

Практическая работа №15_2

Тема: Разработка приложений с не визуальными компонентами

Цель работы: формирование навыков создания приложений Windows Forms с использованием таймера. Изучение классов, реализующих задачу программирования печати и получение навыков по работе в программе с диалоговыми окнами

Задачи:

- совершенствование приемов создания приложений Windows Forms;
- создание проектов с использованием диалоговых окон, таймера.

Материально-техническое обеспечение:

Место проведения: Компьютерный класс.

Время на выполнение работы: 2 часа.

Оборудование: ПК

Средства обучения: операционная система, текстовый процессор MS Word, программные средства определенного вида

Исходные данные:

1. Конспект занятия.
2. Задание для практической работы.

Перечень справочной литературы:

1) Программирование на языке высокого уровня. Программирование на языке C++: учеб. пособие / Т. И. Немцова, С. Ю. Голова, А. И. Терентьев ; под ред. Л. Г. Гагариной. – М. : ИД «ФОРУМ» : ИНФРА-М, 2018. – 512 с. – (Среднее профессиональное образование).

Краткие теоретические сведения:

Создание многооконных приложений

Microsoft Visual Studio позволяет создавать две разновидности многооконных приложений: **SDI** - и **MDI**-приложения. **SDI-приложения** состоят из нескольких независимых форм (окон). По умолчанию будет создано SDI-приложение. В **MDI-приложении** имеется одна главная форма, остальные формы находятся в пределах главной; из главной формы можно управлять подчиненными формами. Единственное меню MDI-приложения находится в главном окне.

Перед созданием многооконного приложения его необходимо проектировать: продумать вопрос о том, какие окна нужны и что на них будет отображено. **Форма** – это разновидность класса. Экземпляры классов, как известно, необходимо создавать. Это правило распространяется и на формы: автоматически создается лишь одна форма – **главная**. Создание всех остальных форм лежит на программисте. Закрытие формы функцией **Close()**; или нажатием на кнопку **x** вызывает уничтожение формы, и в случае необходимости она должна быть создана заново.

В принципе, любую задачу можно решить, как с помощью SDI-приложения, так и с помощью MDI-приложения. Пожалуй, создание SDI-приложения проще. MDI-приложение можно рекомендовать при необходимости создать и работать одновременно с несколькими одинаковыми формами.

Использование таймера

Иногда в приложениях, работающих в графическом интерфейсе Windows, возникает необходимость отображения текущего времени суток (часы:минуты:секунды). Для реализации этой задачи в аппаратуре компьютера предусмотрен доступ к таймеру.

Программируемый таймер предназначен для получения программно-управляемых временных задержек и генерации функций, задающих время. Программируемый таймер разрешает формировать заданные интервалы времени независимо от быстродействия вычислительной системы.

Система Microsoft Visual Studio представляет для пользователя элемент управления **Timer**, реализующий выполнение прерывания от таймера. Также с помощью компонента **Timer** можно организовать параллельное выполнение различных процессов в программе.

Неотображаемый на форме компонент **Timer** предназначен для запуска периодически повторяющихся действий. Свойство **Interval** задает период (в миллисекундах), с которым будет повторяться событие **Tick**. При установке свойства **Enabled = true** таймер включается вместе с

запуском приложения. Метод **Start()** запускает, а **Stop()** останавливает таймер.

Типичные примеры использования таймера – часы и секундомер (вывод времени и даты), а также анимация (имитация плавного изменения положения, размеров и формы объектов).

Ход работы:

Требования к содержанию отчета:

- Номер и название практической работы.
- Цель работы.
- По каждой заданию (задаче/примеру) экранные формы (при наличии) и листинг программного кода, показывающие порядок выполнения практической работы, и результаты, полученные в ходе её выполнения.
- Ответы на контрольные вопросы в тетради.

Порядок выполнения работы:

Все задания практической работы размещать в своей папке проектов в новой папке

Пр15_2_Фамилия

В начале каждого файла проекта установить комментарии: пр.р.№_____ (указать номер), свою Фамилию. Формулировку задания

Задания. Создание приложения с использованием таймера.

Пример 1. Простые часы. Вывод времени и даты по таймеру.

1. Запустите **MS Visual Studio**. Создайте новый проект **Приложение Windows Forms (.NET Framework)** для **C#**, имя проекта **pr15_Таймер_Фамилия**.
2. Разместим на форме размером **540*230** две надписи и зададим их свойства:
 - **label1** для вывода времени (свойства: **BackColor = Green, ForeColor = Yellow, Text = 00:00:00**, размер шрифта **Font = 72**);
 - **label2** для вывода даты (**ForeColor = Olive, Text = дата, Font = 16**).
3. Из категории **Компоненты** (Components) панели элементов перетащите на форму **Timer**. Его значок отобразится в нижней части окна **Конструктор** (рис.1). Задайте его свойства: **Enabled = true, Interval = 100**.

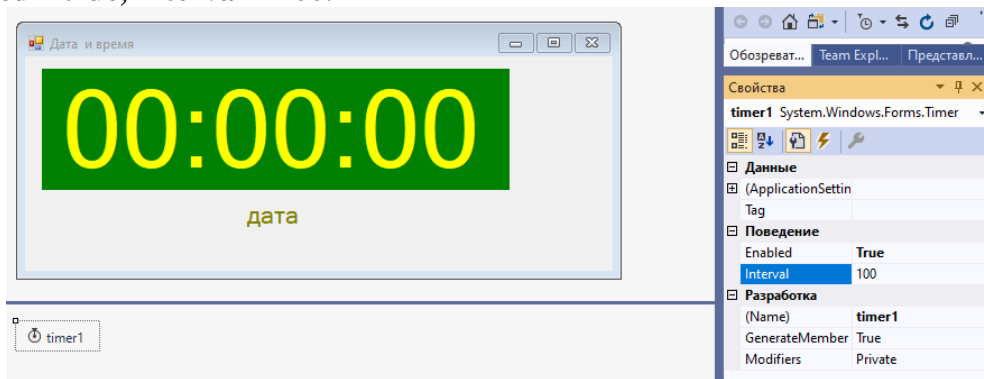
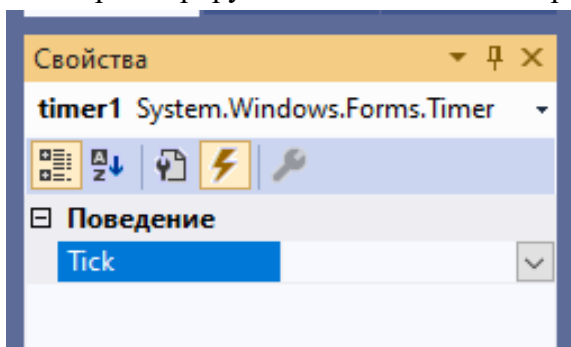


Рис.1. Настройка элементов формы и таймера

4. Зарегистрируйте событие **Tick** таймера.



В шаблон обработчика введите код:

```

20     private void timer1_Tick(object sender, EventArgs e)
21     {
22         label1.Text = DateTime.Now.ToLongTimeString(); // вывод времени
23         label2.Text = DateTime.Now.ToLongDateString(); // вывод даты
24     }

```

5. Протестируйте программу. При необходимости откорректируйте свойства компонентов и программный код.

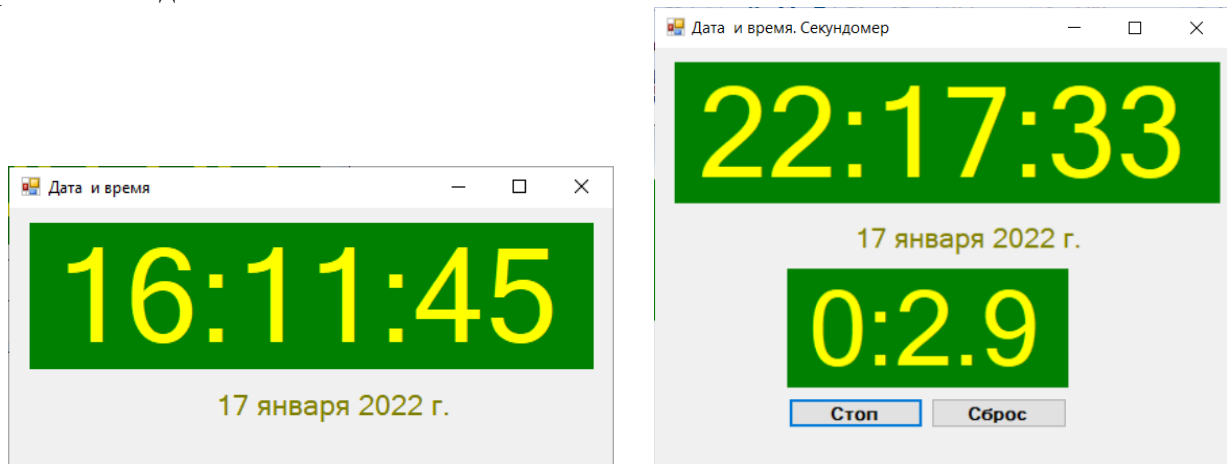


Рис. 2. Простые часы (а) и секундомер (б)

Пример 2. Секундомер. Вывод минут и секунд.

1. Внесите изменения в проект задания 1: разместите на форме размером **380*200** надпись **label1** (свойства **BackColor** = **Green**, **ForeColor** = **Yellow**, **Text** = **0:0.0**, **Font** = **60**) и две кнопки **Старт** и **Сброс** с соответствующими именами в свойстве **Name** (рис. 2, б).
2. Перетащите на форму **Timer** (свойства **Enabled** = **true**, **Interval** = **100**).
3. Зарегистрируйте события **Tick** таймера, а также нажатий кнопок **Click**.

В шаблоны обработчиков введите коды:

```

15     private int m, s, ms; // объявление полей

21     // нажатие кнопки Старт/Стоп
22     ссылка: 1
23     private void Stop_Click(object sender, EventArgs e)
24     {
25         if (timer2.Enabled)
26         {
27             timer2.Stop();
28             StopButton.Text = "Старт"; } // остановка таймера
29         else {
30             timer2.Start();
31             StopButton.Text = "Стоп"; } // запуск таймера
32     }
33     // нажатие кнопки Сброс, обнуление значений
34     ссылка: 1
35     private void SбросButton_Click(object sender, EventArgs e)
36     {
37         m = 0; s = 0; ms = 0;
38         label3.Text = "0:0.0";
39     }

```

```

45 // счет и вывод по событию Tick таймера
46 ссылка: 1 private void timer2_Tick(object sender, EventArgs e)
47 {
48     ms++;
49     s = ms / 10;
50     m = s / 60;
51     label3.Text = m + ":" + s % 60 + "." + ms % 10;
52 }
53

```

4. Протестируйте программу. При необходимости откорректируйте код и свойства компонентов.

Контрольные вопросы:

- 1) компонент Timer: назначение, в каких случаях используется, свойства.

Задания для самостоятельной работы

1. Создайте секундомер с одной кнопкой **Сброс**, который запускается и останавливается щелчком мыши по форме.
2. Создайте приложение, которое по введенной дате (год, месяц и число) показывает день недели.
3. Создайте часы, которые показывают, сколько дней (часов, минут) осталось до наступления нового года.