

Клавіатурний тренажер для вивчення слів англійської мови

Вінницький національний технічний університет

Анотація

Проаналізовано особливості програмної розробки системи вивчення англійської мови.

Ключові слова: програма, англійська мова, інтерфейс, C#, словник, мова програмування.

Abstract

The features of the software development of the English language production system are analyzed.

Keywords: program, English language, interface, C#, dictionary, programming language.

Вступ

У міру того як розширюється інформатизація сучасного суспільства при переході до суспільства майбутнього, зростає значення прикладної (обчислювальної, комп'ютерної, інженерної) лінгвістики, науки, що знаходиться на стику глибоко людської, гуманітарної науки лінгвістики (мовознавства), що вивчає закони розвитку та користування могутнім засобом мовою, - і комп'ютерного знання, за допомогою якого машині передається все більша частина інтелектуальної праці людини.

Переклад – процес та результат створення на основі вихідного тексту однією мовою рівноцінного йому відношення тексту іншою мовою.

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. 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. Саме тому, у якості основної мови програмування, була вибрана C#.

Мова C# є ООП мовою. В ній все є класом, навіть сама програма. Основні поняття ООП наведено нижче [1].

- **Об'єкт.** Об'єкт можна визначити як сутність, яка має стан і поведінку, або, іншими словами, все, що фізично існує у світі, називається об'єктом. Він може представляти собаку, людину, стіл тощо.

- **Клас.** Клас можна визначити як план об'єкта. По суті, це сукупність об'єктів, які діють як будівельні блоки. Вони містять члени даних (змінні) і функції-члени. Ці функції-члени використовуються для маніпулювання членами даних всередині класу.

- Абстракція. Абстракція допомагає в процесі приховування даних. Це допомагає відобразити основні функції, не показуючи користувачеві деталей чи функцій. Він уникає непотрібної інформації або невідповідних деталей і показує лише ту конкретну частину, яку хоче бачити користувач.

- Інкапсуляція. Об'єднання даних і функцій в один блок відоме як інкапсуляція. Цього можна досягти, зробивши область членів даних приватною, а область функції-члена — загальнодоступною для доступу до цих членів даних. Інкапсуляція робить дані недоступними для зовнішнього світу.

- Спадкування. Спадкування – це процес, у якому два класи мають взаємозв'язок між собою, а об'єкти одного класу набувають властивостей та ознак іншого класу. Клас, який успадковує ознаки, відомий як дочірній клас, а клас, ознаки якого він успадкував, називається батьківським класом. Наприклад, Class Vehicle є батьківським класом, а Class Bus, Car і Vike є дочірніми класами.

- Поліморфізм. Поліморфізм означає багато форм. Це здатність приймати більше ніж одну форму. Це функція, яка надає функцію або оператор з більш ніж одним визначенням. Його можна реалізувати за допомогою перевантаження функцій, перевантаження оператора, перевизначення функції, віртуальної функції.

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.

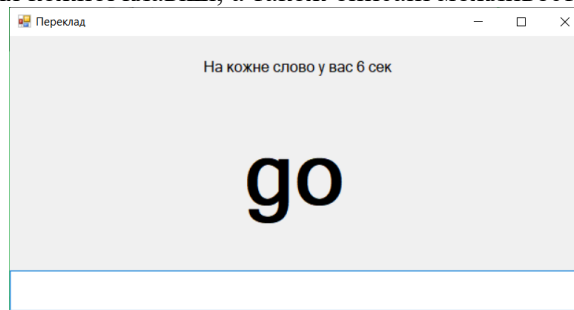


Рисунок 1 – Прототип інтерфейсу програми

На рисунку 1 зображено прототип інтерфейсу розроблюваної програми. У ньому має бути присутнє поле для вводу тексту від користувача. Користувач вводить переклад та для відправки жме Enter.

Висновки

У доповіді було розглянуто можливості та особливості розробки програми вивчення англійської мови. Так, було проаналізовано вимоги до прикладного додатку та способи їхнього рішення. Було розглянуто особливості та переваги мови C#, 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. Антони Синтес. Освой самостоятельно объектно-ориентированное программирование за 21 день = Sams Teach Yourself Object-Oriented Programming in 21 Days. — М.: «Вильямс», 2002. — С. 672. — ISBN 0-672-32109-2.

Крупський Артур Олексійович - студент групи КІ-20мс3, факультет інформаційних технологій та комп'ютерної інженерії, Вінницький національний технічний університет, Вінниця, e-mail: artur245@ukr.net
Снігур Анатолій Васильович – к. т. н. доцент кафедри обчислювальної техніки Вінницького національного технічного університету, Вінниця, e-mail: sanv12@i.ua

Krupskiy Artur Oleksiiovych - student of group KI-20ms3, Faculty of Information Technologies and Computer Engineering, Vinnytsia National Technical University, Vinnytsia, e-mail: artur245@ukr.net

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