# Практическая работа

## Создание проекта с использованием компонентов ввода и отображения чисел, дат и времени

## 1. Цель работы. Получение навыков составления программ в среде Visual Studio.Net

## 2. Литература. Т.А.Павловская “С#. Программирование на языке высокого уровня”, И.Г.Семакин, А.П.Шестаков «Основы программирования», Т.А.Павловская, Ю.А.Щупак «С/С++. Структурное программирование. Практикум», Е.Л.Шиманович «С/С++ в примерах и задачах»

## 3. Подготовка к работе. Повторить конспект лекций по темам «Визуально-событийно управляемое программирование», «Компоненты и их свойства».

## 4. Перечень оборудования. Интегрированная среда Visual Studio.Net.

## 5. Задание.

1. Научиться писать приложения в интегрированной среде Visual Studio.Net.

***6. Теоретический материал***

**Создание DispatchTimer**

XAML не поддерживает функции таймера, а WPF не имеет элемента управления или класса Timer. Класс DispatchTimer, определенный в пространстве имен **System.Windows.Threading**, используется для добавления функциональности таймера в WPF.

Следующий фрагмент кода создает объект DispatchTimer.

DispatcherTimer dispatcherTimer = newDispatcherTimer();

**Настройка тика и интервала**

Обработчик событий Tick выполняется при запуске DispatchTimer на заданном интервале. В следующем фрагменте кода задаются такт и интервал таймера DispatchTimer.

dispatcherTimer.Tick += newEventHandler(dispatcherTimer\_Tick);

dispatcherTimer.Interval = newTimeSpan(0, 0, 1);

**Запустить DispatchTimer**

Метод Start используется для запуска DispatchTimer.

dispatcherTimer.Start();

**Полный пример кода**

Фрагмент кода создает DispatchTimer, устанавливает его событие Tick и свойство Interval и вызывает его метод Start. Метод Start запускает таймер, и обработчик события Tick выполняется на заданном значении Interval. В этом коде в обработчике событий Tick выполняются необходимые действия.

DispatcherTimer dispatcherTimer = newDispatcherTimer();

dispatcherTimer.Tick += DispatcherTimer\_Tick

dispatcherTimer.Interval = new TimeSpan(0, 0, 1);

dispatcherTimer.Start();

privatevoid dispatcherTimer\_Tick(object sender, EventArgs e)

{

}

***7. Порядок выполнения работы***

* + - 1. **Будильник**
* позволяет установить время срабатывания,
* если выбранное время уже прошло — будильник будет перенесён на завтра,
* показывает сколько времени осталось до срабатывания,
* и выводит сообщение "Вставай!", когда наступает нужное время.
* при необходимости позволяет установить дату срабатывания (нельзя устанавливать дату, которая уже прошла)
  + - 1. **Таймер**
* вводит время, на которое запускается таймер (например, 5 минут);
* таймер отсчитывает время в обратную сторону;
* по завершении выводится сообщение ("Время вышло!")

***Содержание отчета.***

7.1. Название и цель практической работы.

7.2. Составленные тексты программ задач с ответами в рабочей тетради.

## 8. Приложение.