

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА»

Кафедра комп'ютерних наук

КУРСОВА РОБОТА
(ПОЯСНЮВАЛЬНА ЗАПИСКА)

з дисципліни: «Об'єктно-орієнтоване програмування»
на тему:

«Тренажер швидкості друку»

студента І курсу групи КН-21-1
спеціальності 122 «Комп'ютерні науки»
Баковецький Максим Олександрович
(прізвище, ім'я та по-батькові)

Керівник ст. викл. каф. КН Марчук Г.В.

Дата захисту: " ____ " _____ 2022 р.
Національна шкала _____
Кількість балів: _____
Оцінка: ECTS _____

Члени комісії

_____	<u>В.Л. Левківський</u>
(підпис)	(прізвище та ініціали)
_____	<u>Г.В. Марчук</u>
(підпис)	(прізвище та ініціали)
_____	_____
(підпис)	(прізвище та ініціали)

Житомир – 2022

		Баковецький М.О.			«Житомирська політехніка».22.122.04.000 – ПЗ	Арк.
		Марчук Г. В.				
Змн.	Арк.	№ докум.	Підпис	Дата		1

ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА»
Факультет інформаційно-комп'ютерних технологій
Кафедра комп'ютерних наук
Освітній рівень: бакалавр
Спеціальність 122 «Комп'ютерні науки»

«ЗАТВЕРДЖУЮ»
Зав. кафедри
_____ М.С. Граф
“ ____ ” _____ 2022р.

ЗАВДАННЯ
НА КУРСОВУ РОБОТУ СТУДЕНТУ

1. Тема роботи: Тренажер швидкості друку,
керівник курсового проекту: ст. викл. каф. КН Марчук Г.В.
2. Строк подання студентом: “ 08 ” червня 2022р.
3. Вхідні дані до роботи: _____
4. Зміст розрахунково-пояснювальної записки(перелік питань. Які підлягають розробці)
1. Аналіз аналогічних розробок;
2. Алгоритми роботи програми;
3. Опис роботи програми.
5. Перелік графічного матеріалу(з точним зазначенням обов'язкових креслень)

6. Консультанти розділів проекту (роботи)

Розділ	Прізвище, ініціали та посади консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1,2	Левківський В.Л., ст. викладач каф. КН	дата	дата
1,2	Марчук Г.В., ст. викладач каф. КН	дата	дата

7. Дата видачі завдання “ 15 ” квітня 2022 р.

		Баковецький М.О.			«Житомирська політехніка».21.122.04.000 – ПЗ	Арк.
		Марчук Г. В.				2
Змн.	Арк.	№ докум.	Підпис	Дата		

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів курсового проекту	Строк виконання етапів проекту	Примітки
1	Постановка задачі	20.05.2022	виконано
2	Пошук, огляд та аналіз аналогічних розробок	21.05.2022	виконано
3	Формулювання технічного завдання	25.05.2022	виконано
4	Опрацювання літературних джерел	02.06.2022	виконано
5	Проектування структури	07.06.2022	виконано
6	Написання програмного коду	23.06.2022	виконано
7	Налагодження	23.06.2022	виконано
8	Написання пояснювальної записки	01.07.2022	виконано
9	Захист	05.07.2022	виконано

Студент _____

—

(підпис)

Баковецький М.О.

(прізвище та ініціали)

Керівник проекту _____

(підпис)

Марчук Г.В.

(прізвище та ініціали)

		Баковецький М.О.			«Житомирська політехніка».21.122.04.000 – ПЗ	Арк.
		Марчук Г. В.				3
Змн.	Арк.	№ докум.	Підпис	Дата		

РЕФЕРАТ

Пояснювальна записка до курсової роботи на тему «Тренажер друку тексту» складається з вступу, трьох розділів, висновків, списку використаної літератури та додатку.

Текстова частина викладена на 33 сторінках друкованого тексту.

Пояснювальна записка має 12 сторінок додатків. Список використаних джерел містить 10 найменувань і займає 1 сторінку. В роботі наведено 26 рисунків, із них 3 блок-схеми та одна діаграма. Загальний обсяг роботи – 45 сторінок.

КЛЮЧОВІ СЛОВА: ТЕКСТ, ДРУК, СИМВОЛИ, ШВИДКІСТЬ, ТОЧНІСТЬ, РЕКОРДИ.

		Баковецький М.О.			«Житомирська політехніка».21.122.04.000 – ПЗ	Арк.
		Марчук Г. В.				
Змн.	Арк.	№ докум.	Підпис	Дата		4

ЗМІСТ

ВСТУП **Ошибка! Закладка не определена.**

РОЗДІЛ 1. АНАЛІЗ ПРОБЛЕМАТИКИ, МЕТОДІВ ТА ЗАСОБІВ

ВИРІШЕННЯ ЗАДАЧІ **Ошибка! Закладка не определена.**

1.1 Аналіз задачі, засобів та методів її вирішення..... **Ошибка! Закладка не определена.**

1.2 Аналіз існуючого програмного забезпечення за тематикою курсового проекту **Ошибка! Закладка не определена.**

Висновки до першого розділу..... **Ошибка! Закладка не определена.**

РОЗДІЛ 2. ПРОЕКТУВАННЯ ТА РОЗРОБКА ПРОГРАМНОГО

ЗАБЕЗПЕЧЕННЯ **Ошибка! Закладка не определена.**

2.1 Проектування загального алгоритму роботи програми **Ошибка! Закладка не определена.**

2.2 Розробка функціональних алгоритмів роботи програми **Ошибка! Закладка не определена.**

2.3 Розробка програмного забезпечення. **Ошибка! Закладка не определена.**

Висновки до другого розділу **Ошибка! Закладка не определена.**

РОЗДІЛ 3. ОПИС РОБОТИ З ПРОГРАМНИМ ДОДАТКОМ ТА ЙОГО

ТЕСТУВАННЯ **Ошибка! Закладка не определена.**

3.1 Опис роботи з програмним додатком **Ошибка! Закладка не определена.**

3.2 Тестування роботи програмного забезпечення..... **Ошибка! Закладка не определена.**

Висновки до третього розділу..... **Ошибка! Закладка не определена.**

ВИСНОВКИ..... **Ошибка! Закладка не определена.**

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ. **Ошибка! Закладка не определена.**

ДОДАТКИ..... **Ошибка! Закладка не определена.**

		Баковецький М.О.			«Житомирська політехніка».21.122.04.000 – ПЗ	Арк.
		Марчук Г. В.				5
Змн.	Арк.	№ докум.	Підпис	Дата		

ВСТУП

Актуальність теми. Мова програмування C# була створена відносно недавно – в кінці 1998 року командою спеціалістів Microsoft. Її ціллю була можливість створення різноманітних програм для платформ Microsoft.NET. Така прив’язаність платформ Microsoft .NET та мови C# в подальшому була усунена, проте сама платформа .NET у всіх версіях містить компілятор кодів даної мови програмування. Це дозволяє запускати додатки без налаштування додаткового програмного забезпечення.

Сама назва «C#» пов’язана з тим, що її синтаксис дуже нагадує C++. Загалом, C# перейняв багато позитивних рис своїх попередників - Delphi, C++, Java та інших. При цьому із C# були вилучені проблемні алгоритми.

Мова C# актуальна в першу чергу тому, що дозволяє більш раціонально створювати популярні на сьогодні інтернет-додатки. C# тісно інтегрована з мовою XML, різноманітними веб-технологіями. Також популярність C# серед розробників обумовлена тим, що вона інтегрувала в собі переваги мов Java та C++, при чому з C# були виключені деякі сумнівні директиви, макроси, відмінені глобальні перемінні.

Сучасні умови економічного розвитку вимагають від працівників різноманітних навичок. Сучасні компанії або кадрові агентства одним із пунктів прийняття на роботу вимагають від працівника високих навичок друку тексту(від 120 символів в хвилину і більше),адже відсутність достатніх вмінь при роботі з клавіатурою є однією із головних причин малоефективної та довготривалої роботи.

Метою розробки курсової роботи є створення додатку для тренування швидкості друку тексту, а також удосконалення навичок роботи із мовою “C#”,

		Баковецький М.О.			«Житомирська політехніка».21.122.04.000 – ПЗ	Арк.
		Марчук Г. В.				
Змн.	Арк.	№ докум.	Підпис	Дата		6

класами та методами. Поставлено завдання: реалізувати додаток з наступними функціями:

- Визначення швидкості друку тексту
- Визначення точності друку
- Запис рекордів у файл
- Взяття тексту для друку із бази даних
- Вибір довжини тексту для друку

Об'єктом дослідження є технологія роботи з Windows Presentation Foundation.

Предметом дослідження Робота с конструкторами класів, їх об'єктами, елементами керування WF, робота із базами даних.

		Баковецький М.О.			«Житомирська політехніка».21.122.04.000 – ПЗ	Арк.
		Марчук Г. В.				7
Змн.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 1. АНАЛІЗ ПРОБЛЕМАТИКИ, МЕТОДІВ ТА ЗАСОБІВ ВИРІШЕННЯ ЗАДАЧІ

1.1 Аналіз задачі, засобів та методів її вирішення

Тематикою курсової роботи є “Тренування швидкості друку тексту”.

Коли йдеться про переваги друку наосліп, зазвичай говорять про високу швидкість набору. Друкувати швидко — значить економити час, якого так не вистачає. Плюси швидкісного друку досить очевидні, проте не всім вони мають критичну значимість. Дехто може цілком резонно сказати: «Мені не потрібно друкувати швидко, тому що у мене немає необхідності набирати великі тексти, а для листування та інших побутових потреб цілком достатньо дивитися на кнопки і натискати їх одним пальцем» — і по-своєму мають рацію, адже їм так зручно. Але справа не лише у швидкості як такої — є два дуже важливі моменти, які стосуються здоров'я, які чомусь залишаються поза увагою.

Перший із них — це наші очі. Якщо набирати довгий текст, весь час переводячи погляд з монітора на кнопки і назад, очі дуже швидко починають втомлюватися. Причина в тому, що відстань від клавіатури до екрана зазвичай різна, відбувається постійне перефокусування. До того ж, положення очей при погляді на клавіші часто близьке до крайнього нижнього, в якому їх непросто утримувати (це, звичайно, залежить від розташування клавіатури). При поганому висвітленні ситуація ще гірша — дається взнаки велика різниця в яскравості та контрастності об'єктів, між якими переміщується наш погляд. В результаті спочатку можуть просто почати боліти очі, а потім (ймовірно) і зір може погіршитися.

Функціонал програмного додатку:

- Визначення швидкості друку тексту
- Визначення точності друку
- Запис рекордів у файл
- Взяття тексту для друку із бази даних

		Баковецький М.О.			«Житомирська політехніка».21.122.04.000 – ПЗ	Арк.
		Марчук Г. В.				
Змн.	Арк.	№ докум.	Підпис	Дата		8

-Вибір довжини тексту для друку

1.2 Аналіз існуючого програмного забезпечення за тематикою курсової роботи

Для початку хотілося би сказати про програму, на основі якої і буде створюватися програма на курсову роботу. *Typeracer* - багатокористувацька онлайн-гра на основі браузера(рис.1.1).

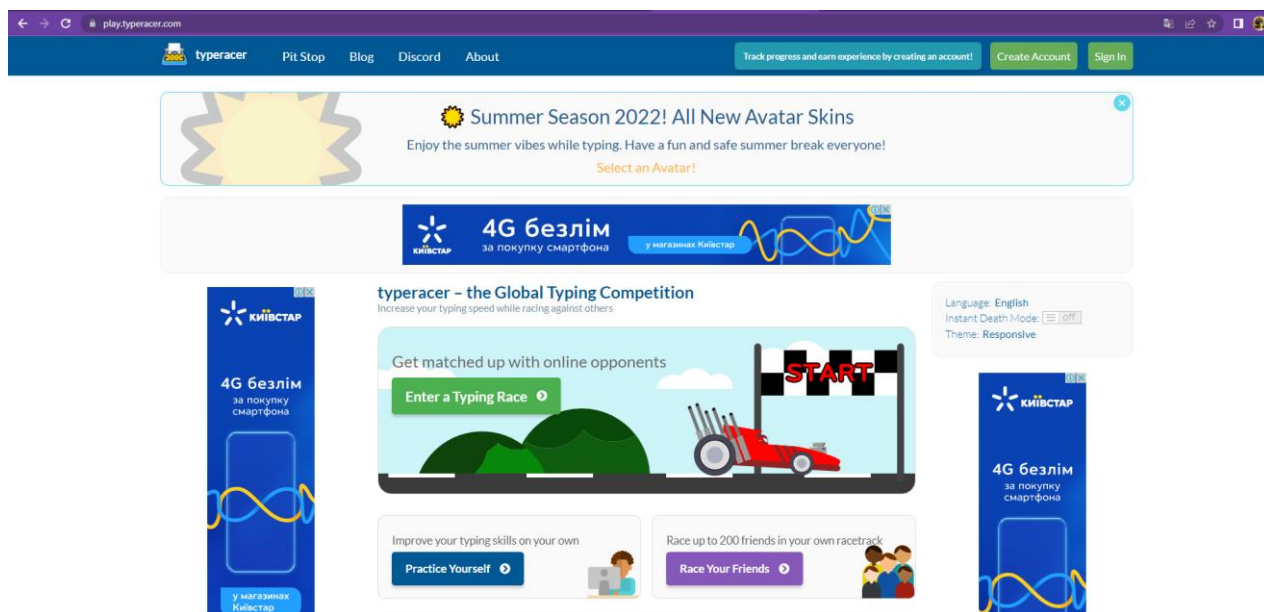


Рис. 1.1. Інтерфейс TypeRacer

У TypeRacer гравці якомога швидше проходять тести набору різних текстів, змагаючись із собою або з іншими користувачами онлайн. Додаток запустили в березні 2008 року. Гравці змагаються в гонках на мініатюрних автомобілях, які рухаються вперед, коли користувачі набирають різні уривки тексту. Довжина яких становить приблизно від 20 до 930 символів.

Для гонок існує параметр за замовчуванням («основна траса»), коли гравці змагаються один проти одного, вводячи випадково вибрані цитати з бази даних.. Тренування в гонках або «привиди» — це варіант гри для одного гравця, де гравці можуть вводити будь-який текст на вимогу та зберігати до п'яти перегонів. Крім того, існують окремі змагання та приватні траси. У змаганнях перемагає гравець, який набрав найбільшу кількість очок до певного часу. У приватних треках гравцям потрібне спеціальне посилання для доступу до треку. Під час введення виділених текстів необхідна точність; будь-які помилки

		Баковецький М.О.			«Житомирська політехніка».21.122.04.000 – ПЗ	Арк.
		Марчук Г. В.				9
Змн.	Арк.	№ докум.	Підпис	Дата		

введення, виявлені в орфографії, великих літерах або пунктуації, повинні бути виправлені гравцем перед продовженням змагання.

Уривки для друку походять із пісень, фільмів, телевізійних шоу, відеоігор та книг. Наприклад, вибрані тексти включають уривки з таких джерел, як «Монті Пайтон і Святий Грааль», «Заводний апельсин» та «Я — Америка» Стівена Колберта (І так ви можете!). Користувачі можуть надіслати додаткові цитати, які будуть розглянуті для додавання в гру. Дана гра є чудовим прикладом того, яку саме програму потрібно розробити на курсовій роботі.

Ще одним гарним прикладом існуючого програмного забезпечення є клавіатурний тренажер Stamina.Stamina – безкоштовний, клавіатурний онлайн тренажер для засвоєння та навчання десятипальцевого сліпого методу друку тексту.

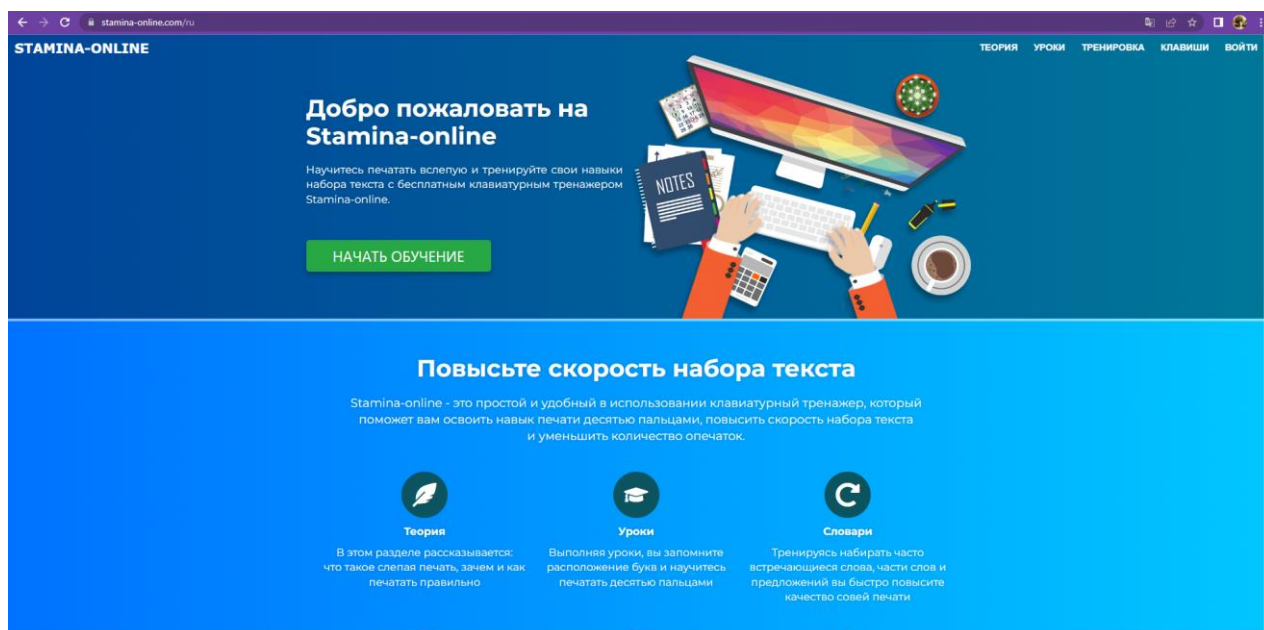


Рис 1.2. Інтерфейс Stamina

На офіційному сайті додатку Stamina дається вибір різних режимів, таких як теорія, уроки, словники, вирази, мови програмування.

Теорія – розділ, в якому розповідається про те що таке сліпий друк, навіщо це потрібно та як друкувати правильно.

Уроки – режим, у якому можна навчитися правильному розташуванню пальців та друку при використанні десяти пальців.

		Баковецький М.О.			«Житомирська політехніка».21.122.04.000 – ПЗ	Арк.
		Марчук Г. В.				
Змн.	Арк.	№ докум.	Підпис	Дата		10

Словники – режим, у якому проходить тренування друку за допомогою слів, або частин слів які часто зустрічаються.

Вирази – режим для збільшення швидкості друку тексту за допомогою різних слів, анекдотів, крилатих виразів, цитат, і так далі.

Мови програмування – режим для практики друку тексту виразів або символів із програмування.

Можна згадати ще одне програмне забезпечення – GNU Typist.



Рис. 1.3. Інтерфейс GNU Typist

Ще один тренажер з широкими можливостями – він підтримує майже кожен наявну мову та розкладку. Опанувати сліпий друк пропонується за 15 уроків, кожен з яких розбито на вправи. Кожен урок створений так, що в перших вправах вивчаються нові клавіші, потім з ними друкуються слова, після чого заняття переходить до набору тексту. Також в уроках міститься коротка підказка. Паралельно з проходженням уроків можна скласти тест на швидкість, також є набір випадкового тексту, ігри з друкуванням. Ще можна додатково опанувати набір на цифровій клавіатурі (блок з цифрами справа).

		Баковецький М.О.			«Житомирська політехніка».21.122.04.000 – ПЗ	Арк.
		Марчук Г. В.				11
Змн.	Арк.	№ докум.	Підпис	Дата		

Висновки до першого розділу

В першому розділі було проаналізовано задачу, методи та засоби її рішення, а саме продумано функції додатку та забезпечення їх правильної роботи. Також було проаналізовано таке існуюче програмне забезпечення як TypeRacer, Stamina, GNU Typist.

		Баковецький М.О.			«Житомирська політехніка».21.122.04.000 – ПЗ	Арк.
		Марчук Г. В.				12
Змн.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 2. ПРОЕКТУВАННЯ ТА РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

2.1 Проектування загального алгоритму роботи програми

Зобразимо загальну структуру роботи додатку, який розробляється.(див.

Рис 2.1)

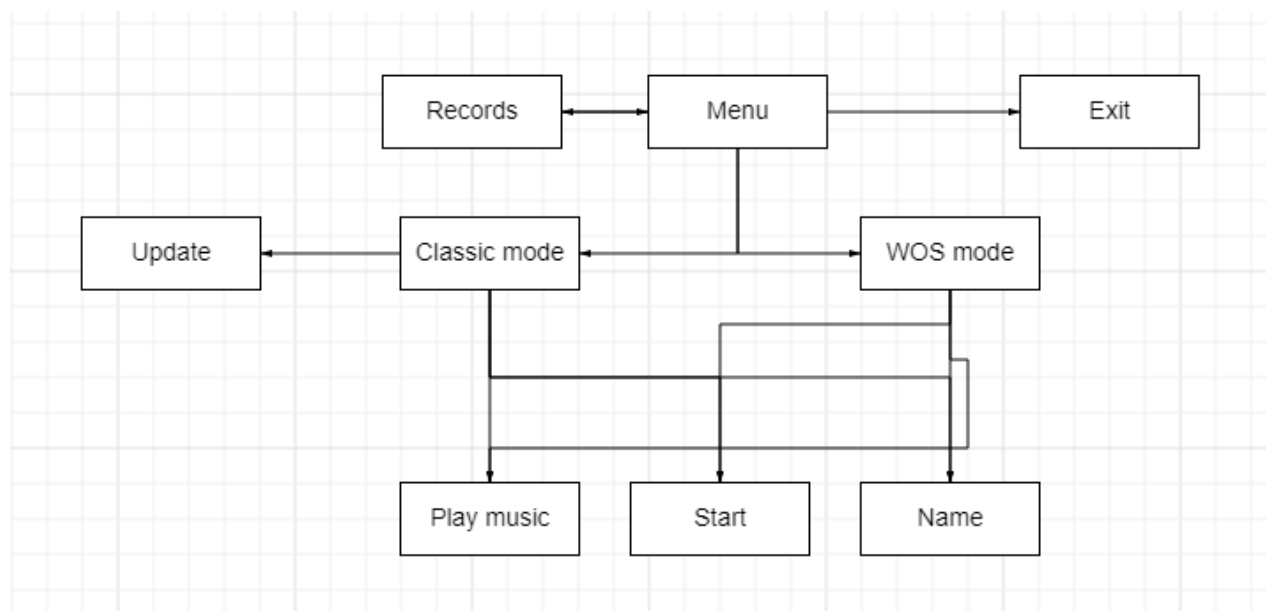


Рис 2.1. Загальна структура

При запуску програми ми потрапляємо у головне меню, звідки і відбувається перехід на усі інші форми. Ми можемо обрати, який режим гри із двох представлених спробувати, або ж ми можемо перейти по на форму “Records”, на якій можна побачити таблицю рекордів у двох режимах. На формі відобразиться результат, нікнейм та дата проходження гри кожного користувача. Також є кнопка “Exit”, тобто вихід із програми. При переході на форми Classic Mode та WOS Mode, ми бачимо кнопку Start-починає гру, checkbox Play music, який відповідає за програвання музики під час гри, та textbox Name, який дозволяє ввести ім’я користувача, або залишити його пустим, кнопку Menu, яка дозволяє повернутися в меню. Також на формі Classic Mode є кнопка Update, яка дозволяє змінити текст для вводу до початку гри. Не забуваємо про таймери на формах: один із них відповідає за відлік часу, тобто за час який залишився до початку гри, а інший за час, який пройшов після початку гри.

Отже, при запуску програми користувач може обрати наступну дію:

- Грати(класичний режим)
- Грати(WOS режим)
- Таблиця рекордів ждя обох режимів
- Змінити мову(Українська,Японська, Англійська)

2.2 Розробка функціональних алгоритмів роботи програми

На рисунку 2.2 можна побачити загальну діаграму класів додатку:

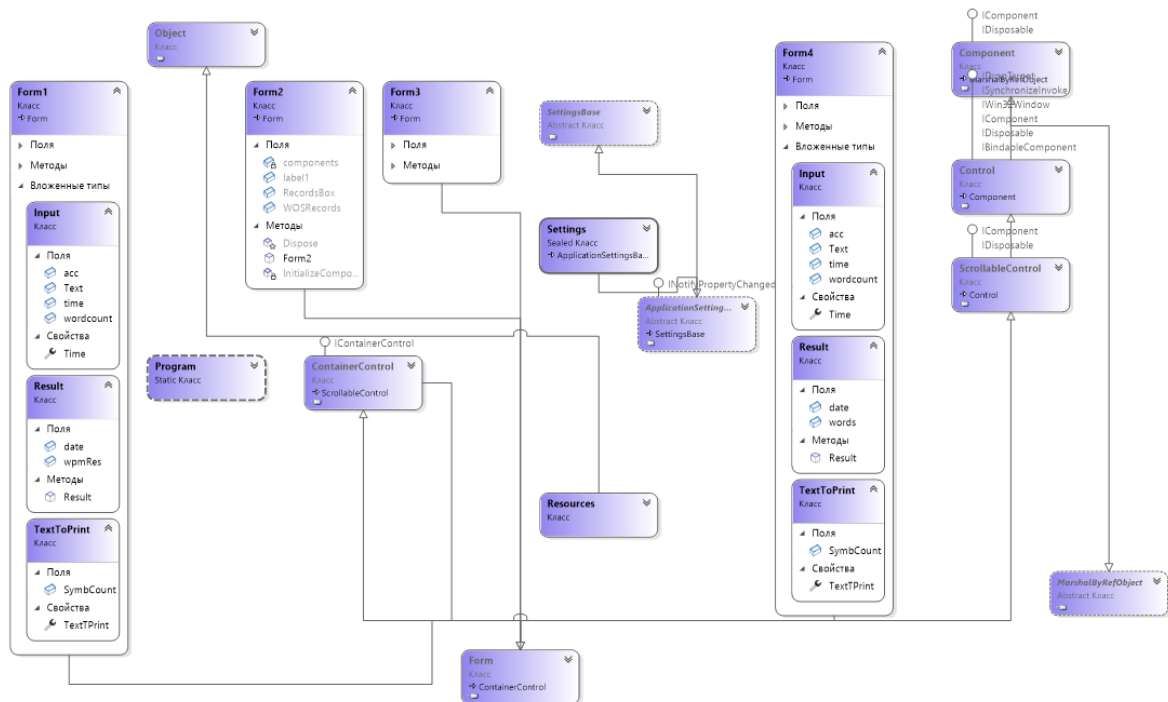


Рис 2.2. Діаграма класів

На діаграмі класів можна побачити залежність та зв'язок між класами. Головним батьківським класом є клас Object, від якого ідуть інші класи, такі як Resources, Settings, Component тощо. Також від нього буде наслідуватися клас Form, який в свою чергу є батьківським класом для класів Form1, Form2, Form3, Form4. Від класів Form1 та Form4 наслідуються такі класи як Input, Result, TextToPrint. Розберемо кожен клас окремо.

Input – клас який зберігає та записує інформацію про введений користувачем текст. Поле “acc” зберігає точність вводу користувачем тексту у відсотках. Поле “Text” – зберігає текст який вводить користувач. “wordcount” –

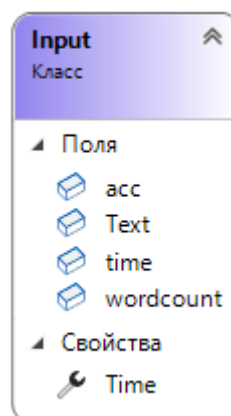
зберігає кількість слів у введеному тексті. Властивість “Time” – зберігає час у секундах, витрачений на друкування тексту у поле time.

Result – зберігає інформацію про результат завершення вводу. Поле “wpmRes” – зберігає швидкість написання тексту. Поле “date” – зберігає дату завершення вводу.

Music – клас для програвання музики. Має 4 методи, які відповідають за повтор пісні, за таймер, за програвання і зупинення музики.

TextToPrint, і його властивість TextTPrint зберігає текст, який потрібно ввести користувачу. Поле “SymbCount” – зберігає кількість символів тексту для вводу.

Основним алгоритмом роботи програми є алгоритм перевірки на правильність введення тексту. За введення та зберігання даних відповідає клас Input.



На формі встановлений таймер, який починає працювати після того, як таймер відліку дойде до нуля. Після цього і починається гра. Користувач повинен вводити текст, який виводиться у richtextbox'і. Тексти для вводу можуть бути як на українській, так і на англійській мовах. При введенні коректного символу він змінює свій колір на молочний, що дає змогу краще зрозуміти, який наступний символ потрібно вводити. Також задній фон

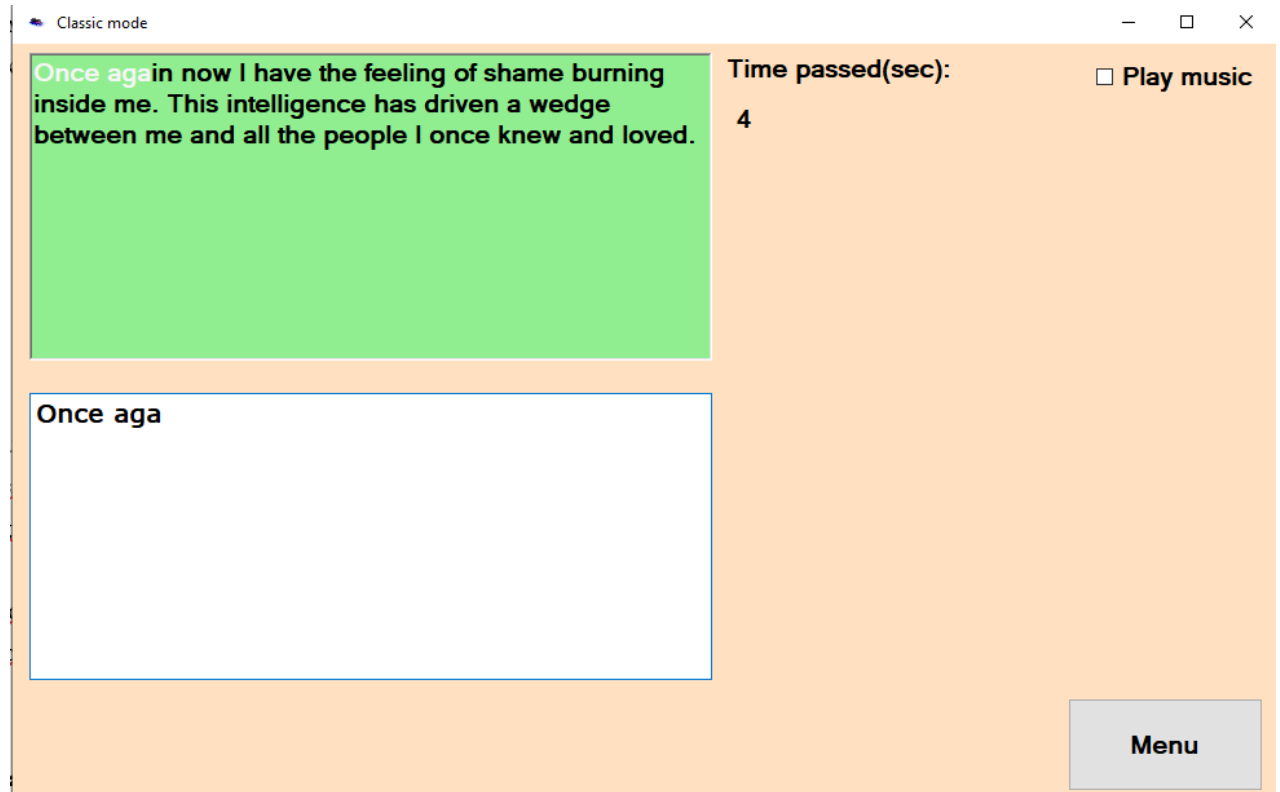


Рис 2.3. Введення коректного символу

При введенні некоректного символу він не буде змінювати свій колір, що дає зрозуміти що символ введено не той, що потрібно. Також у такому випадку задній фон текстбоксу буде змінюватися на червоний. Нагадаємо, що під час гри йде підрахунок витраченого часу(рис 2.4)

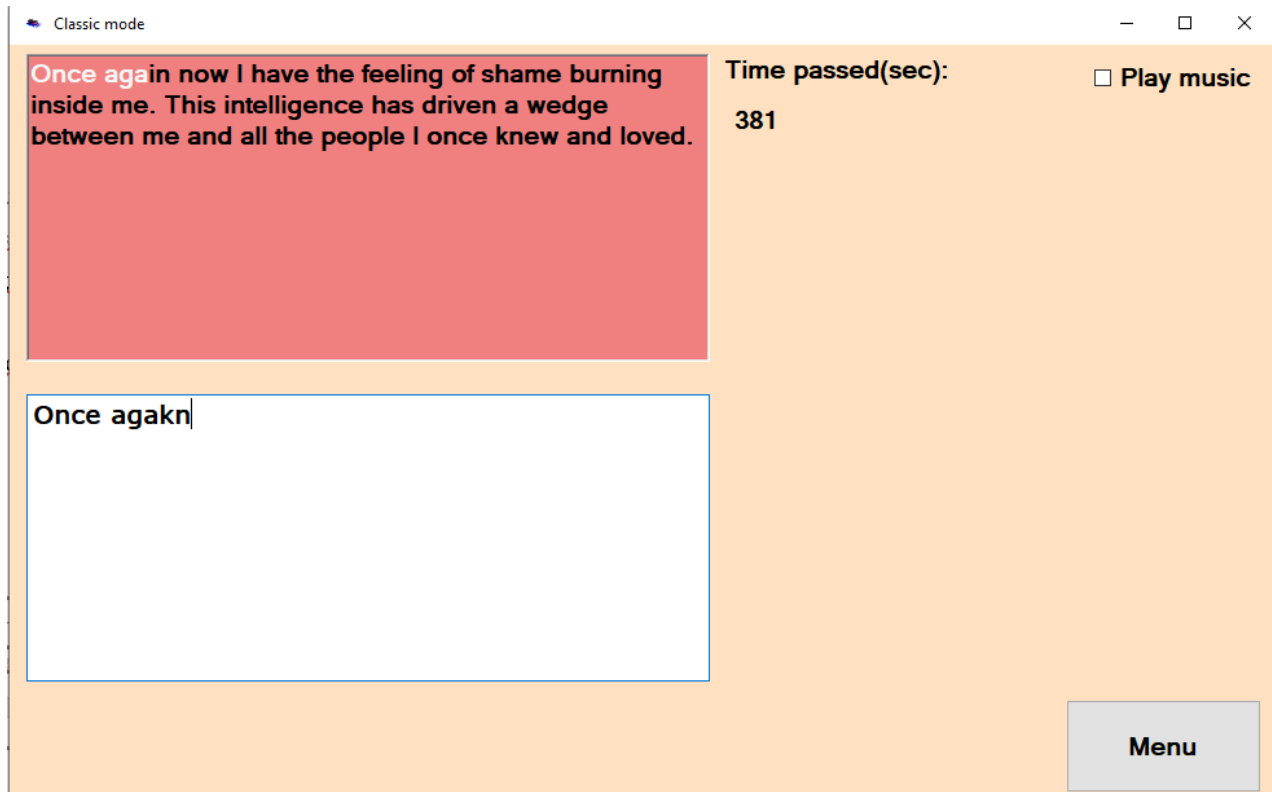


Рис.2.4. Введення некоректного символу

Щодо самого алгоритму, його можна побачити на блок схемі 2.5

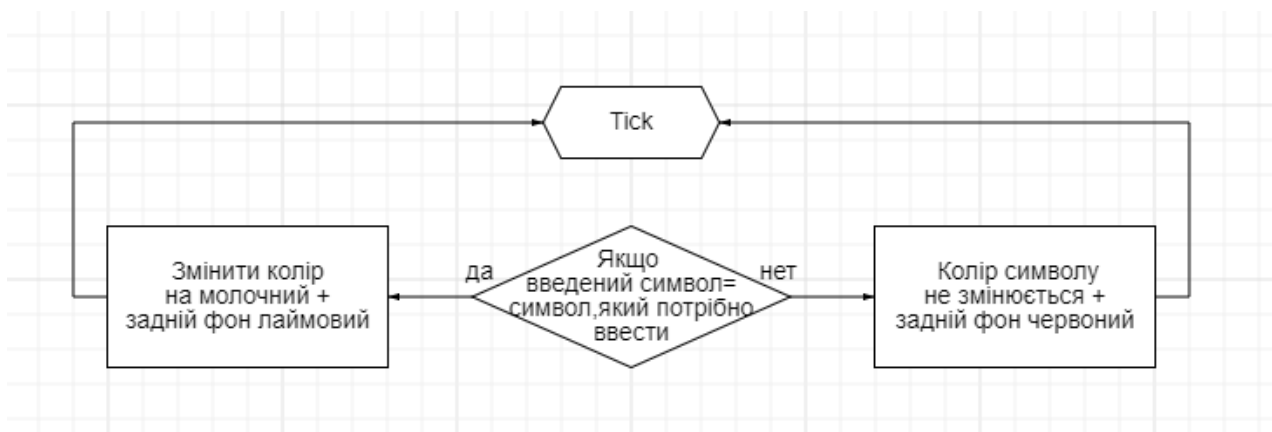


Рис 2.5. Блок-схема алгоритму перевірки вводу

Лістинг алгоритму:

```

void ChangeClr1(RichTextBox TextPanel, Input input)
{
    input.Text = textBox1.Text;
    int lng = input.Text.Length;
    string sub = textToPrnt.TextTPrint.Substring(0, lng);
    string text2 = input.Text;
  
```

		Баковецький М.О.			«Житомирська політехніка».21.122.04.000 – ПЗ	Арк.
		Марчук Г. В.				17
Змн.	Арк.	№ докум.	Підпис	Дата		

```

if (sub == input.Text)
{

    if (lng1 < lng)
    {
        colfl = true;
    }

    if (input.Text.Length > 0 && colfl == true)
    {
        lng1 = lng;
        colfl = false;
        TextPanel.Select(vid,1);
        TextPanel.SelectionColor = Color.WhiteSmoke;
        vid++;
    }
    TextPanel.BackColor = Color.LightGreen
}

else if (sub != input.Text)
{
    TextPanel.BackColor = Color.LightCoral;

    if (text1 != text2 && length > lng)
    {
        misc++;
    }
}
}

```

Також не менш важливою частиною є алгоритм запису рекордів у файл. Рекорди створені для обох режимів. При закінченні гри, тобто коли користувач уввів речення до кінця, рахується швидкість друку, та відсоток точності. Далі ці значення потрапляють у метод CheckOnRecord. У методі створюється список кожного рекорду, який включає в себе wrm, нікнейм та дату поставлення рекорду. В циклі перевіряється кожен рекорд, і якщо дані, які ми отримали тільки що, більші за усі інші рекорди, то усі попередні рекорди зміщуються на одну позицію униз, а новий рекорд стає на верхнє місце. На блок-схемі 2.6 можна побачити алгоритм запису рекордів у файл.

		Баковецький М.О.			«Житомирська політехніка».21.122.04.000 – ПЗ	Арк.
		Марчук Г. В.				
Змн.	Арк.	№ докум.	Підпис	Дата		18

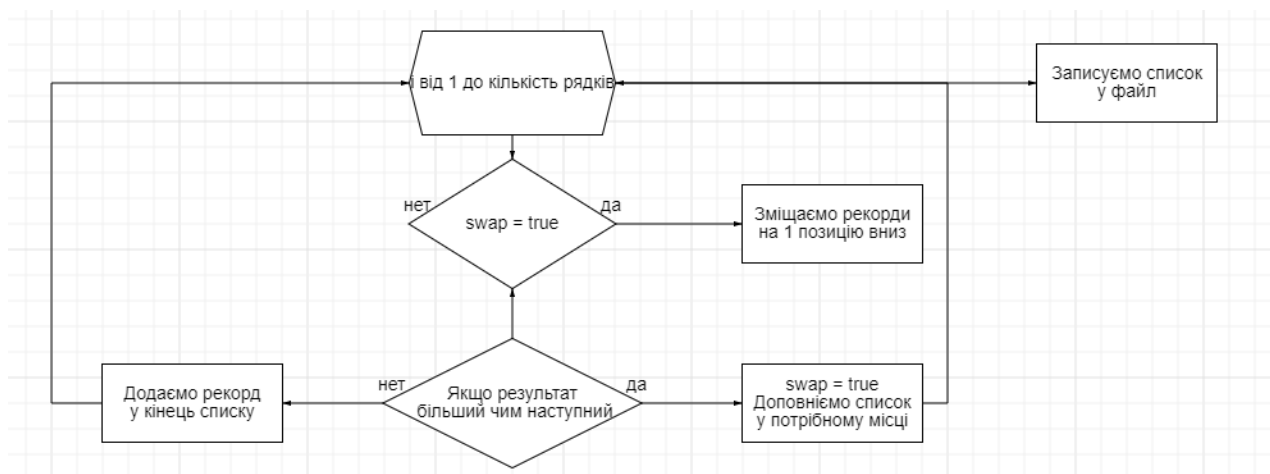


Рис 2.6. Алгоритм запису рекордів

Лістинг алгоритму:

```

public void CheckOnRecord(double wpm)
{
    List<string> list = new List<string>();
    bool swap = false;
    int number;
    Result result = new Result((double)wpm, DateTime.Now);
    lines.CopyTo(linesToWrite, 0);
    for (int i = 0; i < linesToWrite.Length; i++)
    {
        number = 0;
        if (!string.IsNullOrEmpty(linesToWrite[i]))
        {
            string[] words = linesToWrite[i].Split(' ');
            number = int.Parse(words[0]);
        }
        if (swap)
        {
            list.Add(linesToWrite[i - 1]);
        }
        if (result.wpmRes > number && !swap)
        {
            swap = true;
            list.Add($"{(int)result.wpmRes} wpm, Player:{name.Text}
{result.date.ToShortDateString()}");
        }
        else if (!swap)
        {

```

```

        list.Add(linesToWrite[i]);
    }
}
File.WriteAllLines("records.txt", list);
}

```

Вигляд таблиці рекордів можна побачити на рисунку 2.7

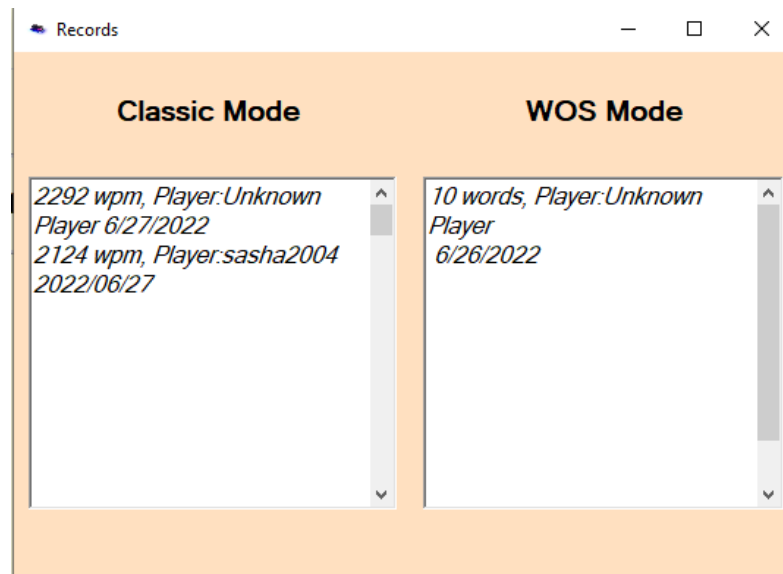


Рис. 2.7. Рекорди

2.3 Розробка програмного забезпечення

Після завершення проектування програмного забезпечення, відразу було розпочато розробку програмного забезпечення. Спочатку було створену форму головного меню, звідки і має відбуватися перехід на усі інші форми. Додано кнопки для режимів(Classic,WOS), кнопка для таблиці рекордів та виходу із програми. Далі було реалізовано локалізацію із вибором трьох мов: української, англійської та японської.(рис 2.8) Після завершення роботи із головним меню була розпочата робота із головним режимом, реалізовано роботу із таймерами, введення тексту та перевірку на його коректність. Після цього реалізовано форму другого режиму, запис рекордів у окрему форму.

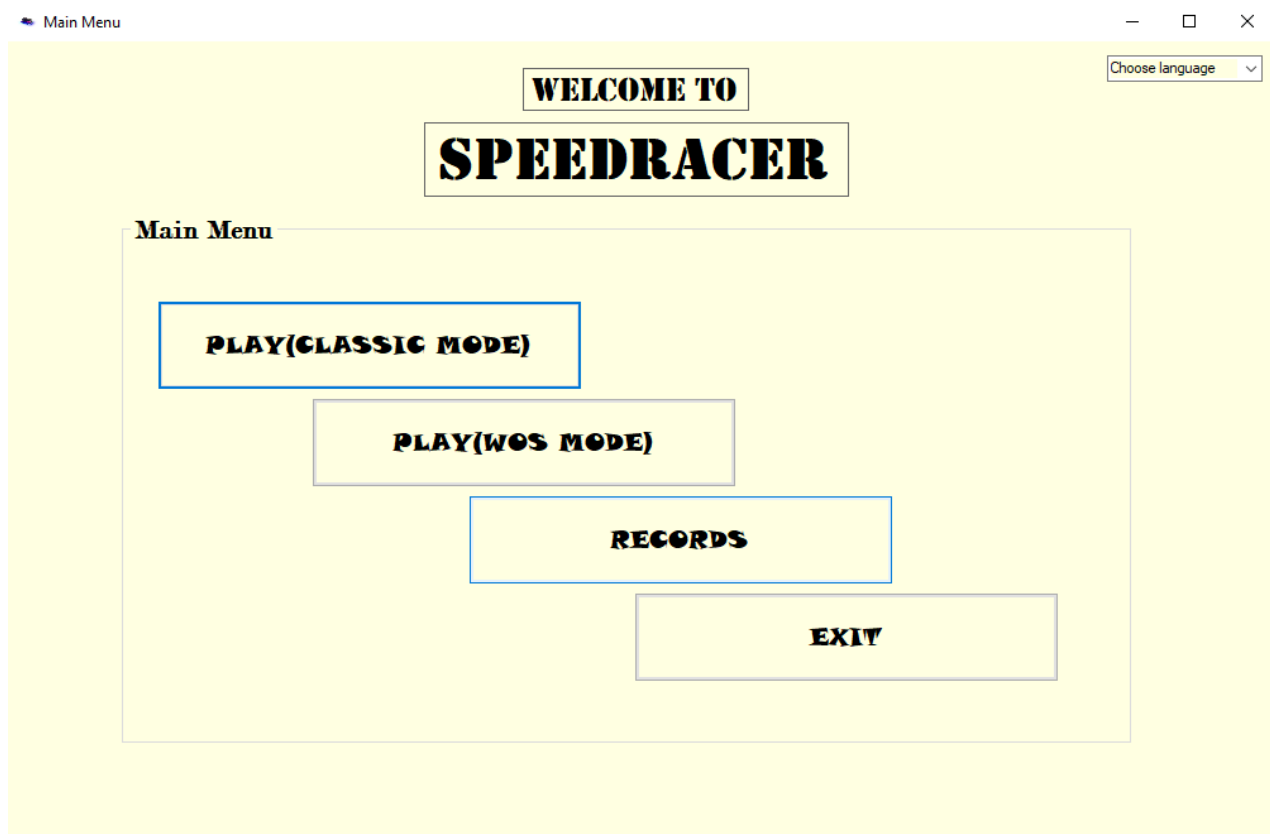


Рис 2.8. Головне меню

Висновки до другого розділу

У даному розділі було розроблено загальний алгоритм роботи програми, реалізовані та описані основні функціональні алгоритми. Було розроблено основне програмне забезпечення, наведено допоміжні блок-схеми та рисунки, наведено лістинг основних алгоритмів.

		Баковецький М.О.			«Житомирська політехніка».21.122.04.000 – ПЗ	Арк.
		Марчук Г. В.				21
Змн.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 3. ОПИС РОБОТИ З ПРОГРАМНИМ ДОДАТКОМ ТА ЙОГО ТЕСТУВАННЯ

3.1 Опис роботи з програмним додатком

Після запуску програмного додатку ми потрапляємо у головне меню, де користувач має можливість обрати мову інтерфейсу програми. На рисунку 3.1 можна побачити інтерфейсу на українській мові, на рисунку 3.2 – на японській. Приклад на англійській мові був продемонстрований на рисунку 2.8.

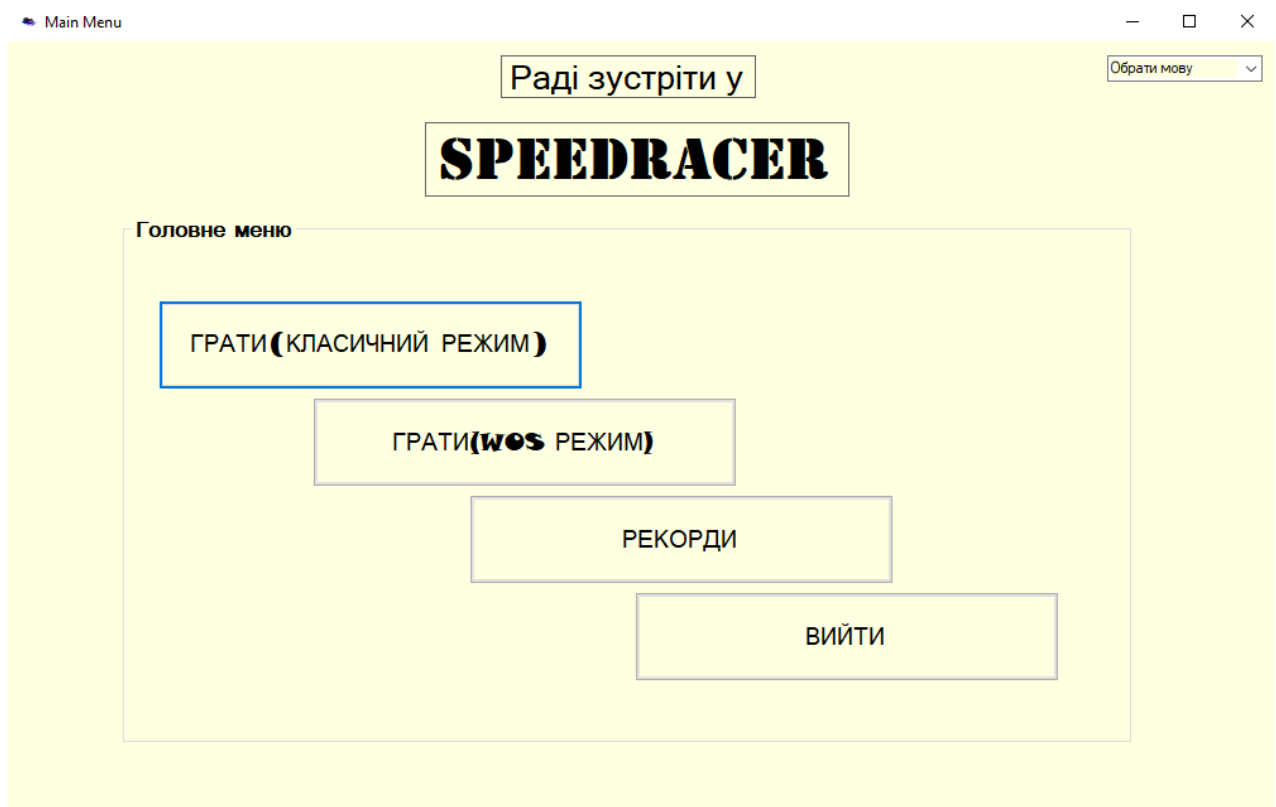


Рис 3.1. Інтерфейс на українській мові

		Баковецький М.О.			«Житомирська політехніка».21.122.04.000 – ПЗ	Арк.
		Марчук Г. В.				22
Змн.	Арк.	№ докум.	Підпис	Дата		



Рис 3.2 Інтерфейс на японській мові

Далі перейдемо до класичного режиму, натиснувши на кнопку Play(Classic mode). При переході ми відразу бачимо два текстбокса: один для тексту, який потрібно ввести, другий для тексту який вводиться. Також на формі ще є текстбокс для введення ім'я користувача, кнопка для початку гри, кнопка виходу у головне меню та чекбокс для програвання музики.(рис 3.3). Потрібно зауважити, що по замовчування ми не можемо вводити текст в обидва головних текстбокса до початку гри. Після початку у головному текстбоксі readonly набуває значення false, тобто дозволяється вводити текст. .

		Баковецький М.О.			«Житомирська політехніка».21.122.04.000 – ПЗ	Арк.
		Марчук Г. В.				23
Змн.	Арк.	№ докум.	Підпис	Дата		

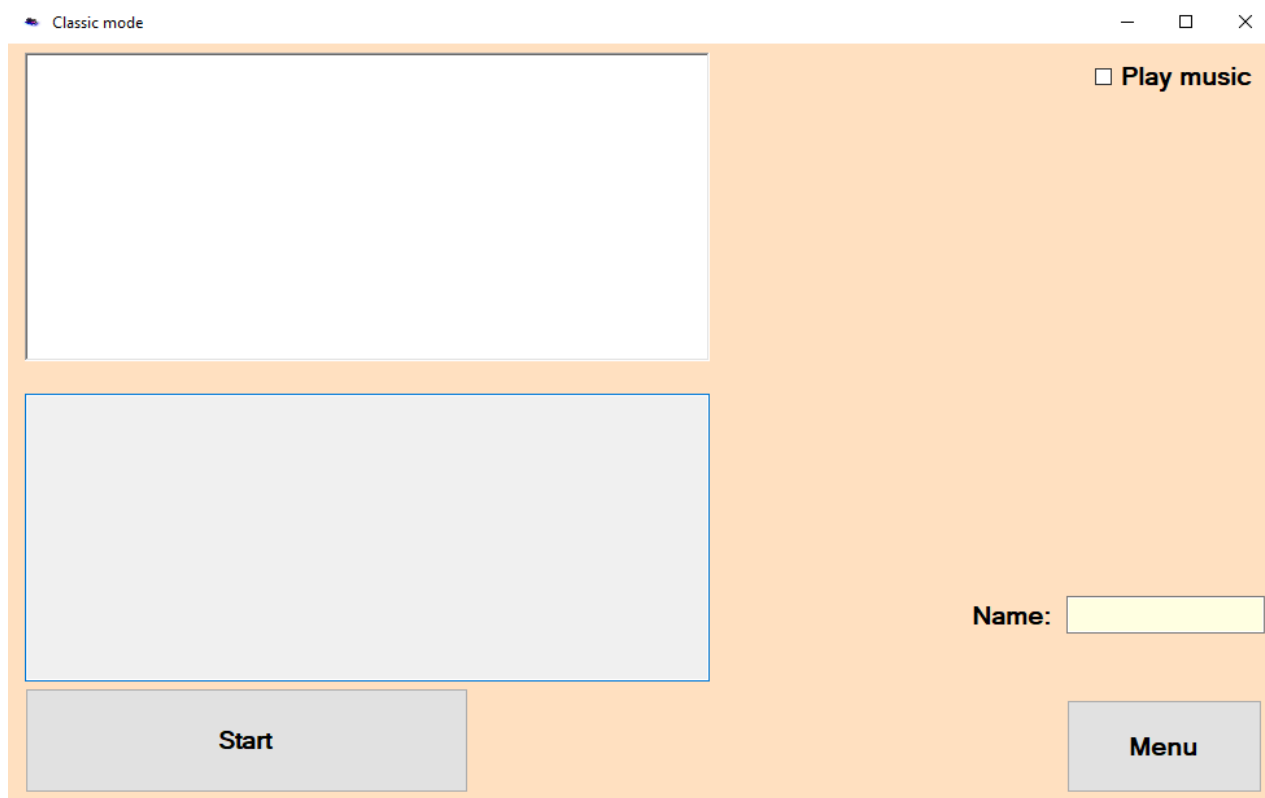
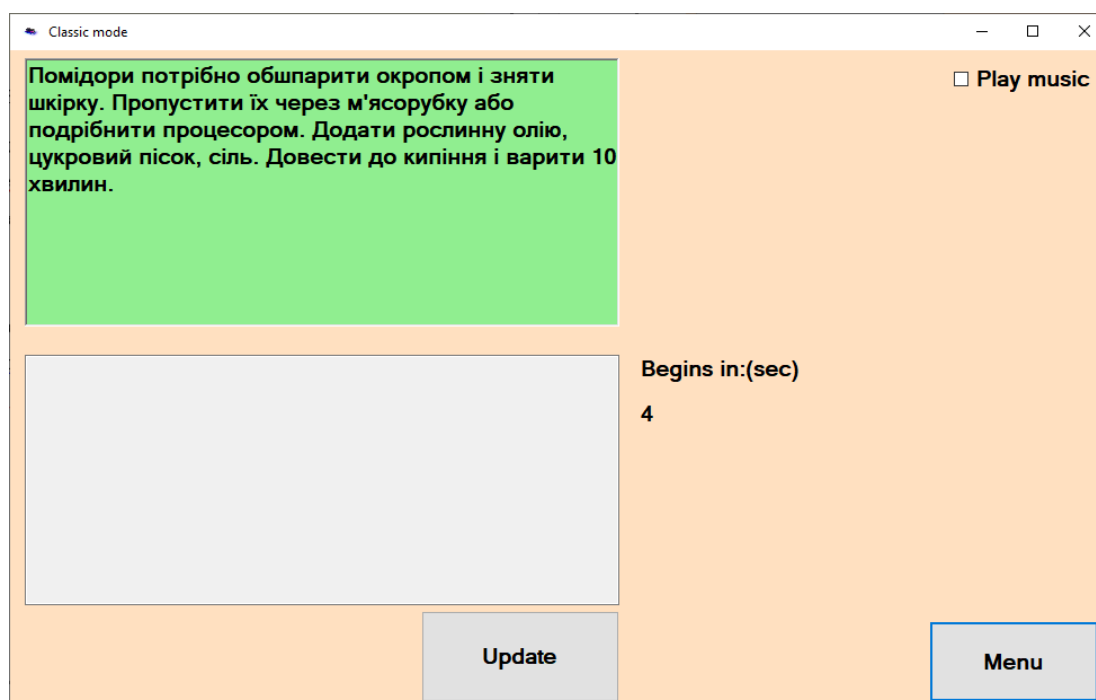


Рис 3.3. Інтерфейс класичного режиму

Після натиснення кнопки Start запускається таймер відліку, який за замовчування дорівнює п'ять секунд. Також кнопка старт зникає і з'являється кнопка update, яка дозволяє змінити текст для друку, якщо він надто складний або просто не подобається. Зникає також і текстбокс для вводу ім'я.(рис 3.4).



		Баковецький М.О.			«Житомирська політехніка».21.122.04.000 – ПЗ	Арк.
		Марчук Г. В.				24
Змн.	Арк.	№ докум.	Підпис	Дата		

Рис 3.4. Перед початком гри

Після того, як таймер відліку дійде до нуля, запуститься другий таймер, який буде рахувати час, проведений за грою. Тоді readonly у головного текстбокса набуде значення false і користувач зможе почати гру. Також кнопка update стане невидимою.(рис 3.5)

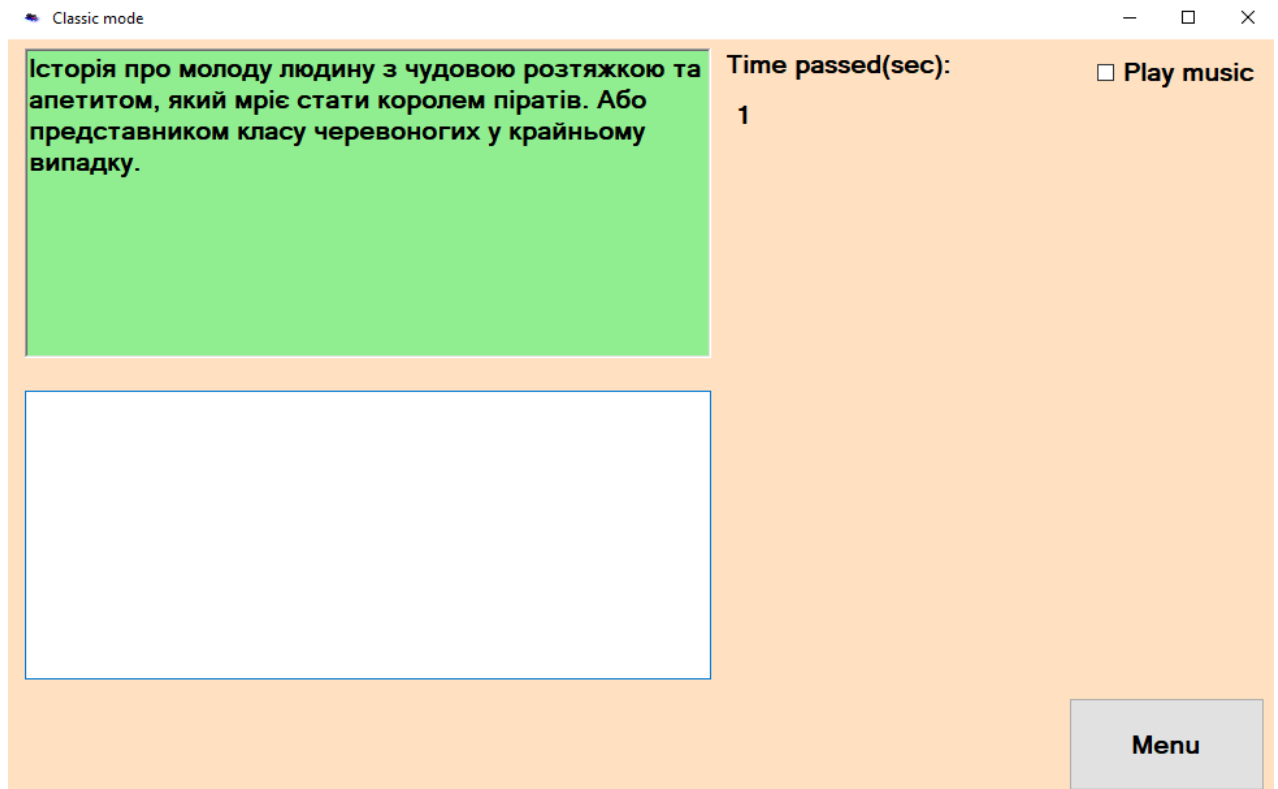


Рис. 3.5. Початок гри

Після закінчення введення тексту на екрані з'явиться повідомлення про закінчення гри та результатом, тобто із швидкістю вводу(слів за хвилину, wrm) та точністю вводу(accuracy,%).(рис 3.6)

		Баковецький М.О.			«Житомирська політехніка».21.122.04.000 – ПЗ	Арк.
		Марчук Г. В.				
Змн.	Арк.	№ докум.	Підпис	Дата		25

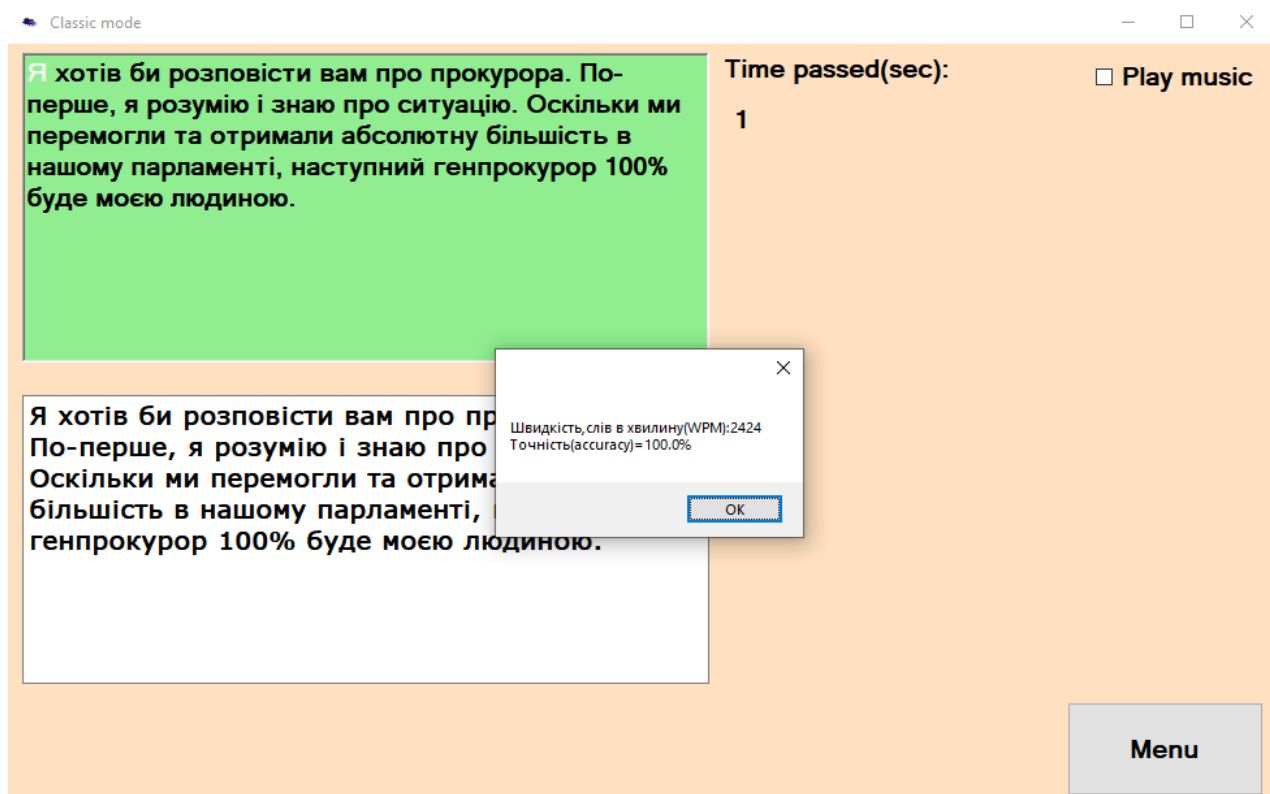
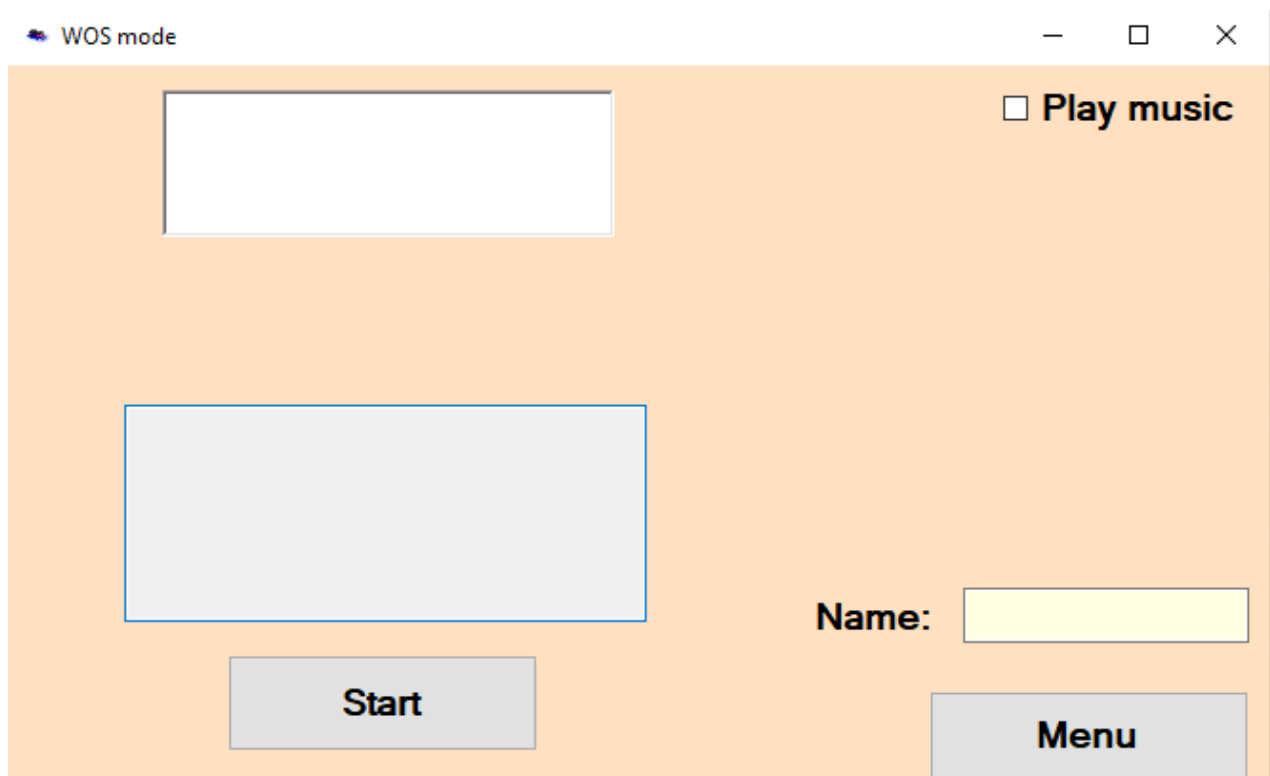


Рис 3.6. Виведення результату

Після натиснення на ОК форма перезапуститься і користувач зможе знову почати гру, або ж вийти до меню. Далі продемонстровано вигляд режиму WOS.(рис. 3.7)



		Баковецький М.О.			«Житомирська політехніка».21.122.04.000 – ПЗ	Арк.
		Марчук Г. В.				26
Змн.	Арк.	№ докум.	Підпис	Дата		

Рис. 3.7. Режим WOS

Елементи форми майже такі ж самі, як і елементи форми класичного режиму, але поменше і відсутня кнопка update. При натисканні кнопки Start починається відлік, і зникають деякі елементи.(рис. 3.8)

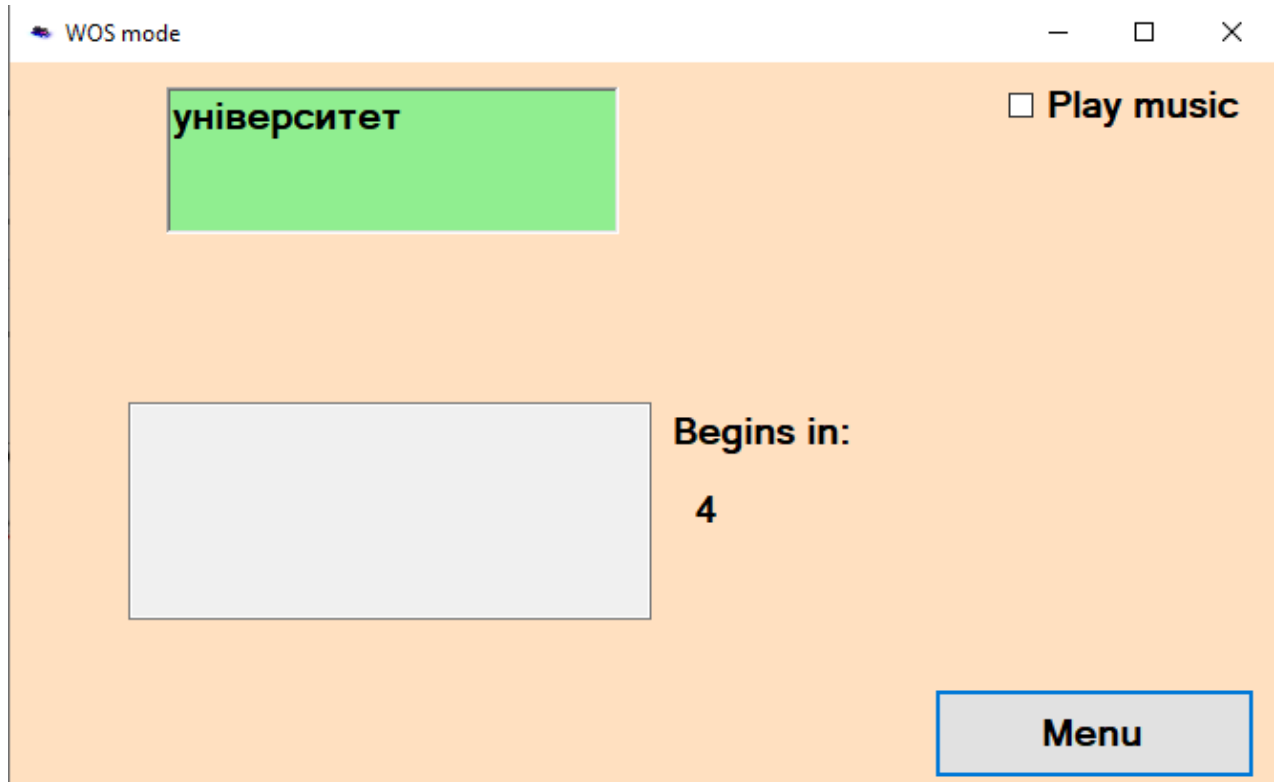


Рис 3.8. Відлік до початку

При закінченні відліку починається гра. Користувач повинен вводити слова, які виводяться у текстові. Після введення слова з'являється нове слово, яке потрібно також ввести. Так триває до того часу, поки основний таймер не дійде до нуля.(рис. 3.9)

		Баковецький М.О.			«Житомирська політехніка».21.122.04.000 – ПЗ	Арк.
		Марчук Г. В.				27
Змн.	Арк.	№ докум.	Підпис	Дата		

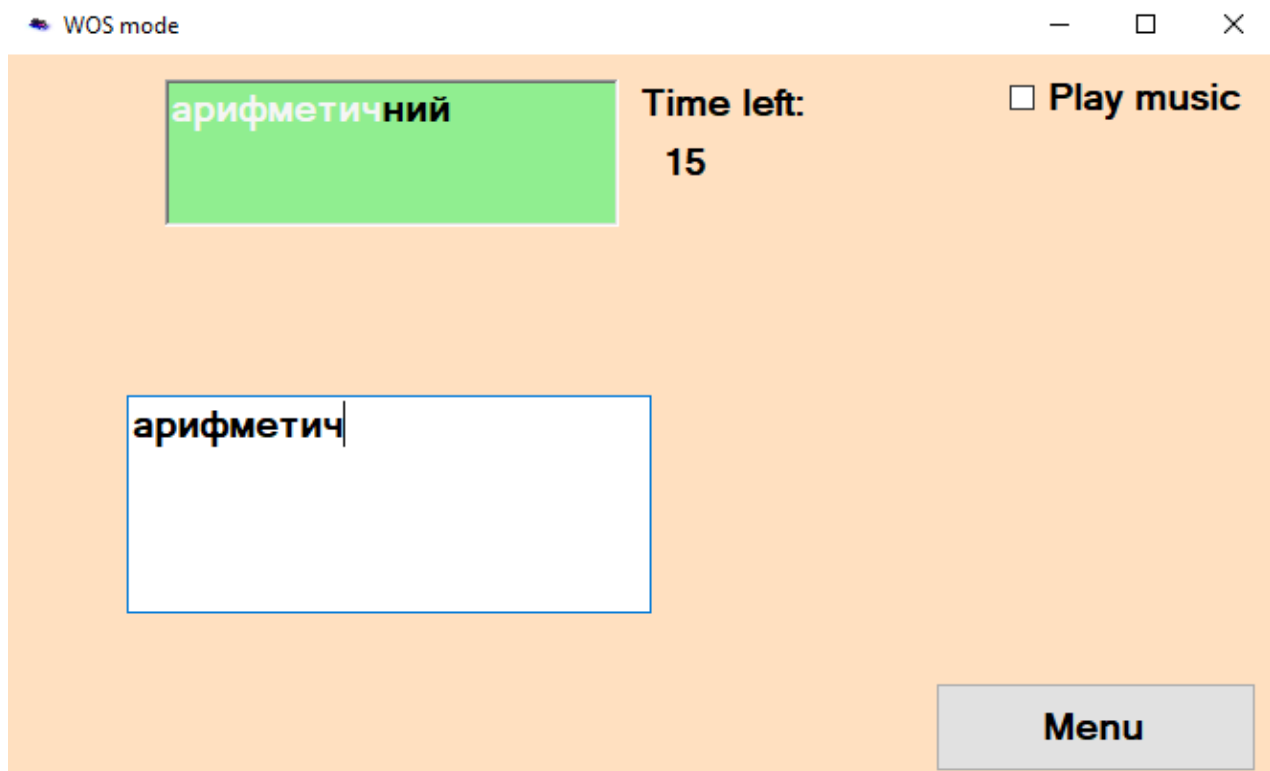
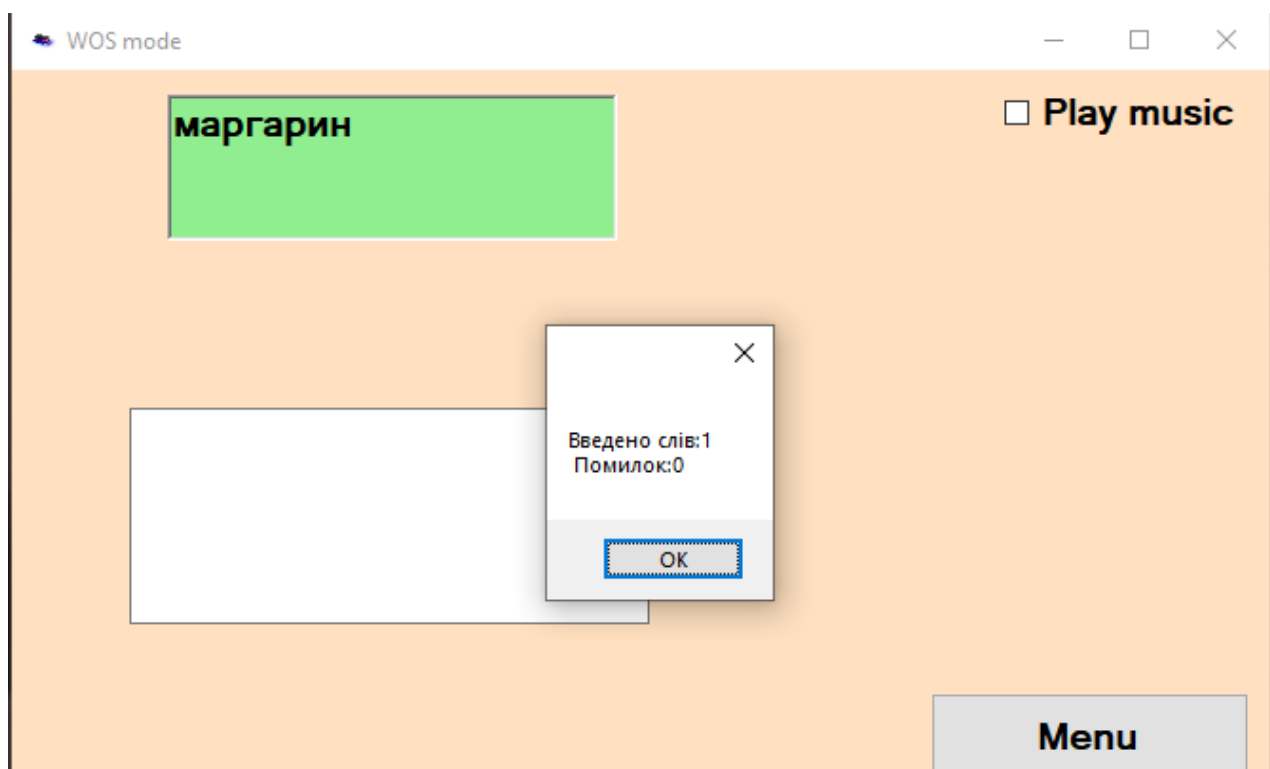


Рис. 3.9 Процес гри

Після закінчення основного таймеру на екрані з'явиться повідомлення про кількість введених слів та кількість помилок, також результат запишеться у рекорди.(рис. 3.10)



		Баковецький М.О.			«Житомирська політехніка».21.122.04.000 – ПЗ	Арк.
		Марчук Г. В.				28
Змн.	Арк.	№ докум.	Підпис	Дата		

Рис. 3.10 Виведення результатів

Далі продемонстровано вигляд таблиці рекордів.(рис. 3.11)

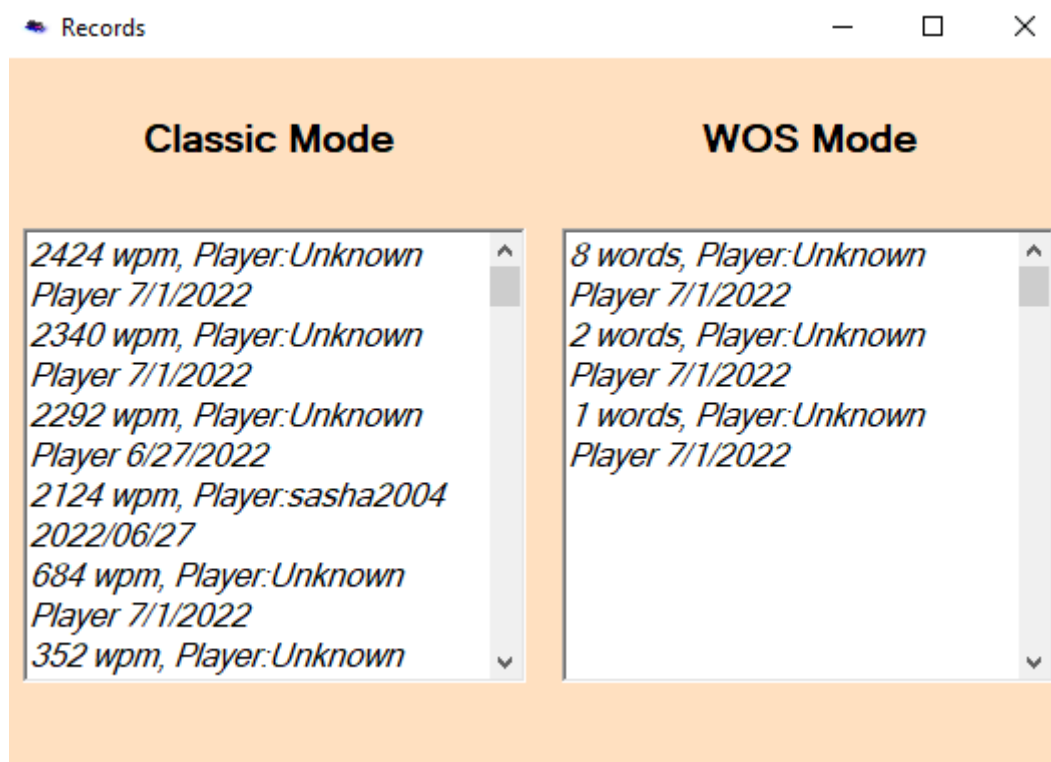


Рис. 3.11. Таблиця рекордів

3.2 Тестування роботи програмного забезпечення

Під час тестування програмного додатку було виявлено дві помилки. Помилки пов'язані із кількістю символів, які вводяться. Якщо кількість цих символів перебільшує кількість символів, які потрібно ввести, то програма видасть помилку. Було виправлено помилку на повідомлення, яке показує, що символів забагато, і у випадку класичного режиму очищує форму для вводу та обнуляє таймер, а у випадку із WOS режимом, просто очищує форму для вводу.(рис. 3.12, рис. 3.13)

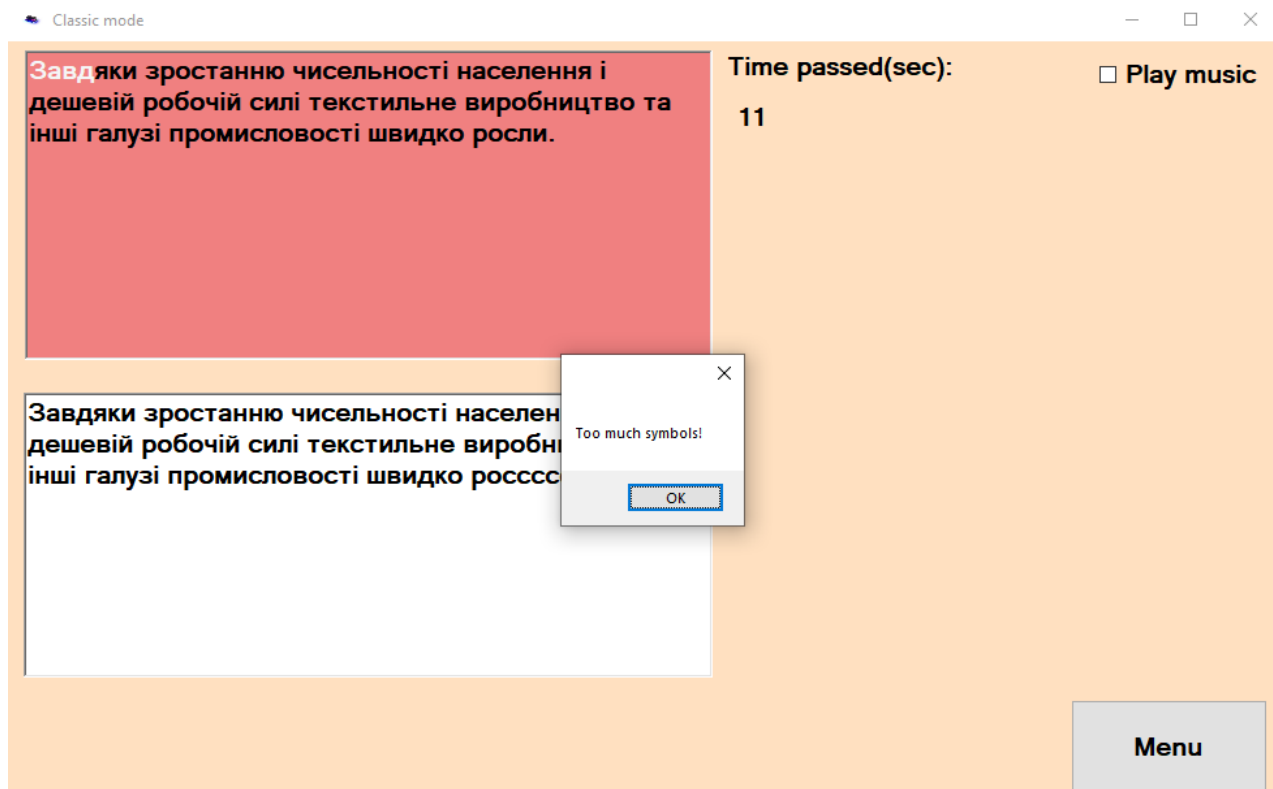


Рис 3.12. Помилка у класичному режимі

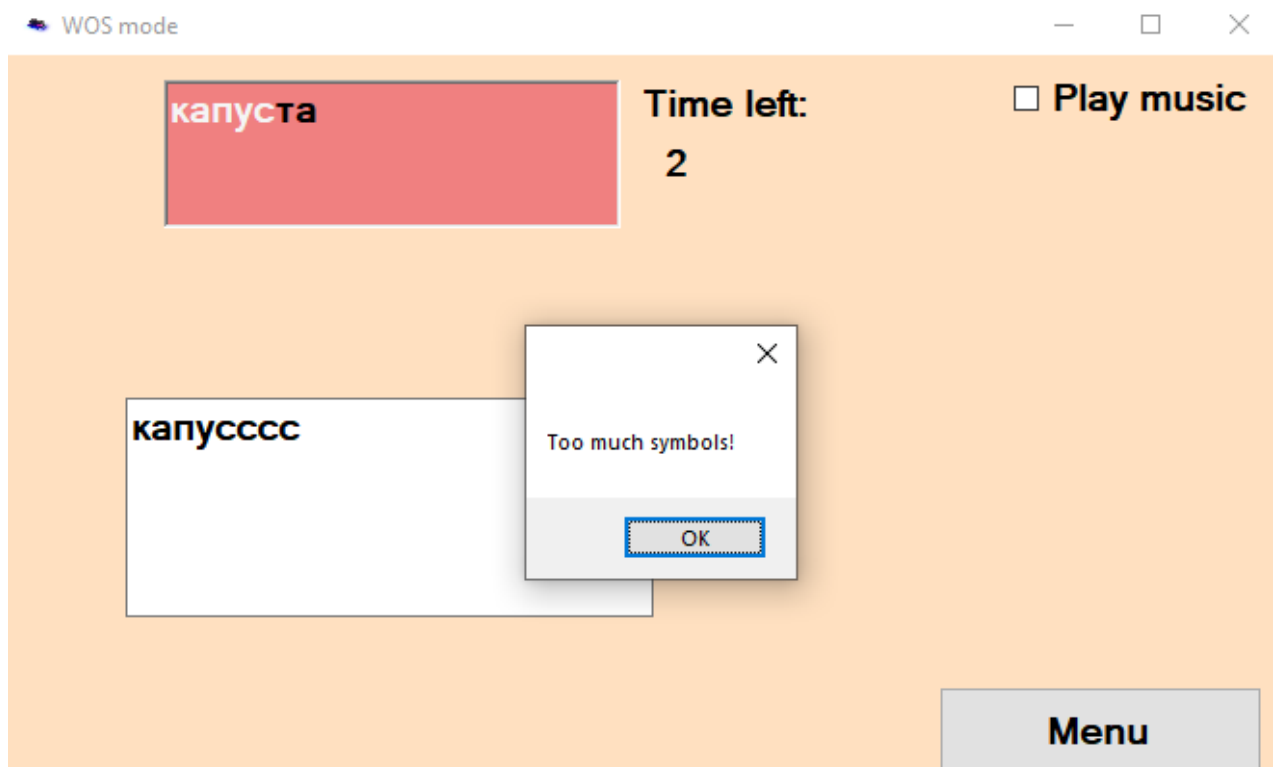


Рис 3.13 Помилка у WOS режимі

		Баковецький М.О.			«Житомирська політехніка».21.122.04.000 – ПЗ	Арк.
		Марчук Г. В.				30
Змн.	Арк.	№ докум.	Підпис	Дата		

Також було виявлено, що властивість readonly у одній із таблиці рекордів набувала значення false, що означало, що користувач може редагувати рекорди. (рис 3.14)

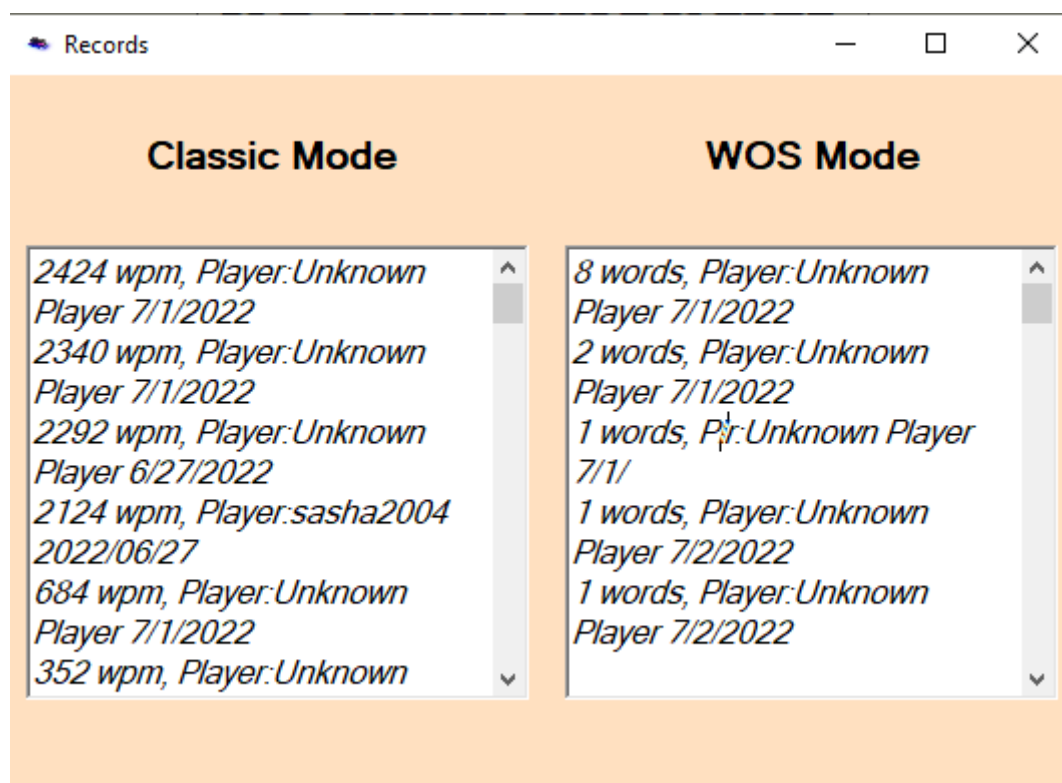


Рис 3.14 Можливість редагування рекорду у таблиці WOS

Висновки до третього розділу

Отже, після реалізації усіх основних алгоритмів програми, та після оформлення зовнішнього вигляду програми, було проведено тестування, під час якого вдалося виявити три помилки. Одну із них було відразу виправлено, а для інших було створено повідомлення, які з'являлись на екрані при невірних діях користувача.

ВИСНОВКИ

Отже, під час написання курсової роботи було отримано корисні знання про тренажери для друку тексту та проаналізований їх функціонал, знання щодо теоретичної частини, вдосконалено навички роботи із класами, файлами, визначено основні цілі та способи реалізації продукту.

Було написано тренажер для друку тексту, зрозумілу у використанні та цікаву програму, яка дозволяє покращити свої здібності щодо швидкості набору тексту на клавіатурі.

Даний програмний продукт було реалізовано за допомогою інтегрованого середовища розробки програмного забезпечення Microsoft Visual Studio 2022, в додатку Windows Forms(WF). Була обрана платформа .Net Framework і мова програмування C#. При реалізації був використаний об'єктно – орієнтований підхід.

		Баковецький М.О.			«Житомирська політехніка».21.122.04.000 – ПЗ	Арк.
		Марчук Г. В.				32
Змн.	Арк.	№ докум.	Підпис	Дата		

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ(мінімум 10)

- 1) Объектно-ориентированное программирование. Практика Создание проекта библиотеки классов
<https://metanit.com/sharp/tutorial/3.29.php>
- 2) Классы, структуры и пространства имен Классы и объекты
<https://metanit.com/sharp/tutorial/3.1.php>
- 3) Работа с файлами. Классы File и FileInfo
<https://metanit.com/sharp/tutorial/5.3.php>
- 4) Отношения классов — от UML к коду Java * UML Design *
<https://habrahabr.ru/post/150041/>
- 5) Сборка мусора и деструкторы
https://professorweb.ru/my/csharp/charp_theory/level5/5_6.php
- 6) Модификаторы доступа в аксессорах
https://professorweb.ru/my/csharp/charp_theory/level5/5_15.php
- 7) File Класс
<https://docs.microsoft.com/ru-ru/dotnet/api/system.io.file?view=net-6.0>
- 8) Построение диаграммы классов
https://flexberry.github.io/ru/gpg_class-diagram.html
- 9) Фленов М. Библия C# / Михаил Фленов. – Питер: БХВ-Петербург, 2011. – 541 с. – (2-е изд).
- 10) Warren G. Подключение к данным в базе данных Access (Windows Forms) [Электронный ресурс] / Genevieve Warren. – 2019.
- 11) Рихтер Джеффри CLR via C#. Программирование на платформе Microsoft.NET Framework 4.5 на языке C# / Джеффри Рихтер. – Питер, 2013. – 896 с. – (4-е изд).
- 12) Шилдт Герберт C# 4.0: полное руководство. : Пер. с англ. — М. : ООО "И.Д. Вильямс", 2011. — 1056 с.: ил.

		Баковецкий М.О.			«Житомирська політехніка».21.122.04.000 – ПЗ	Арк.
		Марчук Г. В.				
Змн.	Арк.	№ докум.	Підпис	Дата		33

		Баковецький М.О.			«Житомирська політехніка».21.122.04.000 – ПЗ	Арк.
		Марчук Г. В.				
Змн.	Арк.	№ докум.	Підпис	Дата		34

ДОДАТКИ

Лістинг Form1:

```

using System;
using System.Drawing;
using System.Windows.Forms;
using System.Globalization;
using System.IO;
using System.Collections.Generic;

namespace Speed_Typing_App
{
    public partial class Form1 : Form
    {
        //основні класи:
        public class Result
        {
            public readonly double wpmRes;

            public readonly DateTime date;

            public Result(double wpm, DateTime date)
            {
                wpmRes = wpm;
                this.date = date;
            }
        }
        public class TextToPrint
        {
            public string TextTPrint { get; set; }
            public int SymbCount = 0;
        }
        public class Input
        {
            public string Text;
            public double acc;
            public double wordcount = 0;
            public double time;
            public double Time
            {
                get { return time; }
                set { time = value; }
            }
        }
        public class Music
        {
            public static WMPLib.WindowsMediaPlayer WMP = new
WMPLib.WindowsMediaPlayer();
            Timer tmr = new Timer();
            public void play_music_1()
            {
                tmr.Interval = 10;
                tmr.Stop();
                WMP.URL = @"2.mp3";
                WMP.settings.volume = 20;
                WMP.controls.play();
                tmr.Tick += new EventHandler(tmr_Tick);
                WMP.PlayStateChange += new
WMPLib._WMPOCXEvents_PlayStateChangeEvent(wplayer_PlayStateChange);
            }
        }
    }
}

```

```

        void tmr_Tick(object sender, EventArgs e)
        {
            WMP.controls.stop();
            play_music_1();
        }
        void wplayer_PlayStateChange(int NewState)
        {
            if (NewState == (int)WMPLib.WMPPlayState.wmppsMediaEnded)
            {
                tmr.Start();
            }
        }
        public void StopMusic()
        {
            WMP.controls.stop();
        }
    }
    //загальні змінні:
    string[] lines = File.ReadAllLines("records.txt");
    string[] linesToWrite = new string[100];
    int ticks = 6;
    int ticks1 = 0, vid = 0;
    bool flag = true;
    int misc = 0, lng1=0;
    bool lang = false, musfl = false, colfl = true, fl1=true;
    string text1 = "";
    int length = 0;
    TextToPrint textToPrnt = new TextToPrint();
    Input name = new Input();
    Music music = new Music();
    public Form1()
    {
        System.Threading.Thread.CurrentThread.CurrentUICulture
        =
        CultureInfo.GetCultureInfo(Properties.Settings.Default.Language);
        System.Threading.Thread.CurrentThread.CurrentCulture
        =
        CultureInfo.GetCultureInfo(Properties.Settings.Default.Language);
        InitializeComponent();
    }

    private void Form1_Load(object sender, EventArgs e)
    {
        TextPanel.ReadOnly = true;
        textBox1.ReadOnly = true;
        label2.Visible = false;
        label3.Visible = false;
        this.FormBorderStyle = FormBorderStyle.Fixed3D;
        button2.Visible = false;
    }

    //метод для тика першого таймеру(таймер відліку)
    private void timer1_Tick(object sender, EventArgs e)
    {
        if (checkBox1.Checked == true && musfl == false)
        {
            music.play_music_1();
            musfl = true;
        }
        else if (checkBox1.Checked == false)
        {
            music.StopMusic();
            musfl = false;
        }
    }

```

```

    }
    Input input = new Input();
    ticks--;
    timer1.Interval = 1000;
    if (ticks < 0)
    {
        timer1.Interval = 1;
        label1.Visible = false;
        textBox1.ReadOnly = false;
        label2.Visible = true;
        label3.Visible = true;
        label4.Visible = false;
        timer2.Start();
        button2.Visible = false;
    }

    label1.Text = ticks.ToString();
    ChangeClr1(TextPanel, input);
    CheckOnEnd(input);
}
//метод для тіка другого таймера(таймер під час гри)
private void timer2_Tick(object sender, EventArgs e)
{
    ticks1++;
    timer2.Interval = 1000;
    label2.Text = ticks1.ToString();
}
//початок гри
private void button1_Click(object sender, EventArgs e)
{
    button1.Visible = false;
    button2.Visible = true;
    nameBox.Visible = false;
    nameLabel.Visible = false;
    label4.Visible = true;
    timer1.Start();
    GenerateSent();
    TextPanel.Text = textToPrnt.TextTPrint;
    if (nameBox.Text == "") name.Text = "Unknown Player";
    else name.Text = nameBox.Text;
}
//метод для перевірки правильності вводу
void ChangeClr1(RichTextBox TextPanel, Input input)
{
    input.Text = textBox1.Text;
    int lng = input.Text.Length;
    string sub = "";
    try
    {
        {
            fl1 = true;
            sub = textToPrnt.TextTPrint.Substring(0, lng);
        }
    }
    catch
    {
        timer1.Enabled = false;
        timer2.Enabled = false;
        MessageBox.Show("Too much symbols!");
        TextPanel.Select(0, textToPrnt.TextTPrint.Length);
        TextPanel.SelectionColor = Color.Black;
        textBox1.Text = null;
        ticks1 = 0;
    }
}

```

```

    }
    if (fl1)
    {
        timer1.Enabled = true;
        timer2.Enabled = true;
    }
    string text2 = input.Text;
    if (sub == input.Text)
    {
        foreach (char c in sub)
        {
            input.wordcount++;

        }
        if (lng1 < lng)
        {
            colfl = true;
        }
        if (input.Text.Length > 0 && colfl == true)
        {
            lng1 = lng;
            colfl = false;
            TextPanel.Select(vid, 1);
            TextPanel.SelectionColor = Color.WhiteSmoke;
            vid++;
        }

        TextPanel.BackColor = Color.LightGreen;

    }

    else if (sub != input.Text)
    {
        TextPanel.BackColor = Color.LightCoral;

        if (text1 != text2 && length > lng)
        {
            misc++;
        }

    }
    text1 = input.Text;
    length = lng;
}
//метод для перевірки кінця гри
void CheckOnEnd(Input input)
{
    input.Text = textBox1.Text;
    if (input.Text.Length == textToPrnt.TextTPrint.Length
        && input.Text == textToPrnt.TextTPrint
        && flag == true)
    {
        flag = false;
        timer1.Stop();
        timer2.Stop();
        input.Time = ticks1;
        Print(input);
    }
}
//метод для для обчислення значень wrm та accuracy та виводу
інформації-
void Print(Input input)
{
    if (misc > 0)

```

```

        misc++;
        double correlem = textToPrnt.TextTPrint.Length - misc;
        input.acc = (correlem / textToPrnt.TextTPrint.Length) * 100.0;
        double wpm = (((input.wordcount / 5) / input.Time) * 60);
        CheckOnRecord(wpm);
        MessageBox.Show($"Швидкість, слів в
хвилину (WPM) : {wpm:f0} \nТочність (accuracy)={input.acc:f1}%");
        Form1 form = new Form1();
        this.Hide();
        form.Show();
    }
    //метод для обробки результату та запису у файл
    public void CheckOnRecord(double wpm)
    {
        List<string> list = new List<string>();
        bool swap = false;
        int number;
        Result result = new Result((double)wpm, DateTime.Now);
        lines.CopyTo(linesToWrite, 0);
        for (int i = 0; i < linesToWrite.Length; i++)
        {
            number = 0;
            if (!string.IsNullOrEmpty(linesToWrite[i]))
            {
                string[] words = linesToWrite[i].Split(' ');
                number = int.Parse(words[0]);
            }
            if (swap)
            {
                list.Add(linesToWrite[i - 1]);
            }
            if (result.wpmRes > number && !swap)
            {
                swap = true;
                list.Add($"{(int)result.wpmRes} wpm, Player:{name.Text}
{result.date.ToShortDateString()}");
            }
            else if (!swap)
            {
                list.Add(linesToWrite[i]);
            }
        }
        File.WriteAllLines("records.txt", list);
    }
    //зміна тексту для вводу
    private void button2_Click(object sender, EventArgs e)
    {
        timer1.Stop();
        timer1.Start();
        GenerateSent();
        TextPanel.Text = textToPrnt.TextTPrint;
    }
    //метод для генерації- тексту для вводу
    void GenerateSent()
    {
        string[] readText = File.ReadAllLines("text.txt");
        Random random = new Random();
        int i = random.Next(readText.Length);
        textToPrnt.TextTPrint = readText[i];
    }
    //в меню
    private void ReturnToMenu_Click(object sender, EventArgs e)
    {
        music.StopMusic();
    }

```



```

        Form3 form3 = new Form3 ();
        this.Hide ();
        form3.Show ();
    }
}

```

Лістинг Form2:

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.IO;
using System.Threading;

namespace Speed_Typing_App
{
    public partial class Form2 : Form
    {
        public Form2 ()
        {
            InitializeComponent ();
        }

        private void Form2_Load(object sender, EventArgs e)
        {

        }
    }
}

```

Лістинг Form3:

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.Globalization;
using System.IO;
using System.Reflection;

namespace Speed_Typing_App
{
    public partial class Form3 : Form
    {
        public Form3 ()
        {
            System.Threading.Thread.CurrentThread.CurrentUICulture
            =
CultureInfo.GetCultureInfo (Properties.Settings.Default.Language) ;
            System.Threading.Thread.CurrentThread.CurrentCulture
            =
CultureInfo.GetCultureInfo (Properties.Settings.Default.Language) ;
            InitializeComponent ();
        }
    }
}

```

```

    }
    //загальні змінні
    bool lang = false;
    Form1 form1 = new Form1();
    Form4 form4 = new Form4();
    //класичний режим
    private void startButton_Click(object sender, EventArgs e)
    {
        form1.Show();
        this.Hide();
    }
    //WOS режим
    private void start2Button_Click(object sender, EventArgs e)
    {
        form4.Show();
        this.Hide();
    }

    //метод для зчитування даних із файлу і запису рекордів у форму
    private void button1_Click(object sender, EventArgs e)
    {
        Form2 form2 = new Form2();
        form2.Show();
        string[] recordsLines = File.ReadAllLines("records.txt");
        string[] wosRecordsLines = File.ReadAllLines("WOSrecords.txt");
        foreach (string line in recordsLines)
        {
            form2.RecordsBox.Text += line + "\n";
        }
        foreach (string line in wosRecordsLines)
        {
            form2.WOSRecords.Text += line + "\n";
        }
        form2.RecordsBox.ReadOnly = true;
        form2.WOSRecords.ReadOnly = true;
        form2.label1.Focus();
    }
    //локалізація
    private void languages_SelectedIndexChanged(object sender, EventArgs e)
    {
        lang = true;
        if (languages.SelectedIndex == 0 && lang)
        {
            System.Threading.Thread.CurrentThread.CurrentUICulture =
CultureInfo.GetCultureInfo("uk-UA");
            System.Threading.Thread.CurrentThread.CurrentCulture =
CultureInfo.GetCultureInfo("uk-UA");
            Properties.Settings.Default.Language = "uk-UA";
            Properties.Settings.Default.Save();
            Application.Restart();
            languages.Text = "Українська";
        }
        else if (languages.SelectedIndex == 1)
        {
            System.Threading.Thread.CurrentThread.CurrentUICulture =
CultureInfo.GetCultureInfo("en-US");
            System.Threading.Thread.CurrentThread.CurrentCulture =
CultureInfo.GetCultureInfo("en-US");
            Properties.Settings.Default.Language = "en-US";
            Properties.Settings.Default.Save();
            Application.Restart();
            languages.Text = "English";
        }
    }

```

```

        else if (languages.SelectedIndex == 2)
        {
            System.Threading.Thread.CurrentThread.CurrentUICulture =
CultureInfo.GetCultureInfo("ja-JP");
            System.Threading.Thread.CurrentThread.CurrentCulture =
CultureInfo.GetCultureInfo("ja-JP");
            Properties.Settings.Default.Language = "ja-JP";
            Properties.Settings.Default.Save();
            Application.Restart();
            languages.Text = "◆-◆本";
        }
    }
    //вихід
    private void button2_Click(object sender, EventArgs e)
    {
        Application.Exit();
    }
}

```

Лістинг Form4:

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Globalization;
using System.IO;
using System.Windows.Forms;

namespace Speed_Typing_App
{
    public partial class Form4 : Form
    {
        //основні класи
        public class TextToPrint
        {
            public string TextTPrint { get; set; }
            public int SymbCount = 0;
        }
        public class Input
        {
            public string Text;
            public double acc;
            public double wordcount = 0;
            public double time;
            public double Time
            {
                get { return time; }
                set { time = value; }
            }
        }
        public class Result
        {
            public readonly double words;

            public readonly DateTime date;

            public Result(double words, DateTime date)
            {
                this.words = words;
                this.date = date;
            }
        }
    }
}

```

```

    }
    public class Music
    {
        public static WMPLib.WindowsMediaPlayer WMP = new
WMPLib.WindowsMediaPlayer();
        Timer tmr = new Timer();
        public void play_music_1()
        {
            tmr.Interval = 10;
            tmr.Stop();
            WMP.URL = @"2.mp3";
            WMP.settings.volume = 20;
            WMP.controls.play();
            tmr.Tick += new EventHandler(tmr_Tick);
            WMP.PlayStateChange += new
WMPLib._WMPOCXEvents_PlayStateChangeEventHandler(wplayer_PlayStateChange);
        }

        void tmr_Tick(object sender, EventArgs e)
        {
            WMP.controls.stop();
            play_music_1();
        }
        void wplayer_PlayStateChange(int NewState)
        {
            if (NewState == (int)WMPLib.WMPPlayState.wmppsMediaEnded)
            {
                tmr.Start();
            }
        }
        public void StopMusic()
        {
            WMP.controls.stop();
        }
    }
    //загальні змінні
    int ticks = 6;
    int ticks1 = 10;
    string text1 = "";
    int length = 0;
    int misc = 0, lng1=0, vid=0;
    bool fl = true, fl1 = true, musfl = false, colfl=true;
    Input input = new Input();
    TextToPrint textToPrnt = new TextToPrint();
    Input name = new Input();
    Music music = new Music();
    string[] lines = File.ReadAllLines("WOSrecords.txt");
    string[] linesToWrite = new string[100];
    public Form4()
    {
        InitializeComponent();
    }
    private void Form4_Load(object sender, EventArgs e)
    {
        TextPanel.ReadOnly = true;
        textBox1.ReadOnly = true;
        label2.Visible = false;
        label3.Visible = false;
    }
    //почати
    private void button1_Click_2(object sender, EventArgs e)
    {
        nameBox.Visible = false;

```

```

        nameLabel.Visible = false;
        StartButton.Visible = false;
        if (nameBox.Text == "") name.Text = "Unknown Player";
        else name.Text = nameBox.Text;
        label4.Visible = true;
        label1.Visible = true;
        timer1.Start();
        GenerateSent();
        TextPanel.Text = textToPrnt.TextTPrint;
    }
    //згенерувати слово
    void GenerateSent()
    {
        string[] readText = File.ReadAllLines("words.txt");
        Random random = new Random();
        int i = random.Next(readText.Length);
        textToPrnt.TextTPrint = readText[i];
    }
    //таймер відліку
    private void timer1_Tick(object sender, EventArgs e)
    {
        ticks--;
        timer1.Interval = 1000;
        if (ticks < 0 && fl == true)
        {
            timer1.Interval = 1;
            label1.Visible = false;
            textBox1.ReadOnly = false;
            label2.Visible = true;
            label3.Visible = true;
            label4.Visible = false;
            timer2.Start();

        }
        label1.Text = ticks.ToString();
        ChangeClr1(TextPanel, input);
        CheckOnEnd(input);
        if (ticks1 < 1 && fl == true)
        {
            fl = false;
            timer2.Stop();
            label3.Visible = false;
            label2.Visible = false;
            CheckOnRecord(input.wordcount);
            Print(input);
        }
    }
    //запис результату у файл
    public void CheckOnRecord(double wordsAmount)
    {
        List<string> list = new List<string>();
        bool swap = false;
        int number;
        lines.CopyTo(linesToWrite, 0);
        for (int i = 0; i < linesToWrite.Length; i++)
        {
            number = 0;
            if (!string.IsNullOrEmpty(linesToWrite[i]))
            {
                string[] words = linesToWrite[i].Split(' ');
                number = int.Parse(words[0]);
            }
            if (swap)

```

```

        {
            list.Add(linesToWrite[i - 1]);
        }
        if (wordsAmount > number && !swap)
        {
            swap = true;
            list.Add($"{wordsAmount} words, Player:{name.Text}
{DateTime.Now.ToShortDateString()}");
        }
        else if (!swap)
        {
            list.Add(linesToWrite[i]);
        }
    }
    File.WriteAllLines("WOSrecords.txt", list);
}
//таймер до кінця
private void timer2_Tick(object sender, EventArgs e)
{
    if (checkBox1.Checked == true && musfl == false)
    {
        music.play_music_1();
        musfl = true;
    }
    else if (checkBox1.Checked == false)
    {
        music.StopMusic();
        musfl = false;
    }
    ticks1--;
    timer2.Interval = 1000;
    label2.Text = ticks1.ToString();
}
//перевірка коректності вводу
void ChangeClr1(RichTextBox TextPanel, Input input)
{
    input.Text = textBox1.Text;
    int lng = input.Text.Length;
    string sub="";
    try
    {
        fl1 = true;
        sub = textToPrnt.TextTPrint.Substring(0, lng);
    }
    catch
    {
        timer1.Enabled = false;
        timer2.Enabled = false;
        MessageBox.Show("Too much symbols!");
        textBox1.Text = null;
        TextPanel.Select(0, textToPrnt.TextTPrint.Length);
        TextPanel.SelectionColor = Color.Black;
    }
    if (fl1)
    {
        timer1.Enabled = true;
        timer2.Enabled = true;
    }
    string text2 = input.Text;
    if (sub == input.Text)
    {
        if (lng1 < lng)
        {
            colfl = true;

```

```

    }
    if (input.Text.Length > 0 && colfl == true)
    {
        lng1 = lng;
        colfl = false;
        TextPanel.Select(vid, 1);
        TextPanel.SelectionColor = Color.WhiteSmoke;
        vid++;
    }
    TextPanel.BackColor = Color.LightGreen;
}
else if (sub != input.Text)
{
    TextPanel.BackColor = Color.LightCoral;

    if (text1 != text2 && length > lng)
    {
        misc++;
    }
}
text1 = input.Text;
length = lng;
}
//перевірка на кінець гри
void CheckOnEnd(Input input)
{
    input.Text = textBox1.Text;
    if (input.Text.Length == textToPrnt.TextTPrint.Length
        && input.Text == textToPrnt.TextTPrint)
    {
        if (ticks1 >= 0)
        {
            vid = 0;
            lng1 = 0;
            input.wordcount++;
            textBox1.Text = null;
            GenerateSent();
            TextPanel.Text = textToPrnt.TextTPrint;
            vid = 0;
        }
    }
}
//виведення результату
void Print(Input input)
{
    MessageBox.Show($"❖'ведено слів:{input.wordcount:f0}\n
Помилка:{misc:f0}");
    Form4 form = new Form4();
    this.Hide();
    form.Show();
}

//до меню
private void ReturnToMenu_Click_1(object sender, EventArgs e)
{
    music.StopMusic();
    Form3 form3 = new Form3();
    this.Close();
    form3.Show();
}
}
}

```

