

Тема: Разработка игрового приложения

Задачи:

- ### Материально-техническое обеспечение:

Время на выполнение работы: 2 часа.

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

1. Конспект занятия.

Перечень справочной литературы:

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:

Элемент управления Windows Forms **PictureBox** используется для вывода изображений в формате точечных рисунков, GIF, JPEG, метафайлов и значков. Отображаемое изображение определяется свойством **Image**, которое можно задать во время выполнения или во время разработки. Свойство **SizeMode** определяет, каким образом изображение и элемент управления соответствуют друг другу.

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:

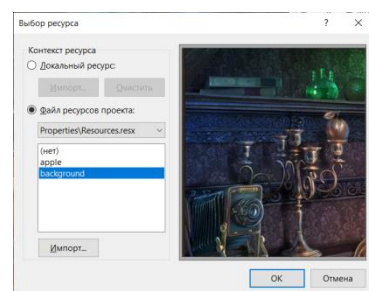
Все проекты практической работы размещать в своей сетевой в новой папке **Пр18 ФИО**

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

Задание 1. Создание простой игры «Найди предметы»

Пояснение задачи: Создать фрагмент простой игры «Найди предметы» с использованием элемента PictureBox. Суть игры, думаю, всем Вам знакома

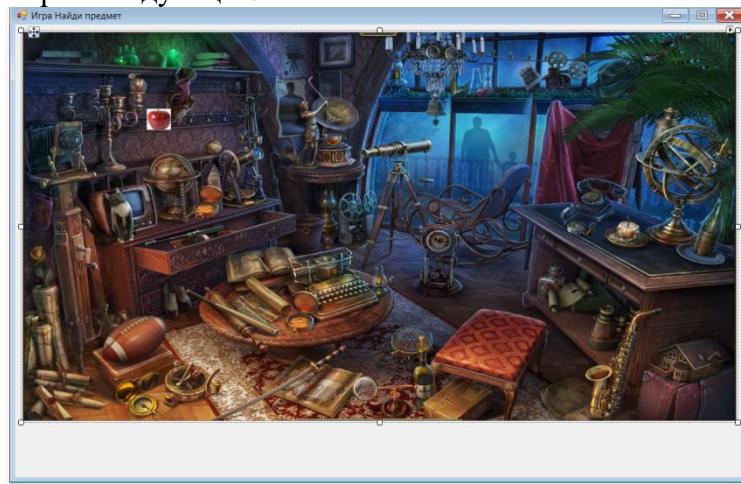
- 1) Создайте новый проект **pr18_1_Фамилия** типа Windows Forms.
- 2) Измените некоторые свойства формы:
 - ✓ название
 - ✓ задайте размер **1200*780**
 - ✓ отключим возможность изменения размеров **формы:**
FormBorderStyle - значение **FixedSingle, MaximizeBox – False**
- 3) Разместим на форме элемент **Panel**, так, чтобы он занял всю поверхность формы, не доходя до нижнего края формы на 150 пикселей. Задайте свойства:
 - ✓ **BackGroundImage** значение **background.jpg** (изображение добавить в качестве ресурса проекта)



✓ **BackgroundImageLayout** как **Stretch**, чтобы рисунок растягивался под размер **Panel**

- 4) Разместите на **Panel** элемент **PictureBox** размером примерно **40 на 35**. Установите рисунок в объекте (свойство **Image**), указав путь к файлу **apple.png** (изображение также добавить в качестве ресурса проекта) и установите свойство **SizeMode** – **StretchImage**.

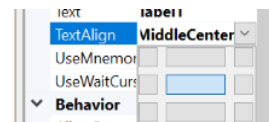
В итоге получите примерно следующее:



Как вы видите, вокруг картинке яблока остался серый выделяющийся фон. Далее, нужно будет сделать его прозрачным с помощью программного кода.

*Примечание: для использования возможности прозрачного фона на картинке, необходимо чтобы графический файл имел формат **PNG***

- 5) Задайте изменение курсора при наведении на рисунок яблока, чтобы было понятно, что пользователь навёл указатель на необходимую область (свойство **Cursor** элемента **PictureBox**) – значение **Hand**
- 6) Используя контекстное меню или сочетание **Ctrl+C** → **Ctrl+V**, скопируйте 5 раз рисунок яблока и расставьте его в различные неприметные места на фоновом рисунке. При необходимости, измените размер **PictureBox** с рисунком яблока.
- 7) Разместите в нижней части формы, не занятой элементом **Panel**, объект **Label**. Настройте свойство выравнивания надписи **TextAlign** на **MiddleCenter**:



Данный объект необходим для вывода некоторых результатов игры.

- 8) Далее приступаем к программированию:

- ✓ Для того, чтобы можно было использовать прозрачный фон для картинки яблока (**pictureBox**), добавьте строку в класс **Form1()** (поместите ее обязательно после инициализации компонентов формы)

```
public partial class Form1 : Form
{
    public Form1()
    {
        InitializeComponent();

        this.SetStyle(ControlStyles.SupportsTransparentBackColor, true);
    }
}
```

- ✓ Реализуйте обработчик события **Load** и объявите глобальную переменную для подсчёта искомых объектов:

```

int itemCount = 5;
private void Form1_Load(object sender, EventArgs e)
{
    foreach (PictureBox pictureBox in panel1.Controls)
    {
        pictureBox.BackColor = Color.Transparent;
        label1.Text = "Вам необходимо найти предметы: " + itemCount.ToString() + " яблок!";
    }
}

```

- ✓ Реализуйте обработчик события щелчка мыши по объекту *pictureBox1*:

```

private void pictureBox1_Click(object sender, EventArgs e)
{
    pictureBox1.Visible = false;
    itemCount--;
    if (itemCount <= 0) MessageBox.Show("Вы нашли все предметы", "Поздравляем!");
    else label1.Text = "Вам осталось найти " + itemCount.ToString() + " предмета";
}

```

- ✓ Аналогично запрограммируйте обработчики событий на щелчок мышью для остальных элементов *pictureBox*.
- ✓ Протестируйте приложение. При необходимости измените свойства компонентов и код.

Задание 2. Для разработанной в задании 1 игры добавьте второй уровень - поиск произвольного предмета (не менее 7ми предметов, разные предметы подобрать самостоятельно). На данном уровне пользователю должно выводиться сообщения: Какой предмет ему необходимо найти и Сколько предметов осталось найти.