Практическая работа №2. Работа с графикой в WinForms.

Задачи:

1. Создать форму и определить её параметры (размер, положение при запуске, минимальный и максимальный размер, и др.).
2. Добавить кнопки управления (тип Button): «Точки», «Параметры», «Кривая», «Ломаная», «Безье», «Заполненная», «Движение», «Очистить».
3. Создать *обработчики нажатия кнопок формы*:

Кнопка формы	Функциональность
Точки	Нажатие кнопки включает, либо выключает режим добавления точек по щелчку мыши.
Параметры	Появляется ***дочернее окно*** для установки параметров графических объектов (толщина линий, размер точек, цвет точек и линий, характер движения: случайный или с сохранением фигуры, и др.)
Кривая	По отмеченным точкам строится замкнутая кривая (`DrawClosedCurve`)
Ломанная	По отмеченным точкам строится многоугольник (`DrawPolygone`)
Безье	По отмеченным точкам строятся кривые Безье (`DrawBeizers`). Обратите внимание, что кривые Безье строятся только при определенном числе точек.
Заполненная	По отмеченным точкам строится заполненная кривая (`FillCurve`)
Движение	Нажатие кнопки включает, либо выключает режим движения
Очистить	Очищение формы.

Предусмотрите два вида движения: «сохраняющий» режим – в этом случае все точки двигаются в одном направлении и с одной скоростью, при достижении границы точки отражаются; «случайный» режим – точки двигаются в разных направлениях и с разными скоростями, при достижении границы формы точки отражаются.

Предусмотрите возможность включения/выключения определенных режимов при нажатии кнопок формы. Например, нажатие кнопки «Точки» выключает режим движения, если он включен, и очищает экран. Нажатие кнопки «Кривая» выключает режим «Точки» и т.п.

4. Создать *обработчики нажатия клавиш:*

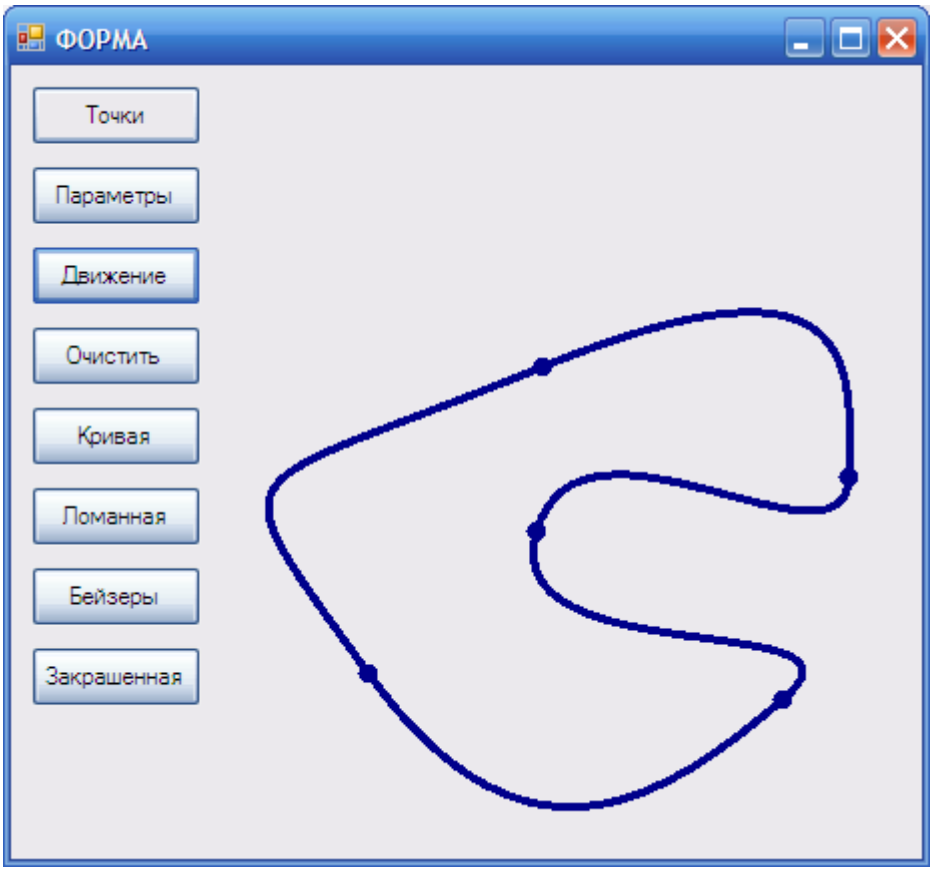
Кнопка	Функционал
Space	Включает/выключает режим движения.
+	Увеличивает скорость движения точек
-	Уменьшает скорость движения точек
Escape	Очищает содержимое формы

5. Создайте *обработчик событий мышки*, обеспечивающий возможность перетаскивания отдельных точек.

6. Добавьте обработчики нажатия клавиш:

Кнопка	Функциональность
Стрелки ↑ , ↓, →, ←	При выключенном режиме движения – перемещает фигуру в соответствующем направлении. При включенном режиме движения – увеличивает скорость в соответствующем направлении.

Внешний вид формы может быть ***произвольным***. Например,



Рекомендации к выполнению работы.

Для отображения графических объектов в форме необходимо получить объект типа **Graphics**, который инкапсулирует поверхность для рисования, связанную с элементом управления. Доступ к объекту Graphics можно получить в обработчике события **Paint**:

```
// Конструктор формы
public Form1()
{
      ..
      Paint += new PaintEventHandler(Form1_Paint);
      // Разрешается использовать сокращенную форму
      // Paint += Form1_Paint;

}
// Обработчик события Paint
void Form1_Paint(object sender, PaintEventArgs e)
{
      Graphics g = e.Graphics;
      g.FillClosedCurve(Brushes.ForestGreen, arPoints);
      ..
}
```

Рисование осуществляется с помощью методов, начинающихся с Fill.. или Draw.. . Методы первой группы отображают заполненные фигуры (эллипсы, многоугольники, кривые и т.д.) и используют графический объект «кисть» типа **Brush**. Класс **Brushes** позволяет получить доступ к уже созданным объектам Brush по имени цвета кисти: **`Brushes.ForestGreen`**. Методы группы Draw.. отображают контуры фигур без заполнения и используют графический объект «перо» типа **Pen**. Получить уже созданное перо определенного цвета можно в классе Pens.

Для отображения фигур необходимо предоставить массив точек типа Point. Объект Point представляет собой структуру с полями x, y типа Int32. Для хранения точек можно использовать массив или список типа List<Point>. В первом случае необходимо правильно обрабатывать ситуацию добавления новых точек. Использование списка упрощает добавление новых точек, но не позволяет изменять координаты существующих точек – требуется создание нового объекта:

```
// arPoints – список типа List<Point>
arPoints[i] = new Point(_x, _y);

// arrP – массив точек типа Point[]
arrP[i].X = _x; arrP[i].Y = _y;
```

Для перерисовки фигур необходимо обновлять поверхность формы. Для обновления можно воспользоваться методом **Refresh**, либо парой методов **Invalidate** и **Update**.

Отрисовку точек и фигур рекомендуется осуществлять в одном месте – в обработчике события Paint. Этот обработчик вызывается при необходимости перерисовки формы, а также при вызове пользователем метода **Refresh.**

В обработчиках нажатия кнопок достаточно осуществлять выбор той или иной кривой для рисования. Все варианты линий можно зафиксировать в перечислении для удобства работы:

```
enum eLineType { None, Curved, Filled, Polygone };

// Обработчика нажатия кнопки «Кривая»
void btCurved_Click(object sender, EventArgs e)
{
      lineType = eLineType.Curved;
      Refresh();
}
```

В зависимости от выбранной линии осуществляем отрисовку в обработчике Paint:

```
void Form1_Paint(object sender, PaintEventArgs e)
{
      Graphics g = e.Graphics;
      // есть точки для рисования
      if(arPoints.Count > 0)
      {
            // рисуем точки
            ..
            // рисуем фигуры
            if(lineType == eLineType.Curved)
                  g.FillClosedCurve(Brushes.ForestGreen, arPoints);
            else if (lineType == eLineType.Filled)
                  ..
      }
}
```

Для добавления точек необходимо обрабатывать событие мышки MouseClick. Обработчик события во втором аргументе содержит информацию о координатах положения указателя мышки при щелчке.

```
void button1_MouseDown(object sender, MouseEventArgs e)
{
      if(bAddPoints)
      {
            arPoints.Add(new Point(e.Location.X, e.Location.Y);
            Refresh();
```

```
        }
    }
```

Движение фигур реализуется как изменение координат точек и перерисовка формы. Для периодического изменения координат можно воспользоваться объектом типа Timer и событием Tick:

```
moveTimer.Interval = 30;
moveTimer.Tick += new EventHandler(TimerTickHandler);
```

Период возникновения события Tick задается свойством Interval в миллисекундах. Скоростью перемещения фигуры необходимо управлять не изменением интервала обновления, а изменением величины смещения точек.

Для обработки нажатия клавиш можно реализовать обработчик события KeyDown. При этом необходимо установить свойство KeyPreview для предварительной обработки событий клавиатуры в форме, а не в элементах управления (активные кнопки формы):

```
KeyPreview = true;
KeyDown += new KeyEventHandler(Form1_KeyDown);
```

Обработчик KeyDown в параметре события типа `KeyEventArgs` содержит свойство KeyCode, которое возвращает код нажатой клавиши и используемые модификаторы (Shift, Alt, Ctrl) в объекте типа Keys. Класс Keys содержит статические члены, которые можно использовать для установления нажатой клавиши:

```
switch (e.KeyCode)
{
    case(Keys.Add) :
                ..
                break;
    case(Keys.Space) :
                 ..
                break;
    default:
                break;
}
```

Перед выходом из обработчика необходимо установить флаг Handled аргумента `KeyEventArgs`, определяющий успешность обработки клавиши:

```
e.Handled = true;
```

Значение false приведет к последующей обработке клавиши дочерними элементами управления, например, кнопкой формы, находящейся в активном состоянии.

Обработка управляющих кнопок (стрелок) требует дополнительной перегрузки метода:

```
protected override bool ProcessCmdKey(ref Message msg,
                                      Keys keyData)
{
      ..
      return base.ProcessCmdKey(ref msg, keyData);
}
```

Статус обработки (успешно или нет) определяется возвращаемым значением: true – клавиша обработана, нет необходимости в последующей обработке; false – клавиша не обработана, необходимо вызывать другие обработчики события KeyDown.

Для реализации режима перемещения точек с помощью мышки необходимо обрабатывать события:

MouseDown (нажатие кнопки мышки): если указатель мышки находится в окрестности какой-либо точки, то включить режим перемещения, т.е. установить флаг перемещения: bDrag = true и запомнить активную точку `iPointToDrag`.

MouseMove (движение мышки): если режим перемещения включен, то перемещать активную точку в соответствии с текущим положением указателя мышки.

MouseUp (отпускание кнопки мышки): снять режим перемещения.