

Лабораторная работа №34

Разработка интерфейса приложения

1 Цель работы

- 1.1 Научиться программно создавать и настраивать элементы управления в приложениях на C#;
- 1.2 Научиться использовать таймер в приложениях на C#.

2 Литература

2.1 Фленов М.Е. Библия C#. – Санкт-Петербург : БХВ-Петербург, 2016. – Режим доступа: <https://ibooks.ru/reading.php?productid=353561>, только для зарегистрированных пользователей. – Загл. с экрана. – гл.5, гл.13.

3 Подготовка к работе

- 3.1 Повторить теоретический материал (см. п.2).
- 3.2 Изучить описание лабораторной работы.

4 Основное оборудование

- 4.1 Персональный компьютер.

5 Задание

5.1 Отображение времени на форме

5.1.1 Создать новую форму и реализовать переход к ней с главной формы.

5.1.2 Добавить на форму из панели элементов Timer, две кнопки с подписями «Старт» и «Пауза» соответственно и метку для вывода времени.

5.1.3 В обработчиках нажатий на кнопки «Старт» и «Пауза» вызвать методы запуска и остановки таймера соответственно.

5.1.4 В обработчике события Tick реализовать вывод времени в формате мм:сс (минуты:секунды).

5.2 Анимация движения персонажа

5.2.1 Создать новую форму и реализовать переход к ней с главной формы.

5.2.2 Добавить на форму из панели элементов Timer, ImageList и PictureBox. Изменить фон формы.

Добавить у ImageList в коллекцию Images набор изображений для анимации персонажа (не менее 8 кадров). Набор изображений можно скачать из ресурсов.

Реализовать анимацию персонажа при запуске формы, используя смену изображений в PictureBox по таймеру, загружая их из ImageList.

```
pictureBox.Image = imageList.Images[i];
```

5.2.3 Для корректного сглаживания пикселей реализовать собственный класс-наследник PictureBox, в котором определить:

```
public class MyPictureBox : PictureBox
{
    protected override void OnPaint(PaintEventArgs pe)
    {
        pe.Graphics.SmoothingMode =
```

```

System.Drawing.Drawing2D.SmoothingMode.HighSpeed;
    pe.Graphics.CompositingQuality                                     =
System.Drawing.Drawing2D.CompositingQuality.HighSpeed;
    pe.Graphics.InterpolationMode                                     =
System.Drawing.Drawing2D.InterpolationMode.NearestNeighbor;
    base.OnPaint(pe);
}
}

```

5.2.4 Использовать для отображения изображения созданный класс.

5.3 Программная генерация клеточного игрового поля на основе данных из файла

5.3.1 Создать новую форму и реализовать переход к ней с главной формы.

5.3.2 Предоставить пользователю возможность выбора текстового файла карты с помощью диалогового окна при нажатии на кнопку на форме или на пункт меню.

5.3.3 После выбора файла требуется найти ширину (количество описанных клеток в одной строке файла) и высоту (количество строк в файле) игрового поля и реализовать формирование игрового поля путем программного создания и отображения его клеток.

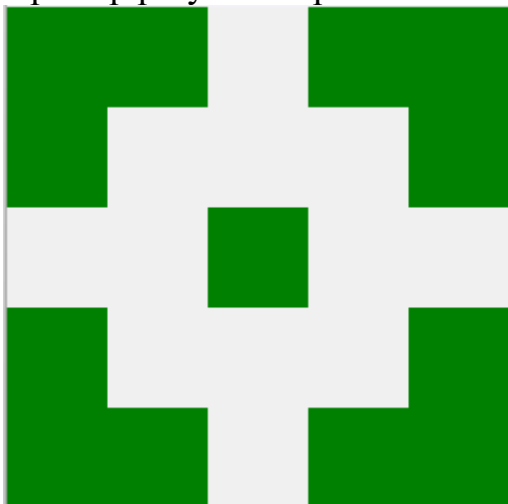
Пример содержимого файла игрового поля:

```

## ##
#  #
   #
#  #
## ##

```

Пример результата работы:



Для рисования карты программно располагайте панели определенного цвета на форме (Location, Size, BackColor)

6 Порядок выполнения работы

6.1 Используя Microsoft Visual Studio, создать проект C# и выполнить

задания из п.5.

6.2 Ответить на контрольные вопросы.

7 Содержание отчета

7.1 Титульный лист

7.2 Цель работы

7.3 Ответы на контрольные вопросы

7.4 Вывод

8 Контрольные вопросы

8.1 Какие основные свойства и события используются у таймера в приложении C#?

8.2 Как запустить и остановить таймер в приложении C#?

8.3 Что такое ImageList и для чего он используется в приложении C#?

8.4 Какие режимы изменения размеров изображения (SizeMode) поддерживаются у PictureBox и как изменяет изображение каждый из режимов?

8.5 Как программно создать и добавить элементы на форму в приложении C#?