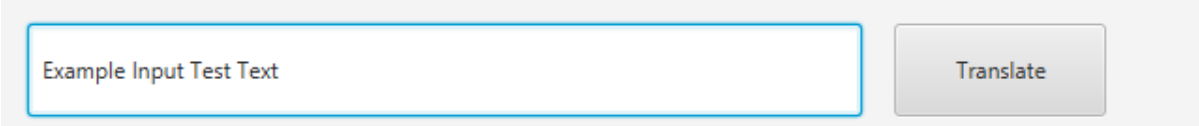

Рисунок 4.11 —Приклад введення тексту з клавіатури

Окрім ручного введення, можна також автоматизовано записати туди слова із файлів формату pdfабо txt. Робиться це шляхом натискання клавіші ReadPDFта ReadTXTвідповідно. При натисканні відкриється вікно із вашими файлами, де можете обрати необхідний вам файл. Для цього оберіть файл того типу, на яку клавішу ви натиснули та натисніть Відкрити, як показано на рисунку 4.12.

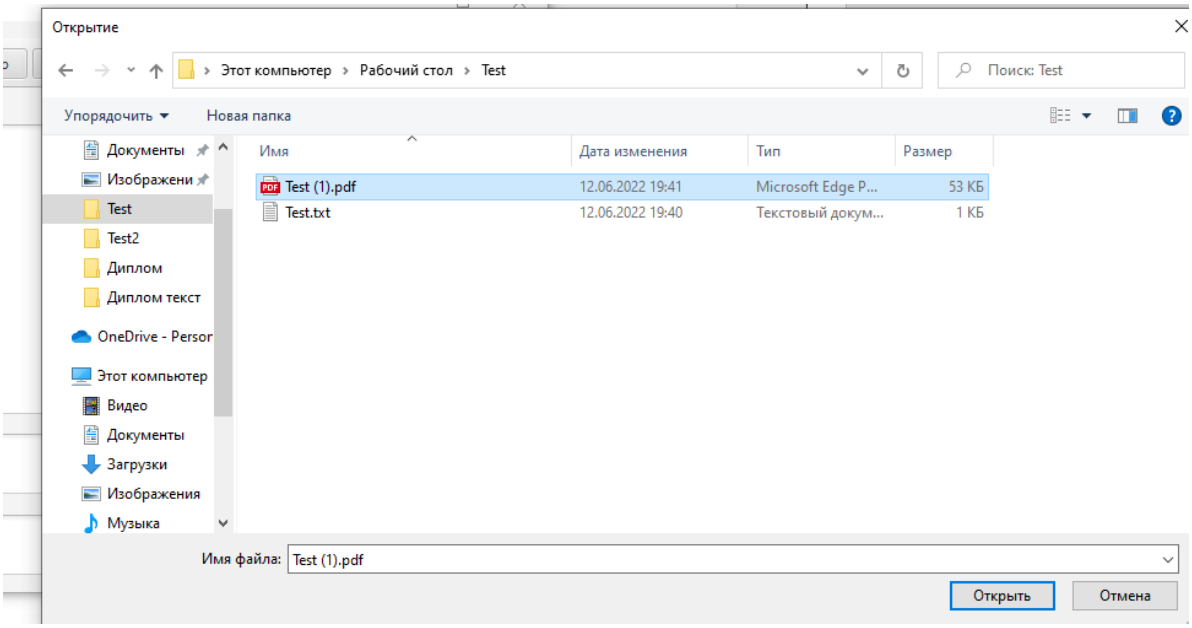

Рисунок 4.12 —Приклад відкриття файлу при натисненні на клавішу ReadPDF

Це автоматично запише текст із документу в текстове поле Text, де вони зможете відредагувати його за бажанням, або ж залишити так, як є. При натисканні на клавішу Translateваш текст розділиться на слова та запишеться в поле Englishtext, де розділиться крапками для зручності. Також ці слова будуть перекладені за допомогою засобів Googleта запишуться в поле Ukrainiantext таким самим чином, як і англійські слова. Ця функція показана на рисунку 4.13. При бажанні ви можете відредагувати текст, який вставили в переклад, та знову натиснути на клавішу Translate. Це змінить текст у полях, що дасть вам змогу зробити словник саме із тими словами, які ви бажаєте. При умові, що текст буде довшим за текстове поле, у вас буде можливість прогортати виведення. Зверніть увагу, що для роботи функції перекладу на вашому пристрої має бути доступ до мережі Інтернет, оскільки переклад здійснюється через інтернет-запит. В іншому випадку, переклад здійснити

буде неможливо. Проте ви зможете користуватися словником, якщо ви створили його раніше.

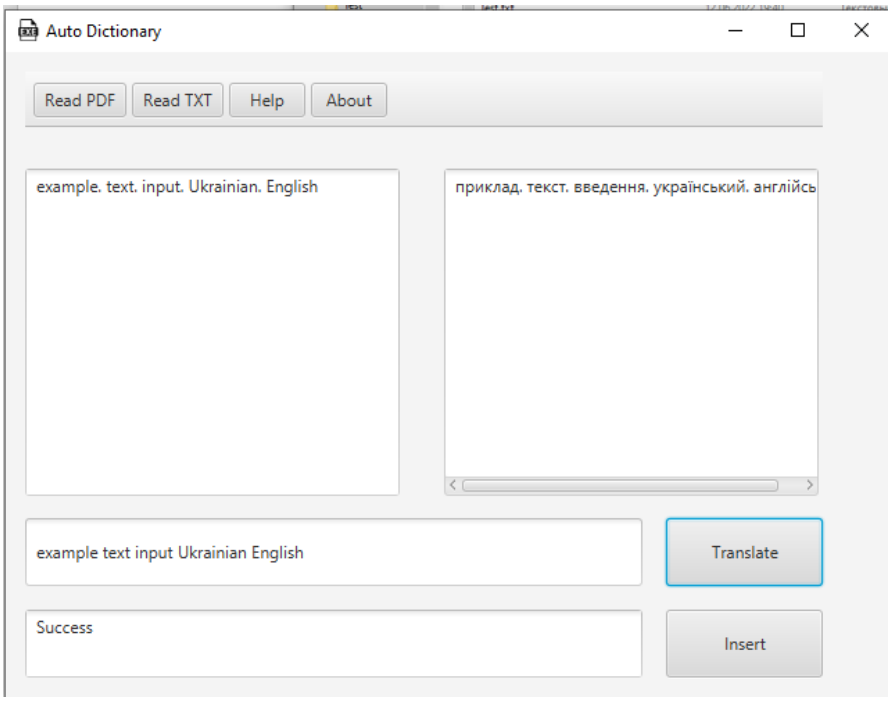

Рисунок 4.10 —Приклад реалізації функції перекладу тексту

Для вашої зручності кожна ваша дія буде описуватися в полі Log, де буде показано, чи виникла якась помилка, або ж усе виконалося успішно. Також є функція виклику інформації про програму та розробника, яка працює через натиснення клавіші About. Вона виведе просте текстове повідомлення, яке буде незалежне від основного вікна.

При бажанні створити словник, вам необхідно лише натиснути на клавішу Insert. Програма автоматично запише слова оригіналу та їхній переклад українською в колони, відповідно до одне одного. Створений словник буде формату txt, що дозволяє відкривати його майже на будь-якому пристрої. Приклад створеного словника показано на рисунку 4.11.

На останок, при відсутності інструкції користувача під рукою та необхідності зрозуміти будь-яку функцію програми, користувач може натиснути клавішу Help. Після цього відкриється нове вікно із зображенням, де буде інтерфейс програми із

підписом кожної функції клавіш, а також підписом текстових полів. На рисунку 4.12 показане саме зображення, яке буде виводитися для користувача.

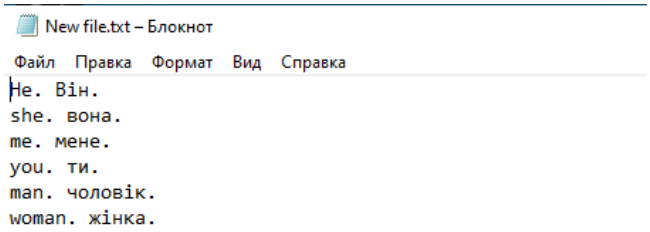

Рисунок 4.11 —Приклад створеного словника із записаними словами

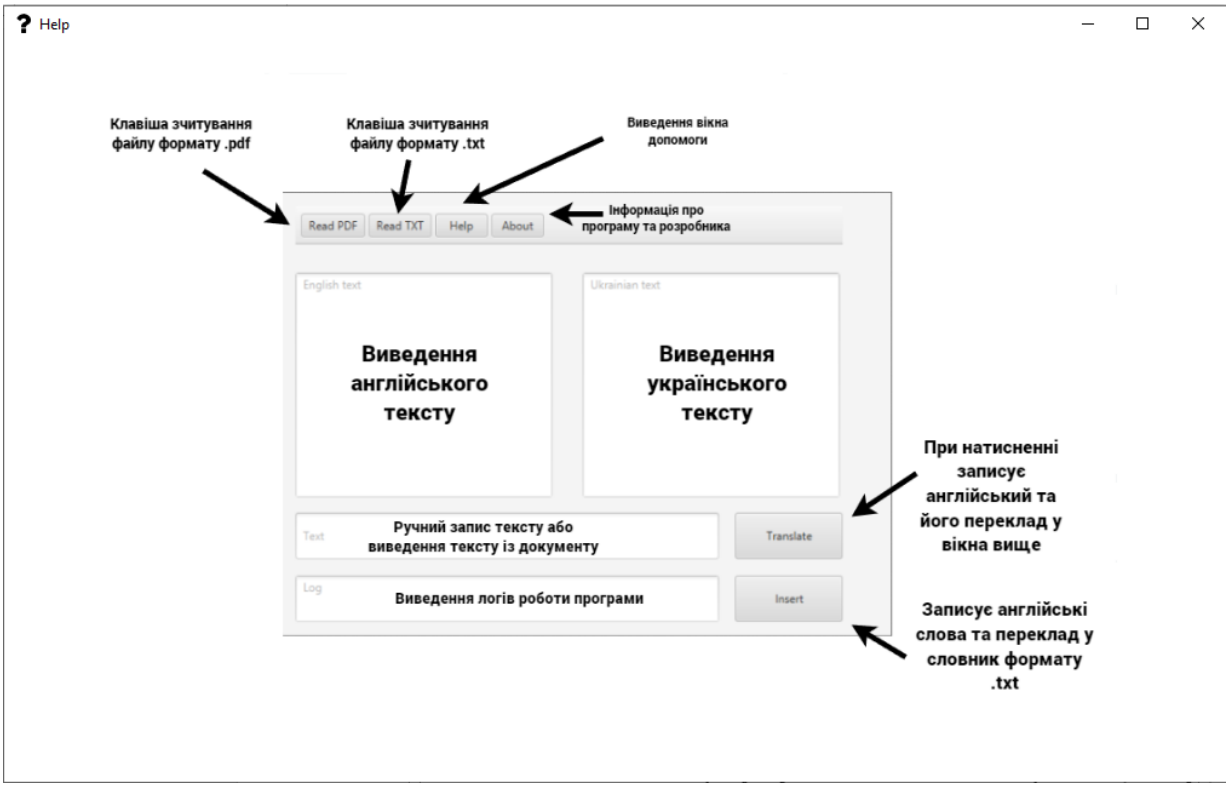

Рисунок 4.12 —Вікно Helpдля допомоги користувачу

ВИСНОВКИ

Уцій дипломній роботі було розробленосистему автоматизованого створення словника з англійської мови.

Даний додаток можна використовувати для створення словників як для вивчення англійської мови чи покращення знань, або ж для використання на підприємствах.

Було проаналізовано основні типи існуючих словників із англійської мови. Розглянуто їхні основні недоліки та переваги. Розглянуто основні параметри, за якими повинні створюватися словники. Проаналізовано та розглянуто існуючі системи, що створюють або використовують словники та виділено основні їхні переваги та недоліки. Проаналізовано існуючі методи та засоби для створення ПЗ.

Також було проаналізовано особливості розробки системи для автоматизованого словника із англійської мови, розглянуто переваги створюваного типу словника. Далі було розглянуто особливості використання електронних словників, якими і буде користуватися розроблюване програмне забезпечення. Крім цього, було проаналізовано особливості використання розроблювального додатку.

Також було розглянуто особливості та переваги обраного середовища для розробки, а також мови програмування. Вирішеноосновні задачі, що потрібно вирішити при налаштуванні інтерфейсу, а також програмні засоби для цього. Описано переваги у використанні сторонніх бібліотек та способи їхнього під’єднання. Побудовано блок-схему програми й розроблено функціонал додатку. Описано реалізацію функції перекладу за допомогою GoogleScriptAPI.

На останок, розроблюване програмне забезпечення було протестовано на момент неправильної роботи чи неполадок у виконанні. Крім цього, було написано інструкцію користувача для забезпечення максимальної зручності роботи.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Використання словників при перекладі [Електронний ресурс]. — 2022. — Режим доступу: https://www.azurit.kiev.ua/uk/2017/08/03/vikoristannya-slovnikiv-pri-perekladi/.

2. Ткаченко О.М. Об’єктно-орієнтоване програмування мовою Java / О.М. Ткаченко, В.А Каплун — Навчальний посібник. —Вінниця: ВНТУ, 2006. —107с.

3. ПотсЯ. JavaFX. Початок роботи з JavaFX / Я. Потс, Н. Хільдебрант, Дж. Гордон, С. Кастілло— Вид.: «Oracle», 2022. —68с.

4. А. В. Снігур О. О. Трошенок / Проектування підсистеми автоматизованого створення словника з англійської мови // Тези доповіді. LI Науково-технічна конференція факультету інформаційних технологій та комп'ютерної інженерії. Вінниця 2022 р. Режим доступу: https://conferences.vntu.edu.ua/index.php/all-fitki/all-fitki-2022/paper/view/15378/12943

5. Як користуватися Duolingo? [Електронний ресурс]. — 2022. — Режим доступу: https://support.duolingo.com/hc/ru/articles/360035932192/.

6. English Galaxy Англійська мова [Електронний ресурс]. — 2022. — Режим доступу: https://play.google.com/store/apps/details?id=ru.englishgalaxy&hl=en_US.

7. Ходячий Словник слова зі слів онлайн [Електронний ресурс]. — 2022. — Режим доступу: https://apkpure.com/ru/com.IvanAndreichuk.HSlovnyk.

8. DictMaster — програма створення словників [Електронний ресурс]. — 2022. — Режим доступу: http://wladm.narod.ru/DictMaster/index.html.

9. BX Language acquisition [Електронний ресурс]. — 2022. — Режим доступу: https://roi4cio.com/categories/category/ehlektronnye-slovari/.

10. Електронні словники [Електронний ресурс]. — 2022. — Режим доступу: http://bxmemo.com/.

11. Хасанраза А. Вивчення IntelliJIdea / А. Хасанраза — Вид.: «Hasanraza Ansari», 2021 — 398с.

12. Завантажити IntelliJ IDEA [Електронний ресурс]. — 2022. — Режим доступу: https://www.jetbrains.com/ru-ru/idea/download/#section=windows.

13. Режим зберігання проти миттєвого режиму [Електронний ресурс]. — 2022. — Режим доступу: https://docs.microsoft.com/en-us/windows/win32/learnwin32/retained-mode-versus-immediate-mode.

14. МоррісС. JavaFX вдії / С. Морріс—Вид.: «Manning Publications», 2009—375с.

15. Посібник з Maven. POM. [Електронний ресурс]. — 2022. — Режим доступу: https://proselyte.net/tutorials/maven/pom/.

16. JavaFX Scene Builder [Електронний ресурс]. — 2022. — Режим доступу: https://www.oracle.com/java/technologies/javase/javafxscenebuilder-info.html.

17. Бібліотеки [Електронний ресурс]. — 2022. — Режим доступу: https://younglinux.info/c/library.

18. Основи Maven. Частина 4 [Електронний ресурс]. — 2022. — Режим доступу: https://javarush.ru/groups/posts/2523-chastjh-4osnovih-maven.

19. Що таке API? [Електронний ресурс]. — 2022. — Режим доступу: https://aws.amazon.com/ru/what-is/api/.

20. Керування скриптами за допомогою API REST [Електронний ресурс]. — 2022. — Режим доступу: https://developers.google.com/apps-script/api/concepts.

21. Тестування програм. Основи тестування програмного забезпечення [Електронний ресурс]. — 2022. — Режим доступу: https://www.softwaretestinghelp.com/application-testing-into-the-basics-of-software-testing/.

ДОДАТОК А

Технічне завдання

Міністерство освіти і наукиУкраїни

Вінницький національний технічний університет

Факультет інформаційних технологій та комп’ютерної інженерії

Кафедра обчислювальної техніки

ЗАТВЕРДЖУЮ

Завідувач кафедри ОТ

проф., д.т.н.. Азаров О.Д.

" " 2022 р.

ТЕХНІЧНЕ ЗАВДАННЯ

на виконання бакалаврськоїдипломної роботи

“Система автоматизованого створення словника з англійської мови”

08-23.БДР.015.00.000 ПЗ

Науковий керівник: доцент к.т.н.

_____ СнігурА.В.

Студент групи 1КІ-18б

_____ ТрошенокО.О.

ВНТУ 2022

1 Підстава для виконання бакалаврської дипломної роботи (БДР)

1.1Важливимєактуальність дослідження у напрямку бакалаврської роботи, яка обумовлена тим, що в даний час англійська мова набуває все більшого поширення, через що збільшується необхідність у додатках для роботи із словниками задля перекладу слів та речень, а також для початку вивчення або покращення знань в англійській мові;

1.2 Наказ про затвердження теми БДР.

2 Мета БДРі призначення розробки

2.1 Мета роботи— аналіз сучасного стану програм для створення словників та розробка програмного засобудля автоматизованого створення словників із англійської мови, що будуть зчитувати дані із файлу;

2.2 Призначення розробки —система автоматизованого створення словника, що може брати слова як із тексту, що вводиться користувачем, так і з текстових файлів формату PDFта TXT. При цьому, слова із перекладом будуть автоматично записуватися в новий файл.

3 Вихідні дані для виконання БДР

3.1 Проведення аналізу існуючих принципів та технологійстворення різних типів словників;

3.2 Розробка структурноїта модульної схем програмної системи для створення словника;

3.3 На основі структурних та модульної схем здійснено розробку додатку, а також функціонування перекладу за допомогою API;

3.4 Тестування програмного засобу для перевірки правильності виконання всіх функцій додатку;

4 Вимоги до виконання БДР

Головна вимога — програма має мати можливість зчитування тексту із клавіатури або ж файлу формату TXTабо PDF, який користувач обирає самостійно, після чого перекладати його із виводом результатів перекладу у двох текстових вікнах із словами англійської та української мови,після чого заносити готовий результат у текстовий словник.

5 Етапи БКРта очікувані результати

Етапи роботи та очікувані результати приведено в Таблиці А.1.

Таблиця А.1 — Етапи БДР

№ етапу	Назва етапу	Термін виконання		Очікувані результати
		початок	кінець	
1	Огляд і аналіз існуючих методів та рішень побудови систем автоматизованого створення словників із англійської мови	13.03.22	16.03.22	Аналітичний огляд літературних джерел, задачі досліджень, розділ 1 ПЗ
2	Дослідження особливостей методів розробки словників	16.03.22	18.03.22	Розділ 2
3	Обґрунтування вибору середовища розробки, розробка блок-схеми додатку та його програмного коду	20.03.22	19.04.22	Розділ 3
4	Тестування додатку	24.04.22	27.04.22	Розділ 4
5	Апробація та впровадження результатів дослідження	27.04.22	29.04.22	Тези доповідей
6	Оформлення пояснювальної записки, графічного матеріалу іпрезентації	05.05.22	01.06.22	Пояснювальна записка, графічний матеріал іпрезентація
7	Підготовка супроводжуючих документів, їх підписування, проходження нормоконтролю та тесту на плагіат	10.06.22	16.06.22	Оформлені документи

6 Матеріали, що подаються до захисту БДР

До захисту подаються: пояснювальна записка БДР, графічні і ілюстративні матеріали, протокол попереднього захисту БДРна кафедрі, відгук наукового керівника, відгукрецензента, протоколи складання державних екзаменів, анотації до

БДРукраїнською та іноземною мовами, довідка про відповідність оформлення БДРдіючим вимогам.

7 Порядок контролю виконання та захисту БДР

Виконання етапів графічної та розрахункової документації БДРконтролюється науковим керівником згідно зі встановленими термінами. Захист БДРвідбувається на засіданні Екзаменаційної комісії, затвердженої наказом ректора.

8 Вимоги до оформлюваннята порядок виконання БДР

8.1 При оформлюванніБДРвикористовуються:

— ДСТУ 3008 : 2015 «Звіти в сфері науки і техніки. Структура та правила оформлювання»;

— ДСТУ 8302 : 2015 «Бібліографічні посилання. Загальні положення та правила складання»;

— ГОСТ 2.104-2006 «Єдина система конструкторської документації. Основні написи»;

— документами на які посилаються у вище вказаних.

ДОДАТОКБ

Лістинг розміткиінтерфейсу додатку

```
<?xml version="1.0" encoding="UTF-8"?>
<?import javafx.scene.control.Button?>
<?import javafx.scene.control.TextArea?>
<?import javafx.scene.control.TextField?>
<?import javafx.scene.control.ToolBar?>
<?import javafx.scene.layout.AnchorPane?>
<AnchorPane maxHeight="-Infinity" maxWidth="-Infinity" minHeight="-Infinity" min-
Width="-Infinity" prefHeight="465.0" prefWidth="640.0"
xmlns="http://javafx.com/javafx/18" xmlns:fx="http://javafx.com/fxml/1"
fx:controller="com.example.test2.HelloController">
<children>
<ToolBar layoutX="14.0" layoutY="14.0" prefHeight="40.0" prefWidth="575.0">
<items>
<Button fx:id="ReadPDF" mnemonicParsing="false" onAction="#OnReadPDF"
text="Read PDF" />
<Button fx:id="ReadTXT" mnemonicParsing="false" onAction="#OnReadTXT"
text="Read TXT" />
<Button fx:id="Help" mnemonicParsing="false" onAction="#OnClickHelp" pref-
Height="25.0" prefWidth="55.0" text="Help" />
<Button fx:id="About" mnemonicParsing="false" onAction="#OnClickAbout" pref-
Height="25.0" prefWidth="55.0" text="About" />
</items>
</ToolBar>
<TextField fx:id="TextField1" layoutX="14.0" layoutY="336.0" prefHeight="49.0"
prefWidth="445.0" promptText="Text" />
```

```
<TextArea fx:id="Word" editable="false" layoutX="14.0" layoutY="84.0" pref-
Height="236.0" prefWidth="270.0" promptText="English text" />
<Button fx:id="TranslateButton" layoutX="476.0" layoutY="336.0" mnemonicPars-
ing="false" onAction="#Translate" prefHeight="49.0" prefWidth="113.0"
text="Translate" />
<TextArea fx:id="Output" editable="false" layoutX="316.0" layoutY="84.0" pref-
Height="236.0" prefWidth="270.0" promptText="Ukrainian text" />
<TextArea fx:id="Log" editable="false" layoutX="14.0" layoutY="402.0" pref-
Height="49.0" prefWidth="445.0" promptText="Log" />
<Button fx:id="Insert" layoutX="476.0" layoutY="402.0" mnemonicParsing="false" on-
Action="#OnInsert" prefHeight="49.0" prefWidth="113.0" text="Insert" />
</children>
</AnchorPane>
```

ДОДАТОКВ

Лістинг конфігурації бібліотек

```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0"
         xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
         xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
https://maven.apache.org/xsd/maven-4.0.0.xsd">
<modelVersion>4.0.0</modelVersion>
<groupId>com.example</groupId>
<artifactId>Test2</artifactId>
<version>1.0-SNAPSHOT</version>
<name>Test2</name>
<properties>
<project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
<junit.version>5.8.2</junit.version>
</properties>
<dependencies>
<!-- https://mvnrepository.com/artifact/org.apache.pdfbox/pdfbox -->
<!-- https://mvnrepository.com/artifact/org.apache.poi/poi -->
<dependency>
<groupId>org.apache.poi</groupId>
<artifactId>poi</artifactId>
<version>5.2.2</version>
</dependency>
<dependency>
```

<groupId>org.apache.poi</groupId>

<artifactId>poi</artifactId>

<version>5.2.2</version>

</dependency>

<dependency>

<groupId>org.apache.poi</groupId>

<artifactId>poi-ooxml</artifactId>

<version>5.2.2</version>

</dependency>

<dependency>

<groupId>org.apache.poi</groupId>

<artifactId>poi-ooxml</artifactId>

<version>5.2.2</version>

</dependency>

<dependency>

<groupId>org.apache.pdfbox</groupId>

<artifactId>pdfbox</artifactId>

<version>2.0.26</version>

</dependency>

<dependency>

<groupId>org.openjfx</groupId>

<artifactId>javafx-controls</artifactId>

<version>18</version>

</dependency>

<dependency>

```
<groupId>org.openjfx</groupId>
<artifactId>javafx-fxml</artifactId>
<version>18</version>
</dependency>
<dependency>
<groupId>org.junit.jupiter</groupId>
<artifactId>junit-jupiter-api</artifactId>
<version>${junit.version}</version>
<scope>test</scope>
</dependency>
<dependency>
<groupId>org.junit.jupiter</groupId>
<artifactId>junit-jupiter-engine</artifactId>
<version>${junit.version}</version>
<scope>test</scope>
</dependency>
</dependencies>
<build>
<plugins>
<plugin>
<groupId>org.apache.maven.plugins</groupId>
<artifactId>maven-compiler-plugin</artifactId>
<version>3.10.1</version>
<configuration>
<source>17</source>
```

```
<target>17</target>
</configuration>
</plugin>
<plugin>
<groupId>org.openjfx</groupId>
<artifactId>javafx-maven-plugin</artifactId>
<version>0.0.8</version>
<executions>
<execution>
<!-- Default configuration for running with: mvn clean javafx:run -->
<id>default-cli</id>
<configuration>
<mainClass>com.example.test2/com.example.test2.HelloApplication</mainClass>
<launcher>app</launcher>
<jlinkZipName>app</jlinkZipName>
<jlinkImageName>app</jlinkImageName>
<noManPages>true</noManPages>
<stripDebug>true</stripDebug>
<noHeaderFiles>true</noHeaderFiles>
<compilerArgs>
                        --add--export java.base/sun.security=All-UNNAMED
</compilerArgs>
</configuration>
</execution>
</executions>
```

```
</plugin>
</plugins>
</build>
</project>
```

ДОДАТОКГ

Лістинг функціонування запуску додатку

```
package com.example.test2;
        import javafx.application.Application;
        import javafx.fxml.FXMLLoader;
        import javafx.scene.Parent;
        import javafx.scene.Scene;
        import javafx.scene.image.Image;
        import javafx.stage.Stage;
    import java.io.IOException;
public class HelloApplication extends Application {
    @Override
    public void start(Stage stage) throws Exception {
        Parent root = FXMLLoader.load(getClass().getResource("hello-view.fxml"));
        Scene scene = new Scene(root);
        stage.getIcons().add(new Image("exelogo.png"));;
        stage.setTitle("Auto Dictionary");
        stage.setScene(scene);
        stage.show();;
    }
    public static void main(String[] args) {
        launch(args);
    }
}
```

ДОДАТОКД

Лістинг функціонування додатку

```
package com.example.test2;

import javafx.event.ActionEvent;

import javafx.fxml.FXML;

import javafx.fxml.Initializable;

import javafx.scene.Group;

import javafx.scene.Scene;

import javafx.scene.control.*;

import javafx.scene.image.Image;

import javafx.scene.image.ImageView;

import org.apache.pdfbox.pdmodel.PDDocument;

import org.apache.pdfbox.text.PDFTextStripper;

import javafx.stage.FileChooser;

import javafx.stage.Stage;

import org.apache.poi.hssf.usermodel.HSSFSheet;

import org.apache.poi.hssf.usermodel.HSSFWorkbook;

import org.apache.poi.ss.usermodel.*;

import org.apache.poi.ss.usermodel.Cell;

import org.apache.poi.util.ArrayUtil;

import org.apache.poi.xssf.usermodel.ListAutoNumber;

import org.apache.poi.xssf.usermodel.XSSFRow;

import org.apache.poi.xssf.usermodel.XSSFSheet;

import org.apache.poi.xssf.usermodel.XSSFWorkbook;

import java.io.*;
```

```
import java.lang.reflect.Array;
import java.net.HttpURLConnection;
import java.net.URL;
import java.net.URLEncoder;
import java.nio.channels.ByteChannel;
import java.util.*;
import java.io.File; import java.io.IOException;
import java.util.stream.Collectors;
public class HelloController implements Initializable {
    @FXML
    public TextArea Log;
    @FXML
    public TextField TextField1;
    @FXML
    public TextArea Word;
    @FXML
    public TextArea Output;
@FXML
public static String translate(String langFrom, String langTo, String text) throws IOExcep-
tion {
    // Script URL
    String urlStr =
"https://script.google.com/macros/s/AKfycbySwhmXt51xF0bXmGiXY7h5RwpxMFAHD
TxBcIG0E3xPwwrra1s9/exec" +
            "?q=" + URLEncoder.encode(text, "UTF-8") +
            "&target=" + langTo +
```

```
            "&source=" + langFrom;
    URL url = new URL(urlStr);
    StringBuilder response = new StringBuilder();
    HttpURLConnection con = (HttpURLConnection) url.openConnection();
    con.setRequestProperty("User-Agent", "Mozilla/5.0");
    BufferedReader in = new BufferedReader(new InputStreamRead-
er(con.getInputStream()));
    String inputLine;
    while ((inputLine = in.readLine()) != null) {
        response.append(inputLine);
    }
    in.close();
    return response.toString();
}
    @Override
    public void initialize(URL url, ResourceBundle rb) {
    }
    @FXML
    public void OnReadPDF(ActionEvent actionEvent) throws IOException{
        TextField1.setText("");
     Stage stage = new Stage();
        FileChooser fil_chooser = new FileChooser();
        File file = fil_chooser.showOpenDialog(stage);
        if (file != null) {
            Log.setText("PDF file is selected");
```

```
    } //PDF Chooser
    PDDocument document = PDDocument.load(file);
     if (!document.isEncrypted()) {
         PDFTextStripper stripper = new PDFTextStripper();
         String text34 = stripper.getText(document);
        TextField1.setText(text34);
         String texx = TextField1.getText();
         String result = texx.replace(".", "");
         TextField1.setText(result);
     }
     else {
         Log.setText("Document is encrypted");
     }
     document.close();
    }
@FXML
public void OnReadTXT(ActionEvent actionEvent) throws IOException {
    TextField1.setText("");
    Stage stage = new Stage();
    FileChooser fil_chooser = new FileChooser();
    File file = fil_chooser.showOpenDialog(stage);
    if (file != null) {
       Log.setText("TXT file is selected");
    }
    //TXT Chooser
```

```
    BufferedReader br = null;
    String c = '', '';
    try {
        br = new BufferedReader(new FileReader(file));
        String line;
        while((line = br.readLine() )!= null){
TextField1.appendText(line);
            Log.setText("TXT file is selected");
        }
        String texx = TextField1.getText();
        String result = texx.replace(".", "");
 TextField1.setText(result);
    }
    catch (IOException e){
        Log.setText("Error");
    }
    finally {
        try {
            br.close();
        }
        catch (IOException e)
            {
                Log.setText("Error");
            }
    }
```

```
}
@FXML
public void OnClickHelp(ActionEvent actionEvent) throws IOException{
    Stage stage = new Stage();
Image image = new Image("help.png");
stage.getIcons().add(new Image("qlogo.png"));
ImageView imageView = new ImageView(image);
//position
imageView.setX(50);
imageView.setY(25);
//height width
imageView.setFitHeight(1009);
imageView.setFitWidth(900);
imageView.setPreserveRatio(true);
Group root = new Group(imageView);
Scene scene = new Scene(root, 1009, 610);
stage.setTitle("Help");
stage.setScene(scene);
//Display
stage.show();
Log.setText("Success");
}
@FXML
public void OnClickAbout(ActionEvent actionEvent) {
```

```
        Alert alert = new Alert(Alert.AlertType.INFORMATION, "Ця програма була створена Олександром Трошенком " +
"1КІ-18б. Це програма для автоматизованого створення словника з англійської мови.");
alert.setHeaderText(null);
  alert.setTitle("About");
        alert.show();
        Log.setText("Success");
    }
    @FXML
    public void Translate(ActionEvent actionEvent) throws IOException{
    String text = TextField1.getText();
      String result = text.replace(",", "").replace(" a ", " ")
              .replace(" the ", " ").replace(".", "").replace("?", "")
               .replace("!", "").replace(":", "").replace(";", "")
              .replace("-", "")
                .replace("=", "").replace(" ", ". ").replaceAll("&#39;", "'");
        Word.setText(result);
      String res = (translate("en", "uk", result) );
      String fin =res.replace("&#39;","'");
      Output.setText(fin);
      Log.setText("Success");
    }
    @FXML
    public void OnInsert(ActionEvent actionEvent) throws IOException{
        try {
```

```
        File filee = new File("New file.txt");
        if (!filee.exists()) {
            filee.createNewFile();
Log.setText("File is created");
        }
        else
            Log.setText("File was already created or unexpected error");
        PrintWriter pw = new PrintWriter(new FileOutputStream(filee));
        String po1 = Word.getText();
        String[] ss1 = po1.split(" ");
        String po2 = Output.getText();
        String[] ss2 = po2.split(" ");
       int n1 = ss1.length;
        int n = TextField1.getText().length();
       int n2 = ss2.length;
   for(int i = 0; i<n1; i++) {
                pw.write(ss1[i] + " " + ss2[i] + "\n");
        }
        Log.setText("Success");
        pw.close();
      }
      catch (IOException e)
      {
        Log.setText("Error");
      }
```

```
  } }
```

ДОДАТОКЕ

Лістинг скрипту перекладу

```
var mock = {
  parameter:{
    q:'hello',
    source:'en',
    target:'uk'
  }
};
function doGet(e) {
  e = e || mock;
  var sourceText = ''
  if (e.parameter.q){
    sourceText = e.parameter.q;
  }
  var sourceLang = '';
  if (e.parameter.source){
    sourceLang = e.parameter.source;
  }
  var targetLang = 'en';
  if (e.parameter.target){
    targetLang = e.parameter.target;
  }
  var translatedText = LanguageApp.translate(sourceText, sourceLang, targetLang, {conte
ntType: 'html'});
```

```
 re-
turn ContentService.createTextOutput(translatedText).setMimeType(ContentService.Mim
eType.JSON);
}
```

ДОДАТОК Ж

ПРОТОКОЛ
ПЕРЕВІРКИ КВАЛІФІКАЦІЙНОЇ
РОБОТИНАНАЯВНІСТЬТЕКСТОВИХЗАПОЗИ
ЧЕНЬ

Назвароботи: «Система автоматизованого створення словника з англійської мови»

Типроботи: бакалаврська дипломна робота
(БДР,МКР)

Підрозділ кафедра обчислювальної техніки
(кафедра,факультет)

ПоказникизвітуподібностіUnicheck

Оригінальність 99% Схожість 1,0%

Аналіззвітуподібності(відмітитипотрібне):

- ✓ Запозичення, виявлення у роботі, оформлені коректно і не містять ознак плагіату.
- □ Виявлені у роботі запозичення не мають ознак плагіату, але їх надмірна кількість викликає сумніви щодо цінності роботи і відсутності самостійності її виконання автором. Роботу направити на розгляд експертної комісії кафедри.
- □ Виявлені у роботі запозичення є недобросовісними і мають ознаки плагіату та/або в ній містяться навмисні спотворення тексту, що вказують на спроби приховування недобросовісних запозичень.

Особа,відповідальназаперевірку _____ Захарченко С.М.
(підпис) (прізвище,ініціали)

Ознайомленізповнимзвітомподібності,якийбувзгенерованийсистемоюUnicheck щодороботи.

Авторроботи _____ Трошенок О. О
(підпис) (прізвище,ініціали)

Керівникроботи _____ Снігур А. В.
(підпис) (прізвище,ініціали)