

Практическая работа №19_3

Тема: Разработка игрового приложения

Создание приложения WinForms для игры "Подбери пару"

Постановка задачи: Создать игру, в которой игрок должен подобрать пару скрытым значкам.

Требуемые задачи на выполнение:

- сохранять объекты, например значки, в объекте **List<T>**;
- использовать цикл **foreach** в C# для перебора элементов в списке;
- отслеживать состояние формы с помощью ссылочных переменных;
- создавать обработчик событий, который можно использовать для нескольких объектов;
- создавать таймер, который отсчитывает время и вызывает событие ровно один раз после запуска.

Все задания практической работы размещать в своей папке проектов в новой папке **Пр19_Фамилия**

Задание 1. Создание проекта игры "Подбери пару" Windows Forms

1. Запустите Visual Studio. Создайте новый проект - шаблон **Приложение Windows Forms (.NET Framework)** для C# под именем *pr19_3Фамилия*

Visual Studio создает решение для приложения. Решение является контейнером для всех проектов и файлов, необходимых приложению.

Задание 2. Создание макета для игры: создание для игры сетки "четыре на четыре".

1. Щелкните форму, чтобы выбрать конструктор Windows Forms. На этой вкладке для C# считывается файл **Form1.cs [Design]**. В окне **Свойства** задайте следующие значения свойств формы.

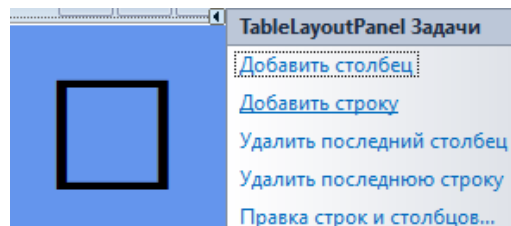
- Измените свойство **Text**
- Задайте размер формы: свойства **Size** значение **550, 550**

2. Перетащите элемент управления **TableLayoutPanel** из категории **Контейнеры** на панели элементов или дважды щелкните его. В окне **Свойства** задайте следующие свойства для панели.

- Задайте для свойства **BackColor** значение **CornflowerBlue** (в диалоговом окне **BackColor** выберите **Интернет - CornflowerBlue** в списке названий доступных цветов).
- Выберите для свойства **Dock** значение **Fill (Заливка)** из раскрывающегося списка, нажав большую кнопку, расположенную посередине. Этот параметр позволяет растянуть таблицу по всей форме.
- Для свойства **CellBorderStyle** установите значение **Inset**. Задание этого значения приведет к тому, что между ячейками поля появятся видимые границы.
- Нажмите треугольную кнопку в правом верхнем углу элемента управления **TableLayoutPanel** для отображения меню задач этой панели.

В меню задачи щелкните команду **Добавить строку** дважды, чтобы добавить еще две строки. Затем дважды щелкните пункт **Добавить столбец**, чтобы добавить еще два столбца.

- В меню задач выберите команду **Правка строк и столбцов**, чтобы открыть окно **Стили столбцов и строк**. Для каждого столбца выберите параметр **Процент**, а затем задайте для каждого столбца ширину **25%**.
- Затем в списке в верхней части окна выберите пункт **Строки** и задайте высоту каждой строки равной **25%**.
- Задав все параметры, нажмите кнопку **ОК**, чтобы сохранить изменения.



Элемент управления **TableLayoutPanel** теперь представляет собой сетку "четыре на четыре" с 16 квадратными ячейками одинакового размера. Эти строки и столбцы задают места, в которых позже появятся значки.

Задание 3. Добавление и форматирование меток для отображения

Создание и форматирование меток, которые будут отображаться во время игры:

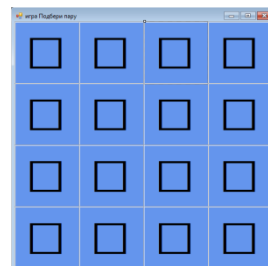
1. Убедитесь, что **TableLayoutPanel** выбран в редакторе формы (элемент **tableLayoutPanel1** должен быть отображен в верхней части окна **Свойства**).

- На панели элементов выберите **Label** и добавьте элемент управления в верхнюю левую ячейку *TableLayoutPanel*. Задайте для *Label* следующие свойства.
 - Задайте для свойства **BackColor** метки значение **CornflowerBlue**.
 - Задайте свойству **AutoSize** значение **False**.
 - Задайте для свойства **Dock** значение **Fill**.
 - Задайте для свойства **TextAlign** значение **MiddleCenter**, чтобы значок отображался в середине ячейки.
 - Выберите свойство **Font**. Задайте для параметра **Font** значение **Webdings**, для параметра **Font Style** — значение **Bold**, а для параметра **Size** — значение **48**.
 - Установите свойство **Text** равным букве **с**.

Теперь в левой верхней ячейке *TableLayoutPanel* располагается черный квадрат на синем фоне, который выравнивается по центру.

- Выберите элемент управления *Label* и скопируйте его в следующую ячейку *TableLayoutPanel*.

Во второй ячейке элемента управления *TableLayoutPanel* появится копия первого элемента управления *Label*. Вставьте его снова, и в третьей ячейке появится еще один элемент управления *Label*. Продолжайте вставлять элементы управления *Label*, пока все ячейки не будут заполнены.



Этот шаг завершает создание макета вашей формы.

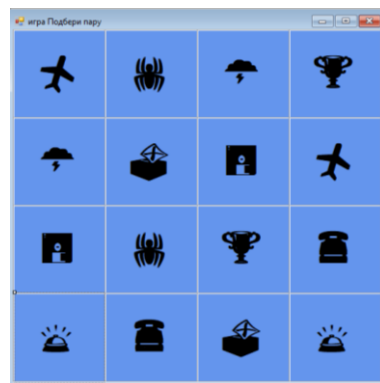
Примечание

Webdings — шрифт значков, который поставляется с операционной системой Windows. В игре "Подбери пару" игроку нужно подобрать пару для значков. Этот шрифт отображает значки, для которых нужно подобрать пары.

Вместо **с** попробуйте использовать в свойстве **Text** разные буквы. Восклицательный знак соответствует науку, прописная буква **N** — глазу, а запятая — перцу чили.

Добавление значков в приложение WinForms для игры "Подбери пару"

В игре "Подбери пару" игрок выбирает квадрат, чтобы просмотреть значок, после чего ему нужно выбрать другой квадрат. Если значки совпадают, они остаются видимыми. Если нет, согласно правилам игры оба значка будут скрыты.



Задание 4. Добавление случайного объекта и списка значков

Вам понадобится использовать два оператора **new**, создающие два объекта. Первый — это объект **Random**, который случайным образом выбирает ячейки в элементе управления *TableLayoutPanel*. Второй объект является объектом **List<T>**. В нем хранятся случайно выбранные символы.

Создайте набор парных символов для игры. Каждый символ добавляется в две случайные ячейки в **TableLayoutPanel** в форме.

- В имеющийся код формы в класс **Form1** (сразу после объявления класса `public partial class Form1 : Form`) добавьте следующий фрагмент (символы перечислить в соответствии с теми, что использованы у вас в метках).

```
Random random = new Random();
// Каждая из этих букв представляет собой интересную иконку
// в шрифте Webdings, и каждый значок появляется дважды в этом списке
List<string> icons = new List<string>()
{
    "!", "!", "с", "с", "Ь", "Ь", "%", "%",
    "?", "?", "Н", "Н", "Й", "Й", "ж", "ж"
};
```

Объекты *List* можно использовать для отслеживания элементов различных типов. Список может содержать числа, значения *true* или *false*, текст и другие объекты. В игре "Подбери пару" объект *list* содержит 16 строк, по одной для каждой ячейки на панели *TableLayoutPanel*. Каждая строка представляет собой одну букву, соответствующую значкам в метках. Эти символы отображаются в шрифте *Webdings* в виде автобуса, велосипеда и т. д.

Назначение каждому элементу управления Label случайного значка

При каждом запуске программы значки назначаются элементам управления Label в форме случайным образом с помощью метода **AssignIconsToSquares()**. Этот код использует ключевое слово **foreach** на C#.

2. Добавьте метод AssignIconsToSquares().

```
/// <summary>
/// Назначаем каждую иконку из списка иконок случайному квадрату
/// </summary>
private void AssignIconsToSquares()
{
    // TableLayoutPanel имеет 16 меток, а в списке значков 16 значков,
    // чтобы иконка вытягивалась случайным образом из списка
    // и добавляем к каждой метке
    foreach (Control control in tableLayoutPanel1.Controls)
    {
        Label iconLabel = control as Label; //преобразует переменную control в метку с именем iconLabel
        if (iconLabel != null) //Если преобразование выполняется, то
        {
            int randomNumber = random.Next(icons.Count); //создается переменная, содержащая случайное число,
            //соответствующее одному из элементов списка значков
            iconLabel.Text = icons[randomNumber]; //один из элементов списка значков присваивается свойству Text этой метки
            //строка закомментирована специально, чтобы проверить оставшуюся часть кода, прежде чем продолжить работу. Вы добавите ее позже в этой процедуре.
            //iconLabel.ForeColor = iconLabel.BackColor; //скрывает значки

            icons.RemoveAt(randomNumber); //удаляет из списка значков, добавленный в форму
        }
    }
}
```

3. Добавьте вызов метода AssignIconsToSquares() в конструктор Form1. Этот метод заполняет игровую доску значками. Конструкторы вызываются при создании объекта.

```
public Form1()
{
    InitializeComponent();

    AssignIconsToSquares();
}
```

4. Сохраните и выполните программу. Должна отобразиться форма со случайными значками, которые назначены каждой метке.

Примечание: Если значки Webdings не отображаются в форме правильно, установите для свойства **UseCompatibleTextRendering** меток в форме значение **True**.

5. Закройте программу, а затем снова запустите ее. Каждой метке назначены разные значки. Значки видимы, поскольку они не были скрыты. Чтобы скрыть их от игрока, можно задать для свойства **ForeColor** каждого элемента управления Label тот же цвет, что и у свойства **BackColor**.

6. Остановите выполнение программы; Удалите метки комментариев с соответствующей строки кода в цикле.

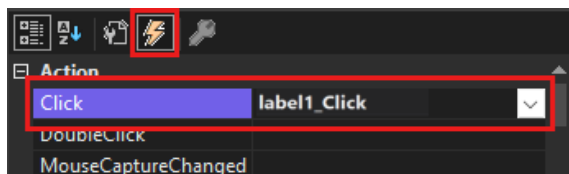
```
iconLabel.ForeColor = iconLabel.BackColor;
```

Задание 5. Добавление обработчиков событий в метки

В этой игре "Подбери пару" игрок открывает один скрытый значок, а затем второй. Если значки совпадают, они остаются видимыми. Если нет, оба значка снова скрываются.

Чтобы заставить вашу игру работать таким образом, добавьте обработчик события **Click**, который изменяет цвет выбранной метки в соответствии с фоном.

1. Выберите первый элемент Label, чтобы выделить его. Затем, удерживая нажатой клавишу **Ctrl**, выберите каждую из оставшихся меток. Убедитесь, что выбраны все метки.
2. В окне **Свойства** нажмите кнопку **События**, которая представляет собой молнию. Для события **Click** выберите в поле **label1_Click**.
3. Нажмите клавишу **ВВОД**. Интегрированная среда разработки добавляет в код обработчик события Click с именем **label1_Click()**. Поскольку вы выбрали все метки, обработчик будет привязан к каждой из этих меток.
4. Добавьте остальную часть кода.



```

/// <summary>
/// Событие Click каждой метки обрабатывается этим обработчиком событий
/// </summary>
/// <param name="sender"> Метка, на которую нажали</param>
/// <param name="e"></param>
private void label1_Click(object sender, EventArgs e)
{
    Label clickedLabel = sender as Label;

    if (clickedLabel != null)
    {
        // Если метка, по которой щелкнули, черная, значит, игрок щелкнул
        // уже открытая иконка – игнорировать щелчок
        if (clickedLabel.ForeColor == Color.Black)
            return;

        clickedLabel.ForeColor = Color.Black;
    }
}

```

5. Запустить программу на проверку. Выберите любую из ячеек в форме. После этого должен отобразиться один из значков.

Добавление ссылочных переменных и элемента управления timer в приложение WinForms для игры "Подбери пару"

Программе игры "Подбери пару" нужно отслеживать, какие элементы управления *Label* выбирает игрок. Поскольку она должна отобразить значок после того, как игрок выберет первую метку. После выбора второй метки программа должна показать оба значка на короткое время, а затем снова скрыть их.

Ваша программа отслеживает, какую метку вы выбрали первой и второй, используя ссылочные переменные. Таймер скрывает значки и определяет, как долго нужно показывать значки.

- Добавление ссылок на элементы управления *Label*.
- Добавление таймера.

Задание 6. Добавление ссылок на элементы управления *Label*

Добавьте в код две ссылочные переменные, которые будут отслеживать (или ссылаться на) объекты *Label*.

1. Добавьте ссылки на метки в свою форму, используя следующий код.

```

public partial class Form1 : Form
{
    Label firstClicked = null;    //первый элемент управления Label, что игрок нажимает,
                                //это будет ноль, если игрок еще не нажал на метку
    Label secondClicked = null;  //указывает на второй элемент управления Label,
                                //что игрок нажимает
}

```

Эти операторы не приводят к отображению элементов управления *Label* на форме, поскольку не указывается ключевое слово *new*. Когда программа запускается, и *firstClicked*, и *secondClicked* имеют значение *null* для C#.

2. Измените свой обработчик событий *Click* для использования новой ссылочной переменной *firstClicked*. Удалите последнюю инструкцию в методе обработчика событий *label1_Click()* (*clickedLabel.ForeColor = Color.Black;*) и замените ее выражением *if*, как показано ниже.


```

private void label1_Click(object sender, EventArgs e)
{
    Label clickedLabel = sender as Label;

    if (clickedLabel != null)
    {
        // Если метка, по которой щелкнули, черная, значит, игрок щелкнул
        // уже открытая иконка – игнорировать щелчок
        if (clickedLabel.ForeColor == Color.Black)
            return;

        //clickedLabel.ForeColor = Color.Black;
        if (firstClicked == null)
        {
            firstClicked = clickedLabel;
            firstClicked.ForeColor = Color.Black;
            return;
        }
    }
}

```

3. Сохраните и выполните программу. Выберите одну из меток и появится ее значок. Выберите следующую метку и обратите внимание, что ничего не происходит.

При этом черным становится только первый выбранный значок. Другие значки останутся невидимыми.

Программа уже отслеживает первую метку, которую выбрал игрок. Поэтому ссылка firstClicked не равна null на C#. Когда выражение if обнаруживает, что firstClicked не равно null, оно выполняет последующие инструкции.

Задание 7. Добавление таймера

Игра "Подбери пару" использует для управления приложением **Timer**. Таймер ожидает, а затем вызывает событие, называемое **тактом**. Он может запускать действие или регулярно повторять его.

При использовании в программе таймер позволяет игроку выбрать два значка. Если эти значки не совпадают, программа скроет их по истечении короткого периода времени.

1. Выберите вкладку **Панель элементов**, в категории **Компоненты** дважды щелкните или перетащите компонент **Таймер** в свою форму. В области под формой появится значок таймера с именем **timer1**.
2. Щелкните значок **Timer1**, чтобы выбрать таймер. В окне **Свойства** выберите кнопку **Свойства**, чтобы просмотреть соответствующие параметры.
3. Задайте для свойства **Interval** значение **750**, то есть 750 миллисекунд.
*Свойство **Interval** задает время ожидания между тактами таймера во время активации события **Tick**. Ваша программа вызывает метод **Start()** для запуска таймера только после того, как игрок выберет вторую метку.*
4. Выберите значок элемента управления **timer**, а затем нажмите **Ввод** или дважды щелкните таймер. Напишите код обработчика события Tick:

```

private void timer1_Tick(object sender, EventArgs e)
{
    timer1.Stop();

    firstClicked.ForeColor = firstClicked.BackColor;
    secondClicked.ForeColor = secondClicked.BackColor;

    firstClicked = null;
    secondClicked = null;
}

```

5. Перейдите в редактор кода и добавьте в начало и конец метода обработчика событий **label1_Click()** код, выполняющий следующие действия:
 - в начале метода проверяет, запущен ли таймер, обращаясь к значению необходимого свойства. Если игрок выбирает первый и второй элемент управления *Label* и таймер запускается, выбор третьего элемента управления *Label* ни к чему не приведет.

- Код в конце метода задает ссылочную переменную **secondClicked** для отслеживания второго элемента управления *Label*. И затем присваивает значку метки черный цвет, чтобы сделать его видимым.
- Затем после переменной **secondClicked** опишите запуск таймера в однократном режиме (то есть ожидает 750 миллисекунд и после этого вызывает одно событие *Tick*).

Обработчик события Tick таймера скрывает два значка и сбрасывает ссылочные переменные firstClicked и secondClicked. Теперь форма готова к тому, чтобы игрок выбрал другую пару значков.

6. Сохраните и выполните программу. Нажмите квадрат, и значок должен стать видимым. Затем нажмите другой квадрат. Значок должен появиться на короткое время, а затем оба значка должны исчезнуть.

Теперь ваша программа может отслеживать выбор первого и второго значков. А также использует таймер для приостановки перед исчезновением значков.

Вывод сообщения в приложении WinForms для игры "Подбери пару"

Далее нужно доработать игру "Подбери пару" так, чтобы совпадающие пары были видны, а при победе игрока выводилось сообщение с поздравлением.

Задание 8. Отмена исчезновения пар значков

Если игрок подобрал пару, игра должна перезагрузиться, чтобы больше не отслеживать метки, использующие ссылочные переменные firstClicked и secondClicked. Однако она не должна сбрасывать цвета для двух совпадающих меток. Эти метки должны и дальше отображаться.

1. Добавьте следующий оператор **if** в метод обработчика событий **label_Click()**. Разместите его ближе к концу кода непосредственно над оператором, запускающим таймер.

```
if (firstClicked.Text == secondClicked.Text)
{
    firstClicked = null;
    secondClicked = null;
    return;
}
```

Первый оператор if проверяет, совпадает ли значок в первой метке, которую выбрал игрок, со значком во второй метке. Если значки одинаковы, программа выполняет три оператора. Первые два оператора сбрасывают ссылочные переменные firstClicked и secondClicked. После этого они перестают отслеживать метки. Третий оператор — оператор return, который пропускает оставшиеся в методе операторы, позволяя не выполнять их.

2. Запустите программу, а затем начните выбирать квадраты в форме.

Если вы выбрали пару, которая не совпадает, произойдет событие таймера *Tick* и оба значка исчезнут.

Если выбрать совпадающую пару, будет выполнен оператор **if**. Оператор **return** заставляет метод пропустить код, который запускает таймер. При этом значки остаются видимыми.

Задание 9. Проверка того, выиграл ли игрок

Вы создали игру, которая должна завершиться, как только игрок победит. Добавьте метод для проверки того, выиграл ли игрок.

1. Добавьте метод **CheckForWinner()** в конец кода, под обработчиком событий **timer1_Tick()**:

```
private void CheckForWinner()
{
    // Проходим по всем меткам в TableLayoutPanel,
    // проверка каждого, чтобы увидеть, соответствует ли его значок
    foreach (Control control in tableLayoutPanel1.Controls)
    {
        Label iconLabel = control as Label;

        if (iconLabel != null)
        {
            if (iconLabel.ForeColor == iconLabel.BackColor)
                return;
        }
    }

    // Если цикл не вернулся, он не нашел любые неподходящие иконки
    // Это означает, что пользователь выиграл. Показать сообщение и закрыть форму
    MessageBox.Show("Вы сопоставили все значки!", "Поздравляем!!!");
    Close();
}
}
```

В этом методе используется еще один цикл `foreach` для C#, чтобы пройти по каждой метке в `TableLayoutPanel`. Он проверяет цвет значка каждой метки, чтобы убедиться, что он совпадает с фоном. Если цвета совпадают, значок остается невидимым, а значит, игрок не подобрал пару оставшимся значкам.

В этом случае программа использует оператор **`return`**, чтобы пропустить оставшуюся часть метода. Если цикл прошел через все метки без выполнения оператора **`return`**, значит, всем значкам в форме была подобрана пара. Программа отображает окно **`MessageBox`** с поздравлением победителя, а затем вызывает метод **`Close()`** для завершения игры. После этого обработчик событий `Click` метки вызывает новый метод **`CheckForWinner()`**.

Найдите строку, где задается цвет второму значку, который вы выбрали, и вызовите метод **`CheckForWinner()`** сразу после этой строки.

`CheckForWinner();`

Убедитесь, что программа проверяет наличие победителя сразу после отображения второго значка, который выбирает игрок. Сохраните и выполните программу. Сыграйте в игру и подберите пару всем значкам. В случае победы программа должна выводить сообщение с поздравлением.

Если выбрать **ОК**, игра "Подбери пару" закроется.

Самостоятельные задания 10:

Попробуйте сделать игру более сложной. Ниже приведены некоторые варианты.

- Замените значки и цвета на другие.
Попробуйте проверить свойство **`ForeColor`** метки.
- Добавьте таймер игры, который отслеживает время, необходимое игроку для победы. Вы можете добавить метку, чтобы отобразить время, затраченное на форму. Разместите ее над **`TableLayoutPanel`**. Добавьте в форму еще один таймер, чтобы иметь возможность отслеживать время. Следующий код служит для запуска таймера, когда игрок начинает игру, и остановки таймера после сопоставления последних двух значков.
- Сделайте игру интереснее, скрывая первый значок, если игрок медлителен.
- Добавьте звуки, которые будут воспроизводиться при нахождении игроком пары, отображении двух несовпадающих значков и повторном сокрытии значков программой. Для воспроизведения звуков можно использовать пространство имен **`System.Media`**.
- Сделайте игру труднее, увеличив игровое поле.
Необходимо сделать больше, чем просто добавить строки и столбцы в `TableLayoutPanel`. Вы должны также учитывать количество создаваемых значков.

