

Лабораторная работа № 7

Тема: Таймеры. Асинхронность в Windows Forms.

Цель: закрепить теоретические знания и применить их на практике.

Задание:

Вариант №1.

Задание 1. 1. Приложение-секундомер с использованием Timer.

Создайте форму с меткой (Label).

Необходимо реализовать:

- Отображение времени в формате ЧЧ:ММ:СС
- Кнопки "Старт", "Пауза", "Сброс"
- *Возможность сохранения промежуточных результатов в ListView

Задание 2. Смена цветов формы.

Создайте приложение, которое меняет цвет фона формы каждые 2 секунды.

Используйте массив из 5-6 заранее заданных цветов.

Реализуйте циклическую смену цветов с помощью Timer.

Добавьте кнопку "Остановить", которая фиксирует текущий цвет и останавливает таймер.

Задание 3. Асинхронное обновление метки.

Создайте форму с кнопкой и меткой (Label).

Напишите асинхронный метод, который изменяет текст метки через 3 секунды после нажатия кнопки.

Используйте Task.Delay и ключевые слова async и await, чтобы выполнить это действие без блокировки интерфейса.

Задание 4. Асинхронный калькулятор факториала

Создайте форму с NumericUpDown для ввода числа, кнопкой Вычислить и меткой для отображения результата. Вычисление факториала должно происходить асинхронно.

Приложение должно содержать:

- NumericUpDown для ввода числа.
- Кнопку Вычислить, запускающую вычисление асинхронно.
- Метку для вывода результата.
- *ProgressBar, который будет показывать прогресс во время вычисления.

Подсказка:

- Используйте Task.Run(() => { ... }) для выполнения вычислений в фоновом потоке.

Задание 5. Асинхронное чтение файла

Создайте приложение, которое позволяет выбрать текстовый файл (.txt), загрузить его содержимое и отобразить в TextBox.

Требования:

- Кнопка Открыть файл, которая вызывает OpenFileDialog.
- TextBox для отображения содержимого файла.
- Асинхронный метод ReadAllTextAsync.

Подсказка:

- Используйте await File.ReadAllTextAsync(path) для чтения файла.

Вариант №2.

Задание 1. Приложение-часы с будильником.

Создайте форму с меткой (Label).

Отображение текущего времени с помощью Timer

Возможность установки времени будильника

Включение/выключение будильника

Звуковой сигнал при срабатывании (используйте System.Media.SoundPlayer) и (или) изменение цвета фона метки. Вывести сообщение "Время вставать!"

Добавить кнопку для отключения будильника.

Задание 2. Анимация текста.

Создайте приложение, где заданный текст появляется постепенно, символ за символом. После полного появления текст исчезает и процесс повторяется. Используйте компонент **Timer**.

Задание 3. Простая асинхронная задача

Создайте форму с кнопкой и меткой (Label).

Напишите асинхронный метод, который увеличивает значение числа на метке каждую секунду до достижения значения 10.

Используйте `Task.Delay` и ключевые слова `async` и `await`, чтобы выполнять это действие без блокировки интерфейса.

Задание 4. Асинхронный калькулятор числа Фибоначчи в Windows Forms

Создайте приложение Windows Forms, которое вычисляет число Фибоначчи по введенному пользователем индексу.

Приложение должно содержать:

- `NumericUpDown` для ввода номера числа Фибоначчи.
- Кнопку Вычислить, запускающую вычисление асинхронно.
- Метку для вывода результата.
- `*ProgressBar`, который будет показывать прогресс во время вычисления.

Подсказка:

- Используйте `Task.Run(() => { ... })` для выполнения вычислений в фоновом потоке.

Задание 5. Асинхронная запись в файл

Приложение должно позволять пользователю вводить текст и сохранять его в файл.

Требования:

- `RichTextBox` для ввода текста.
- Кнопка Сохранить, которая вызывает `SaveFileDialog`.
- Асинхронный метод `WriteAllTextAsync`.

Подсказка:

- Используйте `await File.WriteAllTextAsync(path, text)`

Отчет должен содержать (см. образец):

- номер и тему лабораторной работы;
- фамилию, номер группы студента и вариант задания;
- скриншоты окна Visual Studio с исходным кодом программы и комментариями;
- скриншоты с результатами выполнения программ;
- пояснения, если необходимо;
- выводы.

Отчеты в формате **pdf** отправлять на email: **colledge20education23@gmail.com**