

Проектування підсистеми автоматизованого створення словника з англійської мови

Вінницький національний технічний університет

Анотація

Проаналізовано особливості програмної розробки системи автоматизованого створення словника з англійської мови.

Ключові слова: програма, англійська мова, інтерфейс, Java, словник, мова програмування.

Abstract

The features of the software development of the system for the automated creation of a dictionary from English language are analyzed..

Keywords: program, English language, interface, Java, dictionary, programming language.

Вступ

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. Перед розробкою програми, у даному випадку системи автоматизованого створення, постають декілька питань, які вимагають рішення. Важливим етапом є вибір мови програмування та розробка структури програми, інтерфейсу.

Основна частина

Розкриємо поставлені питання, що по'язані зі створенням розроблювальної системи. Програма має бути функціональною для полегшення роботи із нею майбутньому користувачу. Багато інформації щодо розробки на ній також будуть плюсом, оскільки полегшують пошук вирішення проблем, які будуть з'являтися під час написання програми. Зручність середовища розробки також буде плюсом, оскільки це також полегшить роботу в написанні програми та її налаштування та майбутніх покращень. Також величезною перевагою мови програмування буде кросплатформність, оскільки в сучасному світі прикладні додатки на базі Android або ж iOS набули значного поширення, яке може бути навіть більшим, ніж програми для Windows. Програма має забезпечуватися зручним та зрозумілим інтерфейсом, знову ж таки для полегшення роботи користувача. Також важливою частиною має бути низькі вимоги до системи для забезпечення широкого застосування додатку як для дорогих та потужних систем, так і для відносно старих комп'ютерів та телефонів. Саме тому, у якості основної мови програмування, була вибрана Java.

На сьогоднішній момент мова Java є однією з найпоширеніших та найпопулярніших мов програмування. Перша версія мови з'явилася ще в 1996 в надрах компанії Sun Microsystems, згодом поглиненої компанією Oracle. Java замислювався як універсальна мова програмування, яку можна застосовувати для різноманітних завдань. І до теперішнього часу мова Java пройшла великий шлях, було видано безліч різних версій.

Ключовою особливістю мови Java є те, що код спочатку транслюється в спеціальний байт-код, незалежний від платформи. А потім цей байт-код виконується віртуальною машиною JVM (Java Virtual Machine). У цьому плані Java відрізняється від стандартних мов, що інтерпретуються, як PHP або Perl, код яких відразу ж виконується інтерпретатором. У той самий час Java не є і чисто компілюємою мовою, як або C++.

Подібна архітектура забезпечує кросплатформність та апаратну переносимість програм на Java, завдяки чому подібні програми без перекомпіляції можуть виконуватись на різних платформах – Windows, Linux, Mac OS тощо. Для кожної з платформ може бути своя реалізація віртуальної машини JVM, але кожна з них може виконувати той самий код. [1]

На противагу C++, Java є більш об'єктно-орієнтованою. Всі дані і дії групуються в класи об'єктів. Виключенням з повної об'єктності (як скажімо в Smalltalk) є примітивні типи (int, float тощо). Це було свідомим рішенням проектувальників мови задля збільшення швидкості. Через це Java не вважається повністю об'єктно-орієнтовною мовою.

У Java всі об'єкти є похідними від головного об'єкта (він називається просто Object), з якого вони успадковують базову поведінку і властивості.

Хоча у C++ вперше стало доступне множинне успадкування, але у Java можливе тільки одинарне успадкування, завдяки чому виключається можливість конфліктів між членами класу (методи і змінні), які успадковуються від базових класів.[2]

Інтерфейс програми має бути зручним як з точки зору користувача, так і з точки розробки. Для користувача важливо створити максимально просту програму у візуальному плані, щоб було зрозуміло призначення будь-якої клавіші «з першого погляду». Так, кожна клавіша має мати простий та лаконічний підпис. Інтерфейс має бути максимально ненавантаженим, без зайвих речей. Колір та тони програми мають бути також простими, щоб полегшити роботу користувача та уникнути зайвого навантаження на очі. Для вирішення проблем та можливих питань потрібно додати меню з допомогою, де буде описано функціональність та призначення кожної клавіші, а також описані можливості програми.

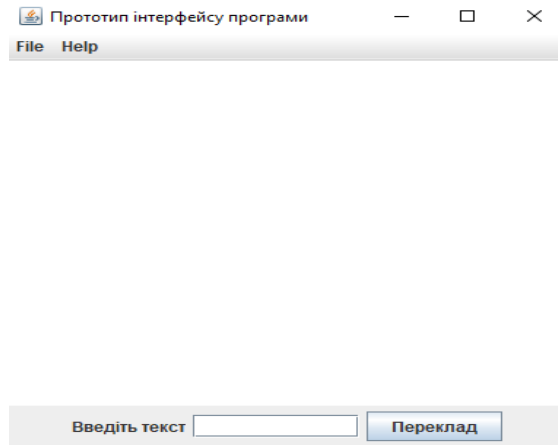


Рисунок 1 – Прототип інтерфейсу програми

На рисунку 1 зображено прототип інтерфейсу розроблюваної програми. У ньому має бути присутнє поле для вводу тексту від користувача, а також клавіша, яка буде запускати функціонал самої програми. На рисунку це клавіша «Переклад». У лівому верхньому кутку наявні дві клавіші – «File» і «Help». Клавішею «Help» викликається меню із описом роботи програми та порадами щодо використання. Клавіша «File» буде відкривати файл із текстом, який необхідно буде перекласти та занести в словник.

Висновки

У доповіді було розглянуто можливості та особливості розробки програми системи автоматизованого створення словника. Так, було проаналізовано вимоги до прикладного додатку та способи їхнього рішення. Було розглянуто особливості та переваги мови Java, яку було обрано для розробки застосунку. Також було наведено приклад прототипу інтерфейсу, який буде вдосконалюватися для подальшого вирішення необхідних задач, пов'язаних із розробкою та використанням додатку майбутнім користувачем.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Java [Електронний ресурс]. URL: <https://uk.wikipedia.org/wiki/Java#Платформа> / (дата звернення 13.05.2022).
2. Введение в Java [Електронний ресурс]. URL: <https://metanit.com/java/tutorial/1.1.php> / (дата звернення 13.05.2022).

Трошенко Олександр Олексійович - студент групи 1KI-18б, факультет інформаційних технологій та комп'ютерної інженерії, Вінницький національний технічний університет, Вінниця, e-mail: mutastak@gmail.com

Снігур Анатолій Васильович – к. т. н. доцент кафедри обчислювальної техніки Вінницького національного технічного університету, Вінниця, e-mail: sanv12@i.ua

Troshenok Oleksandr Oleksiiovych - student of group 1KI-18b, Faculty of Information Technologies and Computer Engineering, Vinnytsia National Technical University, Vinnytsia, e-mail: mutastak@gmail.com

Snigur Anatolii Vasyliovych – PhD (Eng), Associate Professor of the Department of Computer Engineering, Vinnytsia National Technical University, Vinnytsia, e-mail: sanv12@i.ua