

Лабораторная работа № 6.

Тема: Windows Forms. Графика. Компоненты для работы с графикой.

Цель: закрепить теоретические знания и применить их на практике.

Задание:

Вариант №1.

Задание 1.

Задание 1. Рисование с помощью Graphics + PictureBox.

Создайте форму с элементом PictureBox. Реализуйте кнопку "Нарисовать", которая при нажатии рисует на PictureBox прямоугольник, круг и линию с использованием класса Graphics. Добавьте возможность выбора цвета через ComboBox (например: красный, зеленый, синий). Цвет выбирается для всех фигур.

Подсказки:

Используйте метод pictureBox.CreateGraphics() для получения объекта Graphics, который позволяет рисовать на PictureBox.

Для рисования фигур используйте методы:

DrawRectangle(Pen, x, y, width, height) — для прямоугольника.

FillEllipse(Brush, x, y, width, height) — для круга.

DrawLine(Pen, startX, startY, endX, endY) — для линии.

Цвет можно задать через Color и передать его в Pen или Brush. Например:

```
Pen pen = new Pen(selectedColor, thickness);
```

```
Brush brush = new SolidBrush(selectedColor);
```

Очистите PictureBox перед рисованием с помощью:

```
graphics.Clear(Color.White); // Белый фон
```

Задание 2. Graphics + TrackBar + PictureBox.

Разработайте форму с PictureBox и TrackBar. Используйте TrackBar для изменения толщины линии, которую вы рисуете на PictureBox. При перемещении TrackBar обновляйте изображение, чтобы линия перерисовывалась с новой толщиной. Добавьте кнопку "Очистить", чтобы очистить PictureBox.

Подсказки:

Толщину линии можно задать через свойство Width объекта Pen. Например:

```
int thickness = trackBar.Value;
```

```
Pen pen = new Pen(Color.Black, thickness);
```

Используйте событие ValueChanged у TrackBar для обновления толщины линии:

```
private void trackBar_ValueChanged(object sender, EventArgs e)
```

```
{
```

```
    // Перерисовать линию с новой толщиной
```

```
}
```

Для очистки PictureBox используйте метод Clear() объекта Graphics:

```
pictureBox.CreateGraphics().Clear(Color.White);
```

Задание 3. Graphics + RadioButton + PictureBox.

Создайте форму с PictureBox и группой радиокнопок (RadioButton) для выбора фигуры (прямоугольник, круг, треугольник). При выборе радиокнопки и нажатии кнопки "Нарисовать" соответствующая фигура должна отображаться на PictureBox. Добавьте возможность выбора цвета через ColorDialog.

Подсказки:

Чтобы нарисовать треугольник, используйте метод DrawPolygon(Pen, points), где points — массив точек (Point[]):

```
Point[] trianglePoints = { new Point(50, 50), new Point(100, 150), new Point(0, 150) };
```

```
graphics.DrawPolygon(pen, trianglePoints);
```

Для выбора цвета используйте диалоговое окно ColorDialog:

```
ColorDialog colorDialog = new ColorDialog();  
if (colorDialog.ShowDialog() == DialogResult.OK)  
{  
    selectedColor = colorDialog.Color;  
}
```

Проверяйте, какая радиокнопка выбрана, с помощью свойства Checked:

```
if (radioButtonRectangle.Checked)  
{  
    graphics.DrawRectangle(pen, x, y, width, height);  
}
```

Задание 4. Graphics + Timer + PictureBox.

Разработайте анимацию движения круга по PictureBox. Используйте Timer для обновления координат круга каждые 50 миллисекунд. Добавьте кнопки "Старт" и "Стоп" для управления анимацией. Убедитесь, что круг отскакивает от краев PictureBox.

Подсказки:

Используйте компонент Timer для создания анимации. Установите интервал через свойство Interval (в миллисекундах):

```
timer.Interval = 50; // 50 мс
```

```
timer.Tick += Timer_Tick; // Подпишитесь на событие Tick
```

В обработчике события Tick обновляйте координаты круга и проверяйте столкновения с границами:

```
if (x + radius >= pictureBox.Width || x - radius <= 0)  
{  
    dx = -dx; // Меняем направление по X  
}  
if (y + radius >= pictureBox.Height || y - radius <= 0)  
{  
    dy = -dy; // Меняем направление по Y  
}
```

Перерисуйте PictureBox с новыми координатами:

```
graphics.Clear(Color.White);  
graphics.FillEllipse(brush, x, y, radius * 2, radius * 2);
```

Задание 5. Graphics + ListBox + PictureBox.

Создайте форму с ListBox, где пользователь может выбрать одну из фигур (прямоугольник, круг, линия). При выборе фигуры в ListBox она должна отображаться на PictureBox. Добавьте возможность выбора цвета через контекстное меню (ContextMenuStrip), которое появляется при правом клике на PictureBox.

Подсказки:

Добавьте элементы в ListBox:

```
listBox.Items.AddRange(new string[] { "Прямоугольник", "Круг", "Линия" });
```

Обработывайте событие SelectedIndexChanged для определения выбранной фигуры:

```
string selectedShape = listBox.SelectedItem.ToString();
```

Создайте контекстное меню (ContextMenuStrip) с пунктами для выбора цвета. Привяжите его к PictureBox:

```
ContextMenuStrip contextMenu = new ContextMenuStrip();  
contextMenu.Items.Add("Красный", null, (s, e) => selectedColor = Color.Red);  
pictureBox.ContextMenuStrip = contextMenu;
```

Вариант 2

Задание 1. Graphics + ComboBox + PictureBox.

Создайте форму с PictureBox и ComboBox для выбора типа градиента (горизонтальный, вертикальный, диагональный). При выборе типа градиента нарисуйте его на PictureBox с использованием класса Graphics. Добавьте возможность выбора двух цветов через ColorDialog.

Подсказки:

Для создания градиента используйте класс LinearGradientBrush:

```
nearGradientBrush brush = new LinearGradientBrush(rect, color1, color2, angle);  
graphics.FillRectangle(brush, rect);
```

Угол градиента зависит от типа:

Горизонтальный: угол = 0.

Вертикальный: угол = 90.

Диагональный: угол = 45.

Выберите два цвета через ColorDialog:

```
ColorDialog dialog = new ColorDialog();  
if (dialog.ShowDialog() == DialogResult.OK)  
{  
    color1 = dialog.Color;  
}
```

Задание 2. Graphics + CheckBox + PictureBox.

Разработайте форму с PictureBox и чекбоксами (CheckBox) для включения/выключения отображения различных элементов: прямоугольника, круга и текста. При изменении состояния чекбоксов обновляйте изображение на PictureBox. Добавьте возможность выбора шрифта для текста через FontDialog.

Подсказки:

Проверяйте состояние чекбоксов с помощью свойства Checked:

```
if (checkBoxRectangle.Checked)  
{  
    graphics.DrawRectangle(pen, x, y, width, height);  
}
```

Для выбора шрифта используйте FontDialog:

```
FontDialog fontDialog = new FontDialog();  
if (fontDialog.ShowDialog() == DialogResult.OK)  
{  
    selectedFont = fontDialog.Font;  
}
```

Рисуйте текст с помощью метода DrawString:

```
graphics.DrawString("Текст", selectedFont, brush, x, y);
```

Задание 3. Graphics + Button + PictureBox.

Создайте форму с PictureBox и кнопками для выполнения различных действий:

"Нарисовать круг", "Нарисовать квадрат", "Очистить". Каждая кнопка выполняет соответствующее действие на PictureBox. Добавьте возможность изменения цвета через панель инструментов (ToolStrip).

Подсказки:

Используйте ToolStrip для добавления кнопок выбора цвета:

```
ToolStripButton redButton = new ToolStripButton("Красный");  
redButton.Click += (s, e) => selectedColor = Color.Red;  
toolStrip.Items.Add(redButton);
```

Для рисования фигур используйте методы DrawRectangle и FillEllipse.

Задание 4. Graphics + Mouse Events + PictureBox.

Разработайте форму с PictureBox, где пользователь может рисовать линии с помощью мыши. Используйте события MouseDown, MouseMove и MouseUp для отслеживания движений мыши и рисования линий. Добавьте возможность выбора цвета и толщины линии через ComboBox и NumericUpDown.

Подсказки:

Отслеживайте начальные координаты в событии MouseDown:

```
startPoint = e.Location;
```

```
isDrawing = true;
```

Рисуйте линию в событии MouseMove:

```
if (isDrawing)
```

```
{
```

```
    graphics.DrawLine(pen, startPoint, e.Location);
```

```
    startPoint = e.Location;
```

```
}
```

Завершите рисование в событии MouseUp:

```
isDrawing = false;
```

Задание 5. Graphics + Panel + PictureBox.

Создайте форму с Panel и PictureBox. В Panel разместите элементы управления (TrackBar для размера, ComboBox для цвета, RadioButton для выбора фигуры). При изменении параметров в Panel обновляйте изображение на PictureBox. Реализуйте рисование выбранной фигуры с указанными параметрами.

Подсказки:

Разместите элементы управления в Panel:

```
Panel panel = new Panel();
```

```
panel.Controls.Add(trackBar);
```

```
panel.Controls.Add(comboBox);
```

Обновляйте PictureBox при изменении параметров:

```
private void UpdatePictureBox()
```

```
{
```

```
    graphics.Clear(Color.White);
```

```
    // Нарисовать фигуру с новыми параметрами
```

```
}
```

Отчет должен содержать (см. образец):

- номер и тему лабораторной работы;
- фамилию, номер группы студента и вариант задания;
- скриншоты окна Visual Studio с исходным кодом программ и комментариями;
- скриншоты с результатами выполнения программ;
- пояснения, если необходимо;
- выводы.

Отчеты в формате **pdf** отправлять на email: **colledge20education23@gmail.com**