

Лабораторна робота №8. Елемент Timer. Додаток «Друкарська машинка»

Тема: Windows Forms. Елемент Timer. Додаток «Друкарська машинка».

Мета заняття: навчитись працювати з елементом Timer та його обробником події Tick, навчитись працювати з елементом StatusStrip та обробником подій форми Key_Down з допомогою Windows Forms мовою C# у середовищі Microsoft Visual Studio.

1. Теоретичні відомості

1.1 Елемент Timer

Timer є компонентом для запуску дій, що повторюються через певний проміжок часу. Хоча він не є візуальним елементом, але його також можна перетягнути з Панелі Інструментів на форму (рис. 1.1):

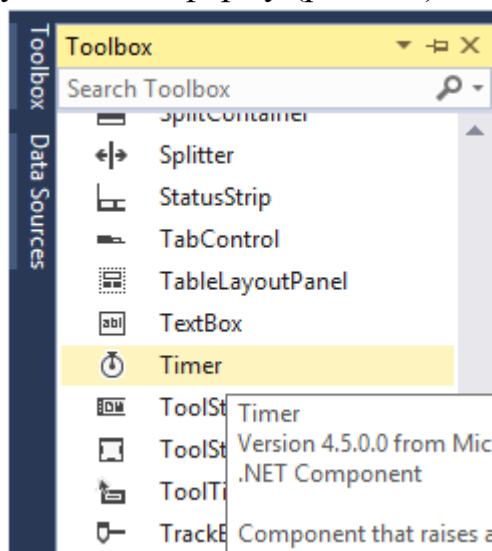


Рисунок 1.1 – Елемент таймер на панелі інструментів у середовищі Microsoft Visual Studio

Найбільш важливі властивості і методи таймера:

- властивість `Enabled`: при значенні `true` вказує, що таймер буде запускатися разом з запуском форми;
- властивість `Interval` : вказує інтервал в мілісекундах, через який буде спрацьовувати обробник події `Tick`;
- метод `Start ()` : запускає таймер;
- метод `Stop ()` : зупиняє таймер.

Для прикладу визначимо просту форму, на яку додаємо кнопку і таймер (рис. 1.2). У файлі коду форми визначимо такий код:

```

public partial class Form1 : Form
{
    int koef = 1;
    public Form1()
    {
        InitializeComponent();

        this.Width = 400;
        button1.Width = 40;
        button1.Left = 40;
        button1.Text = "";
        button1.BackColor = Color.Aqua;

        timer1.Interval = 500; // 500 мс
        timer1.Enabled = true;
        button1.Click += button1_Click;
        timer1.Tick += timer1_Tick;
    }
    // обробник події Tick таймера
    void timer1_Tick(object sender, EventArgs e)
    {
        if (button1.Left == (this.Width-button1.Width-10))
        {
            koef=-1;
        }
        else if (button1.Left == 0)
        {
            koef = 1;
        }
        button1.Left += 10 *koef;
    }
    // обробник натиснення на кнопку
    void button1_Click(object sender, EventArgs e)
    {
        if(timer1.Enabled==true)
        {
            timer1.Stop();
        }
        else
        {
            timer1.Start();
        }
    }
}

```

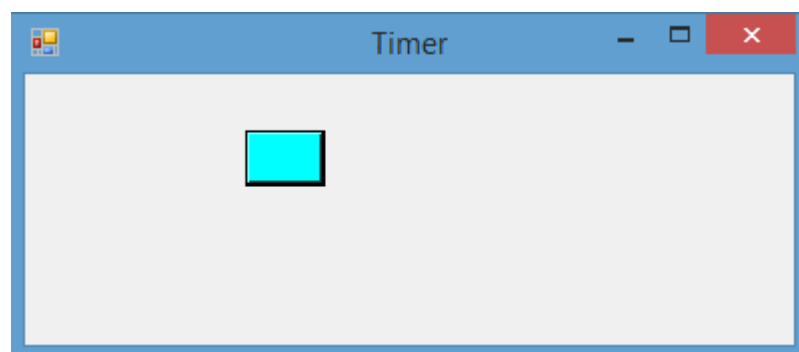


Рисунок 1.2 – Вікно форми для роботи з елементом Timer

У конструкторі форми встановлюються початкові значення для таймера, кнопки і форми. Через кожен інтервал таймера буде спрацьовувати обробник `timer1_Tick`, в якому змінюється положення кнопки по горизонталі за допомогою властивості `button1.Left`. А за допомогою додаткової змінної `coef` можна управляти напрямком руху.

Крім того, за допомогою обробника натискання кнопки `button1_Click` можна або зупинити таймер (і разом з ним рух кнопки), або знову його запустити.

1.2 Рядок стану *StatusStrip*

`StatusStrip` представляє рядок стану, який призначений для відображення поточної інформації про стан роботи програми.

При додаванні на форму `StatusStrip` автоматично розміщується в нижній частині вікна програми. Однак при необхідності зможемо його інакше позиціонувати за допомогою властивості `Dock`, яке може набувати таких значень:

- `Bottom`: Розміщення внизу (значення за замовчуванням).
- `Top`: Прикріплює рядок статусу до верхньої частини форми.
- `Fill`: Розтягує на всю форму.
- `Left`: Розміщення в лівій частині форми.
- `Right`: Розміщення в правій частині форми.
- `None`: Довільне положення.

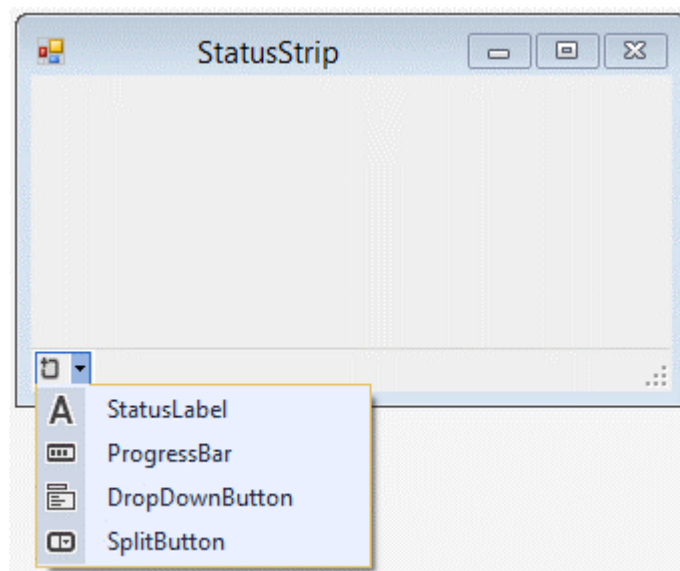


Рисунок 1.3 – Елементи `StatusStrip` у режимі дизайнера

`StatusStrip` може містити різні елементи. У режимі дизайнера можемо додати такі типи елементів (рис. 1.3):

- `StatusLabel`: мітка для виведення текстової інформації. Представляє об'єкт `ToolStripLabel`;
- `ProgressBar`: індикатор прогресу. Представляє об'єкт `ToolStripProgressBar`;
- `DropDownButton`: кнопка з списком, що випадає при натисканні. Представляє об'єкт `ToolStripDropDownButton`;

- SplitButton: ще одна кнопка, багато в чому аналогічна DropDownButton. Представляє об'єкт ToolStripSplitButton.

Можна звернутися до властивості Items компонента StatusStrip і у вікні додати та налаштувати всі елементи (рис. 1.4):

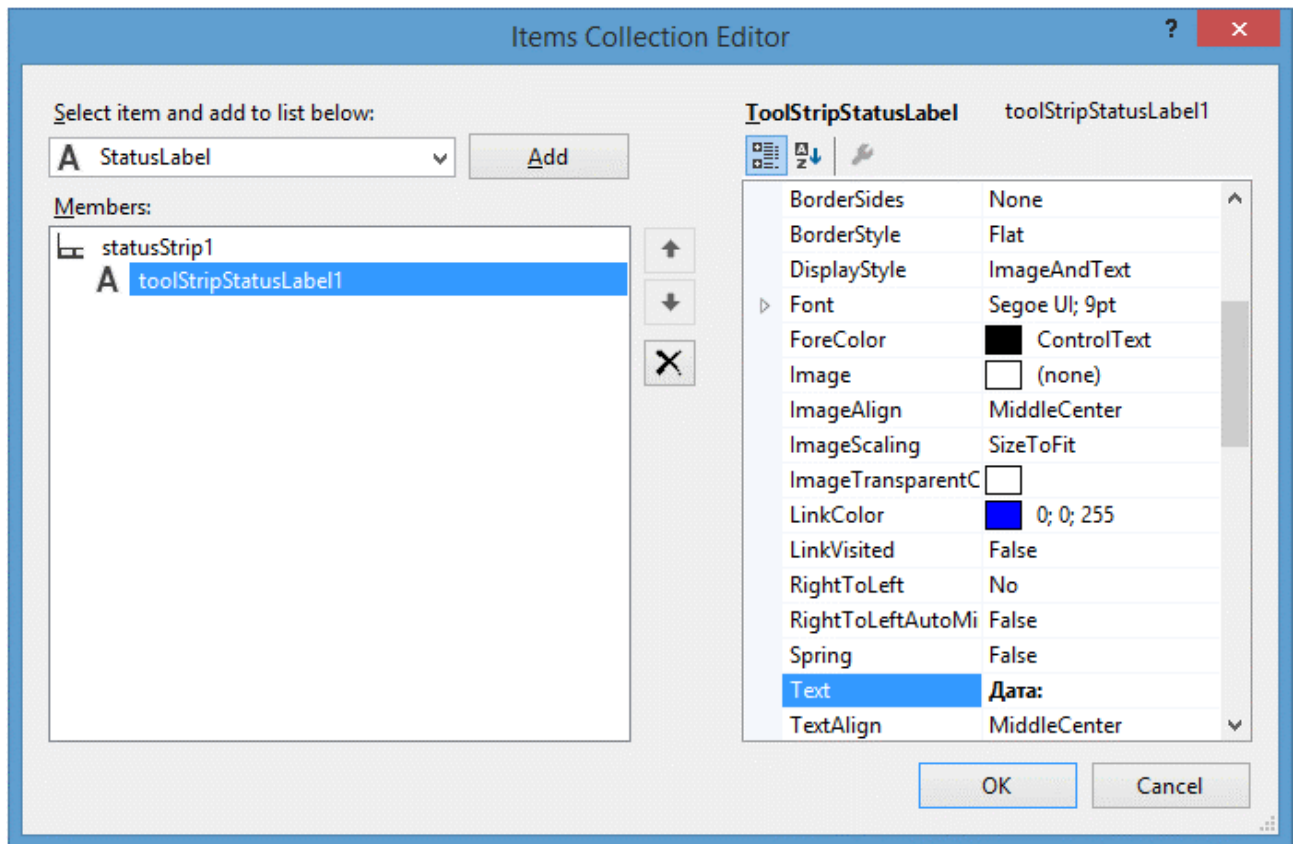


Рисунок 1.4 – Вікно властивостей Items компонента StatusStrip

Також ми можемо додати елементи програмно, як у коді форми, що наведений нижче:

```
public partial class Form1 : Form
{
    ToolStripLabel dateLabel;
    ToolStripLabel timeLabel;
    ToolStripLabel infoLabel;
    Timer timer;
    public Form1()
    {
        InitializeComponent();

        infoLabel = new ToolStripLabel();
        infoLabel.Text = "Текущие дата и время:";
        dateLabel = new ToolStripLabel();
        timeLabel = new ToolStripLabel();

        statusStrip1.Items.Add(infoLabel);
        statusStrip1.Items.Add(dateLabel);
        statusStrip1.Items.Add(timeLabel);
    }
}
```

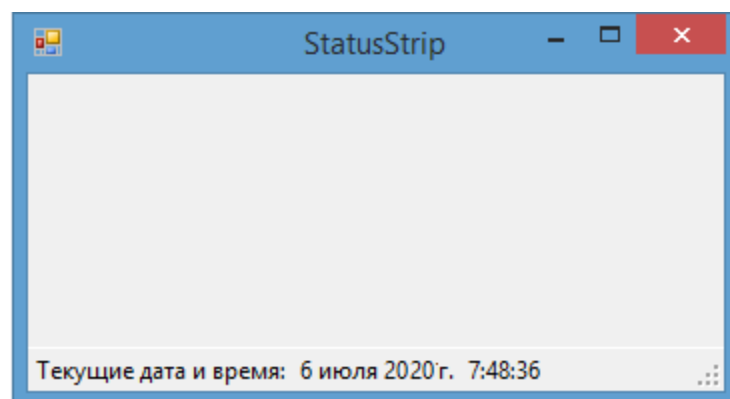
```

        timer = new Timer() { Interval = 1000 };
        timer.Tick += timer_Tick;
        timer.Start();
    }

    void timer_Tick(object sender, EventArgs e)
    {
        dateLabel.Text = DateTime.Now.ToLongDateString();
        timeLabel.Text = DateTime.Now.ToLongTimeString();
    }
}

```

У кодi створюються три мiтки на рядку стану i таймер. Пiсля створення форми таймер запускається, i спрацьовує його подiя Tick, в обробнику якого встановлюємо текст мiток.



2. Практичне завдання



У процесі виконання завдань лабораторної роботи необхідно формувати набори тестових даних для перевірки правильності виконання програмного коду. Створений код i результати перевірки його роботи потрібно помістити у звіт. Тестувати роботу програми рекомендується після додання чи зміни кожного оператора виведення.

Завдання 1. Створити проект для розв'язання задачі.

У формі будуть з'являтися випадкові літери. Якщо гравець вводить букви правильно, вони зникають - рівень зростає i букви з'являються все частіше. Якщо вся форма заповнена літерами гра закінчена.

1. Створюємо проект **winForm** з назвою **Type**. Для властивостей форми виставляємо наступні значення:

FormBorderStyle = Fixed3D

Size = 876; 174

Text = Hit the keys!

KeyPreview = true;

2. Додаємо на форму елемент **ListBox**, Для цього елемента виставляємо наступні значення:

Dock = Fill;

MultiColumn = True;

Font Size = 72

3. Додаємо на форму елемент **Timer**.

4. Додаємо на форму елемент **StatusStrip**, для якого встановлюємо властивість: SizingGrip = False.

5. Додаємо в елемент **StatusStrip** 4 мітки **StatusLabel** (рис. 2.1)

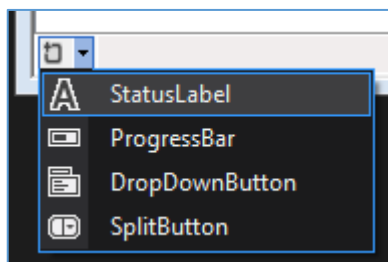


Рисунок 2.1 –Елементи StatusStrip

6. Для міток виставляємо такі властивості:

Мітка 1	Мітка 2	Мітка 3	Мітка 4
Name = correctLabel; Text = Correct: 0;	Name = missedLabel; Text = Missed: 0;	Name = totalLabel; Text = Total: 0;	Name = accuracyLabel; Text = Accuracy: 0%;

7. Додаємо в **StatusStrip**, ще один **StatusLabel** з властивостями:

Spring: True;

TextAlign: MiddleRight;

Text: Difficulty.

8. Додаємо в **StatusStrip** елемент **ProgressBar** (рис. 2.1), властивості Name присвоюємо значення difficultyProgressBar, а властивості Maximum = 701. Форма повинна мати вигляд, як наведено на рис. 2.2.

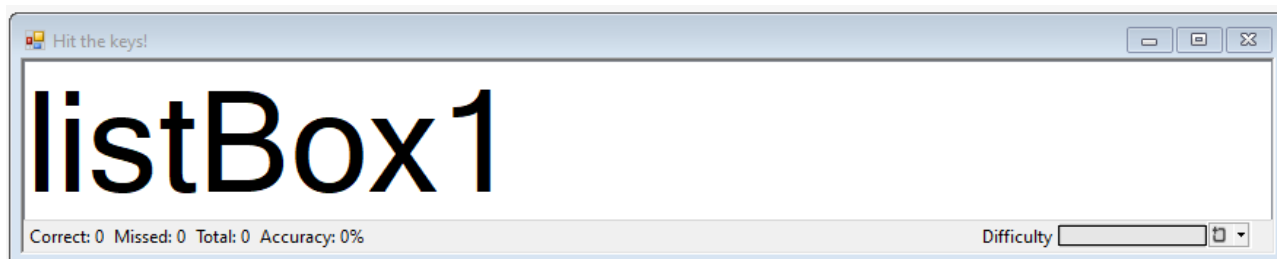


Рисунок 2.2 – Вигляд форми для реалізації завдання 1

9. Виділити елемент **Timer** та встановити властивість Interval: 800.

10. Створюємо обробник події **Tick** для елемента **Timer**. Додаємо наступний код

```

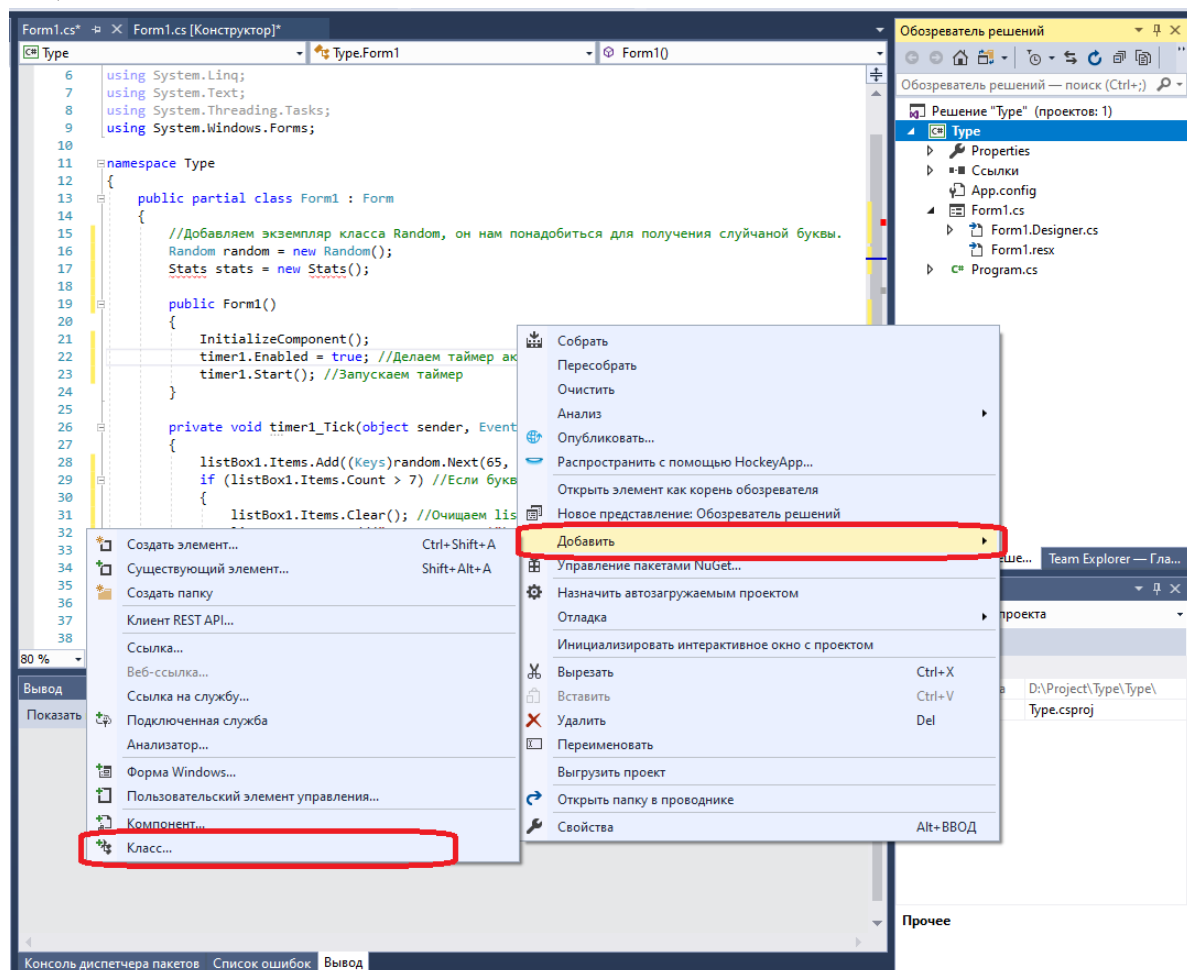
namespace Type
{
    public partial class Form1 : Form
    {
        //Додаємо екземпляр класу Random, який необхідний для отримання випадкової літери
        Random random = new Random();

        public Form1()
        {
            InitializeComponent();
            timer1.Enabled = true; //Робимо таймер активним
            timer1.Start(); //Запускаємо таймер
        }

        private void timer1_Tick(object sender, EventArgs e)
        {
            listBox1.Items.Add((Keys)random.Next(65, 90)); //Додаємо в listBox літеру
            if (listBox1.Items.Count > 7) //Якщо літер більше 7 гра завершена.
            {
                listBox1.Items.Clear(); //Очищаємо listBox
                listBox1.Items.Add("Гра завершена!");
                timer1.Stop(); //Зупиняємо таймер.
            }
        }
    }
}

```

11. Додаємо до проекту новий клас **Stats** для отримання статистики (рис. 2.3).



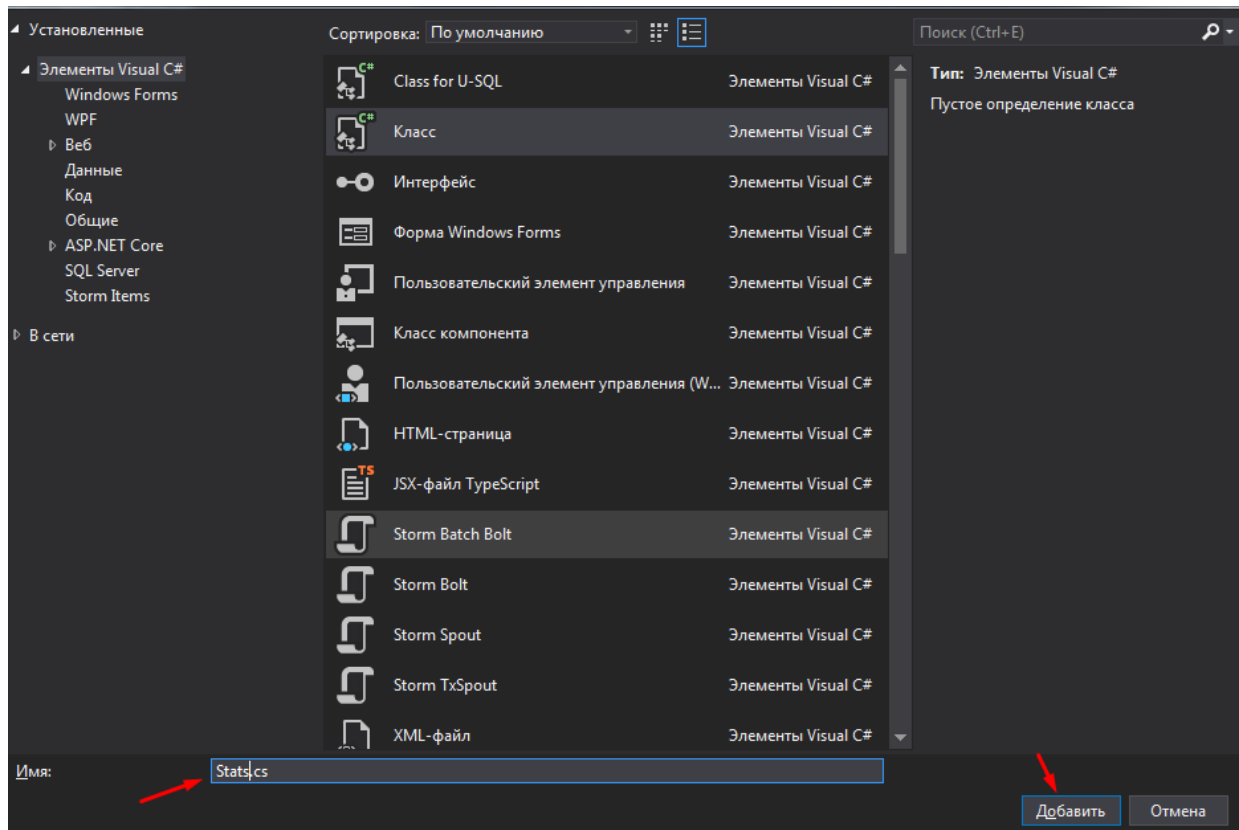


Рисунок 2.3 – Додаємо до проекту новий клас Stats

12. Додаємо в нього такий код:

```
namespace Type
{
    class Stats
    {
        public int Total = 0;
        public int Missed = 0;
        public int Correct = 0;
        public int Accuracy = 0;

        public void Update(bool correctKey)
        {
            Total++;
            if (!correctKey) Missed++;
            else Correct++;

            Accuracy = 100 * Correct / (Missed + Correct);
        }
    }
}
```

13. Повертаємось до класу **Form1.cs** і створюємо в ньому екземпляр класу Stats (рис. 2.4)


```

ссылка: 3
public partial class Form1 : Form
{
    //Добавляем экземпляр класса Random, он нам понадобится для получения случайной буквы.
    Random random = new Random();
    Stats stats = new Stats(); ←
    ссылка: 1
    public Form1()
    {
        InitializeComponent();
    }
}

```

Рисунок 2.4 – Фрагмент кода для створення екземпляр класу Stats

14. У конструкторі форми, вибираємо форму та переходимо в події. Створюємо для форми подію Key_Down (натискаємо 2 рази) (рис. 2.5).

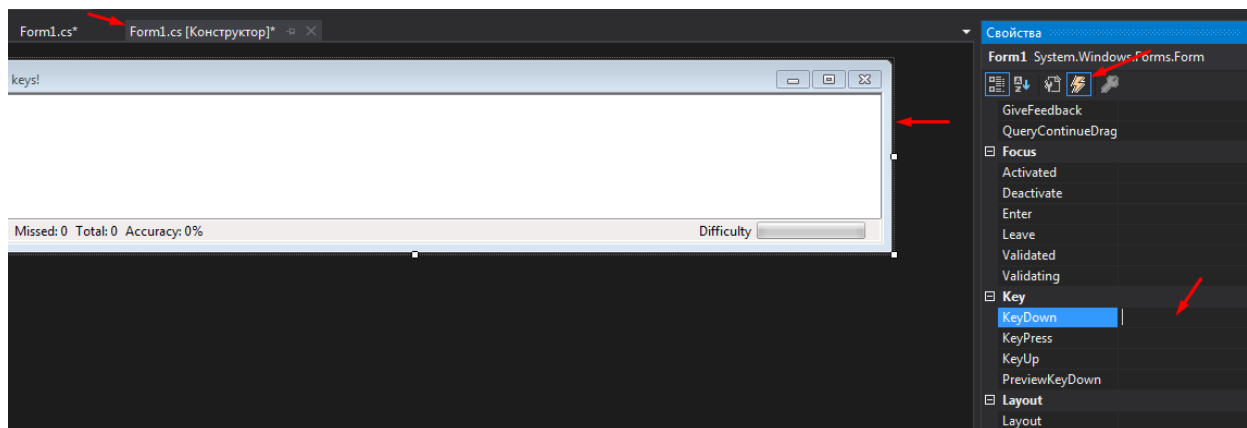


Рисунок 2.5 – Створення для форми події Key_Down

```

private void Form1_KeyDown(object sender, KeyEventArgs e)
{
    //Якщо користувач натискає клавіші, то літери видаляються,
    //а швидкість їх появи збільшується.
    if (listBox1.Items.Contains(e.KeyCode)) //Перевірка на правильність натиснення.
    {
        listBox1.Items.Remove(e.KeyCode); //видалення літери з listBox`a
        listBox1.Refresh();
        if (timer1.Interval > 400) timer1.Interval -= 10;
        if (timer1.Interval > 250) timer1.Interval -= 7;
        if (timer1.Interval > 100) timer1.Interval -= 2;

        difficultyProgressBar.Value = 800 - timer1.Interval;
        //Заповнення ProgressBar`a
        stats.Update(true); //Оновлюємо статистику.
    }
    else
    {
        stats.Update(false);
    }

    correctLabel.Text = "Correct: " + stats.Correct;
    missedLabel.Text = "Missed " + stats.Missed;
    totallabel.Text = "Total " + stats.Total;
    accuracyLabel.Text = "Accuracy: " + stats.Accuracy + "%";
}

```

3. Контрольні запитання

1. Призначення елемента Timer.
2. Як задати час спрацьовування таймера?
3. Як включити/виключити таймер?
4. Як налаштувати обробник події Tick елемента Timer?
5. Яке призначення елементу StatusStrip?
6. Яке призначення властивості Dock?
7. Які елементи можна додати в режимі дизайнера елементу StatusStrip?

Література

1. Евдокимов П. В. С# на примерах. СПб.: Наука и Техника, 2019. 320 с.
2. Маки А. Введение в .NET 4.0 и Visual Studio 2010 для профессионалов; пер. с англ. М. : ООО ИД "Вильямс", 2010. 416 с
3. С# 7.0. Справочник. Полное описание языка.: Пер. с англ. СПб.: ООО "Альфа-книга", 2018. 1024 с.
4. Троелсен, Эндрю, Джепикс, Филипп. Язык программирования С# 7 и платформы .NET и .NET Core. СПб. : ООО "Диалектика", 2018. 1328 с.
5. Офіційний сайт компанії Microsoft щодо технологій WPF та Windows Forms [Електронний ресурс]. – Режим доступу : <http://window-sclient.net>.
6. С#. Теорія та практика. URL: https://www.bestprog.net/uk/sitemap_ua/c-3
7. Сажин А. Справочник по языку программирования С#. URL: <https://brainoteka.com/blogs/c-spravochnik>.
8. С# Tutorial URL <https://www.theengineeringprojects.com>.
9. Уроки С#. URL: <https://itproger.com/course/csharp>.
10. Полное руководство по С# 8 и .NET Core. URL: <https://metanit.com/sharp/>