# Работа с обработчиками событий

**ЗАДАНИЕ 1. ТАЙМЕР ОБРАТНОГО ОТСЧЕТА**

**Порядок выполнения работы**

1. Создайте решение и проект **Example\_1**, выбрав шаблон Приложение Windows Forms.
2. В **Обозревателе решений** изучите структуру проекта, выделите пункт **Form1.cs** и вызовите конструктор формы (**Вид – Конструктор**).
3. В конструкторе формы (**вкладка Form1.cs [Конструктор]**) выделите форму и перейдите в окно

## Свойства (Вид – Окно свойств).

### Установите следующие свойства формы:

Таблица 1. Свойства главной формы проекта Example\_1

|  |  |  |
| --- | --- | --- |
| **Свойство** | **Значение** | **Описание** |
| Name | mainForm | Имя компонента. |
| Text | Таймер обратного отсчета | Заголовок окна. |
| Icon | Изображение (\*.ico) | Иконка окна. |
| Size | (400;200) | Размер окна в пикселях. |
| StartPosition | CenterScreen | Расположение окна при запуске  приложения (по центру экрана). |

1. Откройте окно **Панель элементов** (**Вид – Панель элементов**).

### Разместите на форме компоненты согласно рисунку 1 и таблице 2:

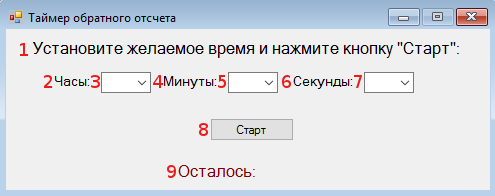


Рисунок 1. Примерный вид главной формы проекта Example\_1.

Таблица 2. Типы и имена компонентов главной формы Example\_1

|  |  |  |
| --- | --- | --- |
| **№** | **Тип компонента** | **Имя (св-во Name)** |
| 1 | Label | labelInfo |
| 2 | labelHours |
| 4 | labelMinutes |
| 6 | labelSeconds |
| 9 | labelTime |
| 3 | ComboBox | comboBoxHours |
| 5 | comboBoxMinutes |
| 7 | comboBoxSeconds |
| 8 | Button | buttonStart |

### У главной формы создайте обработчик события **Load**, выполняемого при ее первоначальном появлении:

* 1. выделите форму;
  2. в окне **Свойства** перейдите на вкладку **События**;

### найдите в списке событие **Load** и дважды щелкните левой кнопкой мыши;

* 1. в создавшийся шаблон функции добавьте следующий программный код:

private void mainForm\_Load(object sender, EventArgs e)

{

//заполнение списка значениями часов

for (int i = 0; i <= 24; i++) comboBoxHours.Items.Add(i);

//заполнение списка значениями минут

for (int i = 0; i <= 59; i++) comboBoxMinutes.Items.Add(i);

//заполнение списка значениями секунд

for (int i = 0; i <= 59; i++) comboBoxSeconds.Items.Add(i);

//установка начального значения для всех списков, равного первому пункту comboBoxHours.SelectedIndex = comboBoxMinutes.SelectedIndex = comboBoxSeconds.SelectedIndex = 0;

//до начала отсчета скрываем вывод времени

labelTime.Visible = false;

}

### Для всех компонентов **ComboBox** обработайте событие ввода с клавиатуры:

* 1. создайте функцию со следующим программным кодом (через Свойства – События):

private void comboBox\_KeyPress(object sender, KeyPressEventArgs e)

{

ComboBox cmbBox = (dynamic)sender; //создаем общий объект для выпадающих списков

//разрешаем ввод только 2х цифр и нажатие клавиши backspace

if ((((e.KeyChar < '0') || (e.KeyChar > '9')) && (e.KeyChar != '\b')) || ((cmbBox.Text.Length == 2) && (e.KeyChar != '\b')))

{

e.Handled = true; //событие не требует передачи обработчику по умолчанию

}

}

### в окне свойств в качестве обработчиков события **KeyPress** выберите созданную функцию.

1. Для того чтобы запретить использование пустых полей, для всех выпадающих списков создайте обработчик события **Leave**, выполняемого при потере компонентом фокуса ввода (можно прописать вручную в коде):

private void comboBox\_Leave(object sender, EventArgs e)

{

ComboBox cmbBox = (dynamic)sender; //создаем общий объект для выпадающих списков

if (cmbBox.Text == "") //если в поле списка введено пустое значение

buttonStart.Enabled = false; //блокируем кнопку "старт" else //иначе

buttonStart.Enabled = true; //снимаем блокировку

}

### В классе **mainForm** объявите необходимые переменные:

public partial class MainForm : Form

{

public DateTime timerValue; //время для отсчета

public DateTime timerNullValue = new DateTime(2020,07,11,0,0,0);//нулевое значение

public TimeSpan oneSecond = new TimeSpan(0, 0, 1); //интервал в 1 секунду

public bool IsTimerOn = false; //таймер не установлен

### Добавьте на форму невизуальный компонент таймер (**Timer**).

1. Двойным щелчком по иконке компонента создайте обработчик события **Tick** и поместите в него следующий код:

private void timer1\_Tick(object sender, EventArgs e)//действия,выполняемые по таймеру

{

if (timerValue == timerNullValue) //если время стало равно 0:00:00

{

timer1.Enabled = false; //выключаем таймер

MessageBox.Show("Время вышло"); //выводим сообщение на экран

buttonStart.Text = "Старт

IsTimerOn = false; //сбрасываем флаг (таймер не установлен)

labelTime.Visible = false;

}

else

{

timerValue = timerValue - oneSecond; //уменьшаем время на 1 секунду

//выводим время на экран

labelTime.Text = "Осталось: " + System.Convert.ToString(timerValue.TimeOfDay);

}

}

### При нажатии на кнопку **buttonStart** реализуйте запуск таймера обратного отсчета и его приостановку:

private void buttonStart\_Click(object sender, EventArgs e)

{

if (!IsTimerOn) //если таймер не установлен

{

//сбрасываем переменную, содержащую время

timerValue = new DateTime(2020, 07, 11, 0, 0, 0);

//устанавливаем количество часов

timerValue = timerValue.AddHours(System.Convert.ToInt32(comboBoxHours.Text));

//устанавливаем количество минут

timerValue = timerValue.AddMinutes(System.Convert.ToInt32(comboBoxMinutes.Text));

//устанавливаем количество секунд

timerValue = timerValue.AddSeconds(System.Convert.ToInt32(comboBoxSeconds.Text));

//выводим на экран время

labelTime.Text = "Осталось: " + System.Convert.ToString(timerValue.TimeOfDay); timer1.Enabled = true; //запускаем таймер

IsTimerOn = true; buttonStart.Text = "Пауза"; labelTime.Visible = true;

}

else

if (IsTimerOn && timer1.Enabled) //если таймер установлен и запущен

{

timer1.Enabled = false; //делаем паузу

buttonStart.Text ="Продолжить";

}

else

if (IsTimerOn && !timer1.Enabled) //если таймер установлен и не запущен

{

timer1.Enabled = true; //возобновляем обратный отсчет

buttonStart.Text = "Пауза";

}

}

1. Выполните отладку приложения (**Отладка – Начать отладку**) и проверьте правильность его работы.

### **Для самостоятельного выполнения (по желанию)**

### **Задание 2:** Калькулятор для расчета количества кафельной плитки.

1. Создайте новое приложение **Example\_2** типа **Windows Forms**.

### Для главной формы установите имя **mainForm**.

1. Разместите компоненты согласно рисунку 2 и таблице 3 отредактируйте их внешний вид:

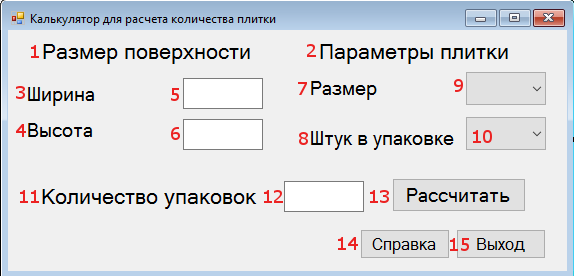


Рисунок 2. Примерный вид главной формы Example\_2

Таблица 3.Типы и имена компонентов главной формы Example\_2

|  |  |  |
| --- | --- | --- |
| **№** | **Тип компонента** | **Имя (св-во Name)** |
| 1 | Label | labelSize |
| 2 | labelParams |
| 3 | labelWidth |
| 4 | labelHeight |
| 7 | labelSizeTile |
| 8 | labelCountTile |
| 11 | labelResult |

Таблица 4.Типы и имена компонентов главной формы Example\_2 (продолжение)

|  |  |  |
| --- | --- | --- |
| 5 | TextBox | textBoxWidth |
| 6 | textBoxHeight |
| 12 | textBoxResult |
| 9 | ComboBox | comboBoxTileSize |
| 10 | comboBoxTileCount |
| 13 | Button | buttonStart |
| 14 | buttonHelp |
| 15 | buttonExit |

### Запретите ввод любых символов, кроме цифр в текстовые поля, создав для них обработчик события **KeyPress**:

private void textBox\_KeyPress(object sender, KeyPressEventArgs e)

{

TextBox txtBox = (dynamic)sender; //преобразуем к общему объекту

//запрещаем ввод любых символов, кроме цифр и клавиши backspace

//запрещаем ввод, если в поле уже 4 символа

if ((((e.KeyChar < '0') || (e.KeyChar > '9')) && (e.KeyChar != '\b')) || ((txtBox.Text.Length > 3) && (e.KeyChar != '\b')))

{

e.Handled = true;

}

}

### Запустите программу и проверьте правильность работы текстовых полей.

1. Заполните выпадающие списки:
   1. в классе **Form1** объявите следующие переменные:

public partial class Form1 : Form

{

private static short n = 4;

private short [,] tileSize = new short[n, 2]; //список возможных размеров плитки

private short [] tileCount = new short [n]; //список для выбора количества плитки

### в конструкторе формы заполните массивы исходными данными:

public Form1()

{

//заполнение списка "размер плитки" tileSize[0, 0] = 10; //ширина

tileSize[0, 1] = 10; //высота

tileSize[1, 0] = 15;

tileSize[1, 1] = 15;

tileSize[2, 0] = 10;

tileSize[2, 1] = 15;

tileSize[3, 0] = 20;

tileSize[3, 1] = 20;

//заполение списка "количество" tileCount[0] = 8;

tileCount[1] = 10;

tileCount[2] = 12;

tileCount[3] = 20;

### при загрузке формы заполните выпадающие списки значениями из массивов:

for (int i = 0; i < n; i++) //заполняем список "размеры плитки" comboBoxTileSize.Items.Add(String.Format("{0}\*{1}", tileSize[i, 0], tileSize[i, 1]));

for (int i = 0; i < n; i++) //заполняем список "количество" comboBoxTileCount.Items.Add(String.Format("{0}", tileCount[i]));

### запретите пользователю ввод в поля списка, установив значение свойства **DropDownStyle**

равным **DropDownList**;

### установите значения по умолчанию:

//по умолчанию выбран 1й пункт

comboBoxTileSize.SelectedIndex = 0;

comboBoxTileCount.SelectedIndex = 0;

### запустите программу и проверьте правильность заполнения списков.

### Создайте программный код для события нажатия на кнопку «Расчет» (компонент **buttonResult**):

private void buttonResult\_Click(object sender, EventArgs e)

{

double w = Convert.ToDouble(textBoxWidth.Text); //ширина поверхности

double h = Convert.ToDouble(textBoxHeight.Text); //высота поверхности

double a = tileSize[comboBoxTileSize.SelectedIndex, 0]; //ширина плитки

double b = tileSize[comboBoxTileSize.SelectedIndex, 1]; //высота плитки

short c = tileCount[comboBoxTileCount.SelectedIndex]; //кол-во плиток в упаковке

//расчет количества упаковок c "округлением" результата в большую сторону

double res = Math.Ceiling((w / a) \* (h / b) / c);

textBoxResult.Text = res.ToString(); //вывод результата в текстовое поле

}

### Запустите программу и проверьте правильность выполнения вычислений.

1. Реализуйте механизм выхода и программы:
   1. создайте обработчик события **FormClosing** главной формы, возникающего при закрытии формы:

private void Form1\_FormClosing(object sender, FormClosingEventArgs e)

{

//выводим диалоговое окно, если пользователь нажал кнопку Да

if (MessageBox.Show("Выйти из программы?", Application.ProductName, MessageBoxButtons.YesNo, MessageBoxIcon.Question) == DialogResult.Yes)

e.Cancel = false; //разрешаем выход из программы

else

}

e.Cancel = true; //иначе не закрываем программу

### при нажатии на кнопку Выход (**buttonExit**) вызовите метод завершения работы приложения:

Application.Exit();

### запустите программу и проверьте работу кнопки Выход, а так же закрытие формы с помощью кнопки в строке заголовка.

1. Реализуйте вывод справочной информации:
   1. добавьте в проект новую форму с именем **Form2.cs**: **Проект – Добавить форму Windows…**;

### в свойстве **Text** определите подпись окна Справка.

* 1. установите свойство **FormBorderStyle** равным **FixedDialog** (стиль рамки окна);
  2. разместите компонент **Textbox** с именем **textBoxInfo**;

### установите многострочный режим текстового поля (свойство **Multiline** должно быть равным

**true**);

### растяните текстовое поле на всю форму (свойство **Dock** должно быть равным **Fill**);

* 1. в поле **Text** введите справочную информацию, содержащую пояснение по работе с программой;
  2. сделайте текстовое поле доступным только для чтения;
  3. при нажатии на кнопку **Справка** на главной форме реализуйте появление окна справки:

private void buttonHelp\_Click(object sender, EventArgs e)

{

Form2 formHelp = new Form2(); //создание экземпляра формы

formHelp.ShowDialog(); //открытие формы в модальном режиме

}

### запустите программу и проверьте правильность работы