**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ**

**Федеральное государственное бюджетное образовательное учреждение высшего**

**профессионального образования**

**«Казанский национальный исследовательский технический университет**

**им. А.Н. Туполева-КАИ»**

Институт компьютерных технологий и защиты информации

Кафедра Прикладной математики и Информатики имени Ю.В. Кожевникова

Лабораторная работа №5

по дисциплине «Объектно-ориентированное программирование»

**Выполнил**

студент группы 4210

Нигамадянов Фанис Магефурович

Казань 2021

**Лабораторная работа №5. Создание библиотеки классов для графических примитивов с использованием виртуальных методов.**

Описание классов:

public abstract class Abstract\_properties

{

public int lx\_edge = 430;

public int ly\_edge = 38;

public int rx\_edge = 1100;

public int ry\_edge = 750;

public bool rhomb\_is\_visible { get; set; }

public int A { get; set; }

public int B { get; set; }

public Point[] points = new Point[4];

public Graphics gc;

public int x1 { get; set; }

public int y1 { get; set; }

public int x2 { get; set; }

public int y2 { get; set; }

protected Abstract\_properties() {}

public virtual void Show(Graphics gc, Color color)

{

}

public void Move(int x, int y)

{

}

public void MoveRhomb(int x, int y)

{

}

}

}

public class rectangle : square // пытаюсь все испортить

{

Random rnd = new Random();

public bool rectangle\_is\_visible { get; set; }

public rectangle()

{

}

public rectangle(int x1, int y1, int x2, int y2, bool rectangle\_is\_visible)

{

}

public override void Show(Graphics gc, Color color)

{

}

}

public class circle : Abstract\_properties

{

Random rnd = new Random();

public bool circle\_is\_visible { get; set; }

private int R;

public int r

{

get

{

return R;

}

set

{

R = value;

}

}

public circle()

{

}

public circle(int x1, int y1, int R, bool circle\_is\_visible)

{

}

public override void Show(Graphics gc, Color color)

{

}

}

public class ellipse : circle

{

Random rnd = new Random();

public int radius;

public bool ellipse\_is\_visible {get;set;}

public ellipse()

{}

public ellipse(int x1, int y1, int a, int b, bool ellipse\_is\_visible)

{

}

public override void Show(Graphics gc, Color color)

{

}

public void rotate\_for\_90()

{

}

}

public class rhomb:square

{

Random rnd = new Random();

public bool rhomb\_is\_visible { get; set; }

public int A { get; set; }

public int B { get; set; }

Point[] points = new Point[4];

public rhomb()

{

}

public rhomb(int x, int y, int a, int b)

{

}

public override void Show(Graphics gc, Color color)

{

}

}

public class square : Abstract\_properties

{

Random rnd = new Random();

public bool square\_is\_visible { get; set; }

public square()

{

}

public square(int x1, int y1, int x2, int y2, bool square\_is\_visible)

{

}

public override void Show(Graphics gc, Color color)

{

}

public void to\_double()

{

}

}

Описание свойств, полей и методов:

Для абстрактного класса:

1. Ограничение границ рисования

public int lx\_edge = 430;

public int ly\_edge = 38;

public int rx\_edge = 1100;

public int ry\_edge = 750;

1. Создание массива, содержащий пару координат

public Point[] points = new Point[4];

1. Объект класса Graphics

public Graphics gc;

1. Координаты

public int x1 { get; set; }

public int y1 { get; set; }

public int x2 { get; set; }

public int y2 { get; set; }

1. Конструктор по умолчанию

protected Abstract\_properties() {}

1. Виртуальный метод Show

public virtual void Show(Graphics gc, Color color)

{

Console.WriteLine("Just writing smthg 'cause I will override it");

}

1. Реальный метод Move

public void Move(int x, int y)

{

moving\_to moving = moving\_to.Right;

if (x > 0 && y == 0)

{

moving = moving\_to.Right;

}

else if (x < 0 && y == 0)

{

moving = moving\_to.Left;

}

else if (x == 0 && y > 0)

{

moving = moving\_to.Down;

}

else if (x == 0 && y < 0)

{

moving = moving\_to.Up;

}

switch (moving)

{

case moving\_to.Right:

x1 += x;

Show(gc, Color.Aquamarine);

break;

case moving\_to.Left:

x1 += x;

Show(gc, Color.Aquamarine);

break;

case moving\_to.Up:

y1 += y;

Show(gc, Color.Aquamarine);

break;

case moving\_to.Down:

y1 += y;

Show(gc, Color.Aquamarine);

break;

}

}

1. Реальный метод Move для ромба

public void MoveRhomb(int x, int y)

{

moving\_to moving = moving\_to.Right;

if (x > 0 && y == 0)

{

moving = moving\_to.Right;

}

else if (x < 0 && y == 0)

{

moving = moving\_to.Left;

}

else if (x == 0 && y > 0)

{

moving = moving\_to.Down;

}

else if (x == 0 && y < 0)

{

moving = moving\_to.Up;

}

switch (moving)

{

case moving\_to.Right:

points[0].X += x;

points[1].X += x;

points[2].X += x;

points[3].X += x;

Show(gc, Color.Aquamarine);

break;

case moving\_to.Left:

points[0].X += x;

points[1].X += x;

points[2].X += x;

points[3].X += x;

Show(gc, Color.Aquamarine);

break;

case moving\_to.Up:

points[0].Y += y;

points[1].Y += y;

points[2].Y += y;

points[3].Y += y;

Show(gc, Color.Aquamarine);

break;

case moving\_to.Down:

points[0].Y += y;

points[1].Y += y;

points[2].Y += y;

points[3].Y += y;

Show(gc, Color.Aquamarine);

break;

}

}

}

Для круга:

1)Ограничение поля случайности

int lx\_edge = 430;

int ly\_edge = 38;

int rx\_edge = 1100;

int ry\_edge = 750;

2) Объект класса Random

Random rnd = new Random();

4) Радиус окружности

public int R { get; set; }

5) Поля для отслеживания видимости

public bool circle\_is\_visible { get; set; }

6) Конструктор по умолчанию

public circle()

{

circle\_center.x = rnd.Next(lx\_edge, rx\_edge - 100);

circle\_center.y = rnd.Next(ly\_edge, ry\_edge + 100);

this.R = rnd.Next(10, 200);

this.circle\_is\_visible = true;

}

7) Конструктор с заданными параметрами

public circle( int x1, int y1, int R, bool circle\_is\_visible)

{

circle\_center.x = x1;

circle\_center.y = y1;

this.R = R;

this.circle\_is\_visible = circle\_is\_visible;

}

8) Показ фигуры

public override void show(Graphics gc, Color color)

{

if (circle\_is\_visible == true)

{

Pen pen = new Pen(color, 5);

gc.DrawEllipse(pen, circle\_center.x-R, circle\_center.y-R, 2\*R, 2\*R);

}

}

Для прямоугольника:

1. Конструктор по умолчанию

public rectangle()

{

x1 = rnd.Next(lx\_edge, rx\_edge - 80);

x2 = rnd.Next(100, 400);

y1 = rnd.Next(ly\_edge, ry\_edge - 80);

y2 = rnd.Next(100, 400);

rectangle\_is\_visible = true;

}

1. Конструктор с заданными параметрами

public rectangle(int x1, int y1, int x2, int y2, bool rectangle\_is\_visible)

{

this.x1 = x1;

this.x2 = x2;

this.y1 = y1;

this.y2 = y2;

this.rectangle\_is\_visible = rectangle\_is\_visible;

}

1. Показ фигур

public override void Show(Graphics gc, Color color)

{

if (rectangle\_is\_visible == true)

{

Pen pen = new Pen(color, 5);

gc.DrawRectangle(pen, x1, y1, x2, y2);

}

}

Для класса ромб:

1. Последняя строка – создание массива, хранящего пару значений

Random rnd = new Random();

public bool rhomb\_is\_visible { get; set; }

public int A { get; set; }

public int B { get; set; }

Point[] points = new Point[4];

1. Конструктор для создания кольца

public rhomb()

{

A = rnd.Next(50, 500);

B = rnd.Next(50, 500);

Point point1 = new Point(rnd.Next(200, 1000), rnd.Next(200,800));

Point point2 = new Point(point1.X+B/2, point1.Y+A/2);

Point point3 = new Point(point1.X, point1.Y+A);

Point point4 = new Point(point1.X - B / 2, point1.Y + A / 2);

points[0] = point1;

points[1] = point2;

points[2] = point3;

points[3] = point4;

rhomb\_is\_visible = true;

}

1. Конструктор для создания кольца по заданным параметрам

public rhomb(int x, int y, int a, int b)

{

A = a;

B = b;

Point point1 = new Point(x, y);

Point point2 = new Point(point1.X + B / 2, point1.Y + A / 2);

Point point3 = new Point(point1.X, point1.Y + A);

Point point4 = new Point(point1.X - B / 2, point1.Y + A / 2);

points[0] = point1;

points[1] = point2;

points[2] = point3;

points[3] = point4;

rhomb\_is\_visible = true;

}

1. Метод для показа фигуры

public override void Show(Graphics gc, Color color)

{

if (rhomb\_is\_visible == true)

{

Pen pen = new Pen(color, 5);

gc.DrawPolygon(pen, points);

}

}

Для класса эллипс:

1. Вторая полуось эллипса

public int radius;

1. Конструктор по умолчанию для эллипса

public ellipse()

{

x1 = rnd.Next(lx\_edge, rx\_edge - 80);

y1 = rnd.Next(100, 400);

this.r = rnd.Next(10, 250);

this.radius = rnd.Next(10, 250);

ellipse\_is\_visible = true;

}

1. Конструктор для создания эллипса с заданными параметрами

public ellipse(int x1, int y1, int a, int b, bool ellipse\_is\_visible)

{

this.x1 = x1;

this.y1 = y1;

r = a / 2;

radius = b / 2;

this.ellipse\_is\_visible = ellipse\_is\_visible;

}

1. Метод для показа эллипса

public override void Show(Graphics gc, Color color)

{

if (ellipse\_is\_visible == true)

{

Pen pen = new Pen(color, 5);

gc.DrawEllipse(pen, x1-r, y1-radius, 2\*r, 2\*radius);

}

}

1. Метод для поворота эллипса

public void rotate\_for\_90()

{

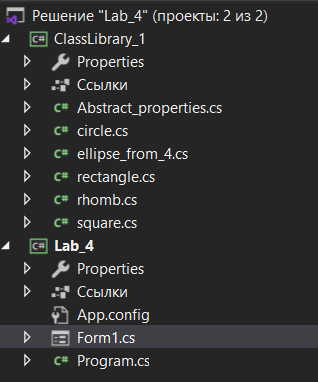
int changer = this.r;

this.r = radius;

radius = changer;

}

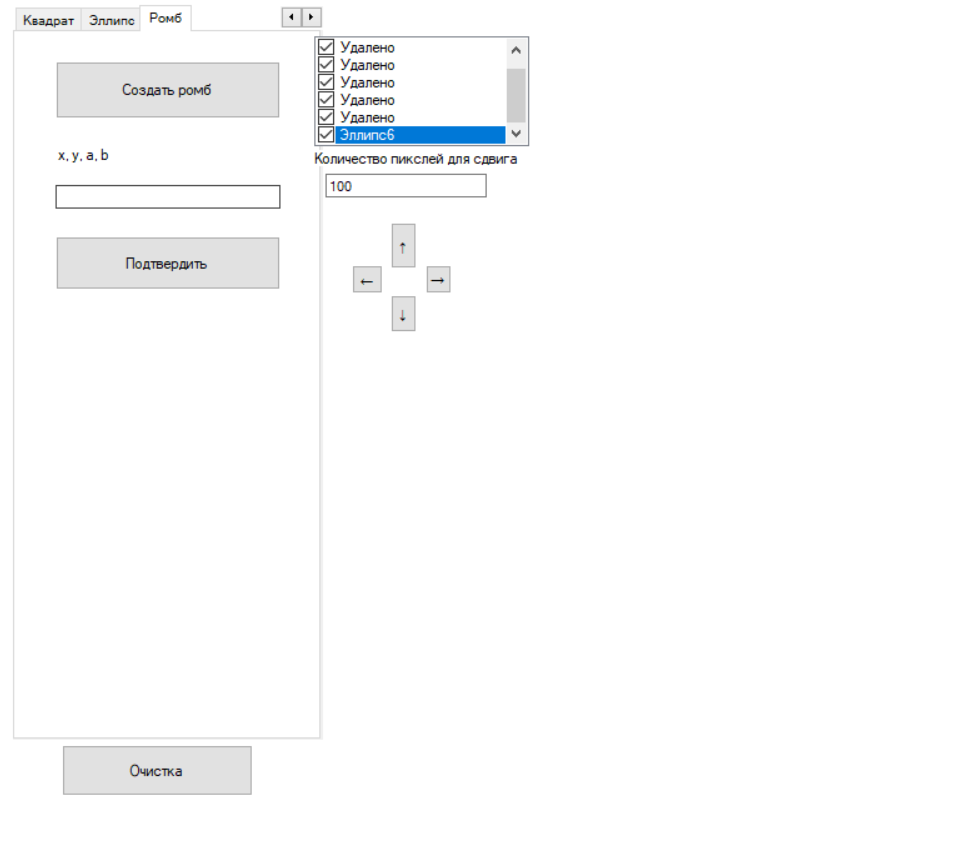
Структура программы:



1. ClassLibrary1 - библиотека классов, определяющие фигуры. Также имеется абстрактный класс
2. Файл Form1.cs, содержащий файл Form1.designer.cs, определяющий кнопки и их обработчики, файл Form1.res.
3. Файл Program.cs, запускающий форму

Руководство пользования:

Имеются пять вкладок для разных фигур. При помощи кнопки «Создать (соответствующая фигура)» вы можете создать фигуру со случайными координатами и размерами. Если в текстовом поле ввести соответствующие координаты и размеры, можно получить фигуру с заданными координатами. Для перемещения и удаления фигур воспользуйтесь окном CheckBox, выберите интересующую фигуру и нажимайте на соответствующие кнопки (стрелки для перемещения, кнопку очистки и т.д.)



Листинг программы:

public partial class Form1 : Form // пытаюсь все испортить

{

string row\_name; // для имени элемента в CheckBox

int cb\_for\_circ = 0;

int cb\_for\_rect = 0; // индекс для фигур в Checkbox

int cb\_for\_square = 0;

int cb\_for\_ellipse = 0;

int cb\_for\_rhomb = 0;

int move\_pixels = 10;

int for\_circ = 0;

int for\_rect = 0; // индексация по массиву

int for\_square = 0;

int for\_ellipse = 0;

int for\_rhomb = 0;

string[] circ\_data;

int user\_circ\_x; // координаты, если пользователь сам захочет их ввести

int user\_circ\_y;

int user\_circ\_R;

string[] rect\_data;

int user\_rect\_x;

int user\_rect\_y;

int user\_rect\_a;

int user\_rect\_b;

string[] square\_data;

int user\_square\_x;

int user\_square\_y;

int user\_square\_a;

string[] ellipse\_data;

int user\_ellipse\_x;

int user\_ellipse\_y;

int user\_ellipse\_a;

int user\_ellipse\_b;

string[] rhomb\_data;

int user\_rhomb\_x;

int user\_rhomb\_y;

int user\_rhomb\_a;

int user\_rhomb\_b;

Graphics gr;

static int num\_of\_circles = 20;

static int num\_of\_rectangles = 20;

static int num\_of\_squares = 20;

static int num\_of\_ellipses = 20;

static int num\_of\_rhombes = 20;

circle[] m\_circles = new circle[num\_of\_circles];

rectangle[] m\_rectangles = new rectangle[num\_of\_rectangles];

square[] m\_squares= new square[num\_of\_squares];

ellipse[] m\_ellipses = new ellipse[num\_of\_ellipses];

rhomb[] m\_rhomb = new rhomb[num\_of\_rhombes];

public Form1()

{

InitializeComponent();

this.WindowState = FormWindowState.Maximized;

this.BackColor = System.Drawing.Color.White;

}

private void button1\_Click(object sender, EventArgs e)

{

if (textBox2.Text != "")

{

m\_circles[for\_circ] =

new circle(user\_circ\_x, user\_circ\_y, user\_circ\_R, true);

}

else

{

m\_circles[for\_circ] = new circle();

}

m\_circles[for\_circ].Show(gr = CreateGraphics(), Color.Aquamarine); //заполнение массива объектов одной группы

for\_circ++;

MessageBox.Show("Круг успешно создан!");

row\_name = "Круг";

checkedListBox1.Items.Add(row\_name + cb\_for\_circ);

row\_name = "";

cb\_for\_circ++;

}// создание круга

private void button2\_Click(object sender, EventArgs e) // создание прямоугольника

{

if (textBox3.Text != "")

{

m\_rectangles[for\_rect] =

new rectangle(user\_rect\_x - user\_rect\_a/2, user\_rect\_y - user\_rect\_b/2,

user\_rect\_a, user\_rect\_b, true);

}

else

{

m\_rectangles[for\_rect] = new rectangle();

}

m\_rectangles[for\_rect].Show(gr = CreateGraphics(), Color.Aquamarine); //заполнение массива объектов одной группы

for\_rect++;

MessageBox.Show("Прямоугольник успешно создан!");

row\_name = "Прямоугольник";

checkedListBox1.Items.Add(row\_name + cb\_for\_rect);

row\_name = "";

cb\_for\_rect++;

}

private void button3\_Click(object sender, EventArgs e) // создание квадрата

{

if (textBox4.Text != "")

{

m\_squares[for\_square] =

new square(user\_square\_x - user\_square\_a / 2, user\_square\_y - user\_square\_a / 2,

user\_square\_a, user\_square\_a, true);

}

else

{

m\_squares[for\_square] = new square();

}

m\_squares[for\_square].Show(gr = CreateGraphics(), Color.Aquamarine); //заполнение массива объектов одной группы

for\_square++;

MessageBox.Show("Квадрат успешно создан!");

row\_name = "Квадрат";

checkedListBox1.Items.Add(row\_name + cb\_for\_square);

row\_name = "";

cb\_for\_square++;

}

private void button13\_Click(object sender, EventArgs e)

{

if (textBox5.Text != "")

{

m\_ellipses[for\_ellipse] = new ellipse(user\_ellipse\_x, user\_ellipse\_y,

user\_ellipse\_a, user\_ellipse\_b, true);

}

else

{

m\_ellipses[for\_ellipse] = new ellipse();

}

m\_ellipses[for\_ellipse].Show(gr = CreateGraphics(), Color.Aquamarine); //заполнение массива объектов одной группы

for\_ellipse++;

MessageBox.Show("Эллипс успешно создан!");

row\_name = "Эллипс";

checkedListBox1.Items.Add(row\_name + cb\_for\_ellipse);

row\_name = "";

cb\_for\_ellipse++;

}// создание эллипса

private void button16\_Click(object sender, EventArgs e)// создание ромба

{

if (textBox6.Text != "")

{

m\_rhomb[for\_rhomb] = new rhomb(user\_rhomb\_x, user\_rhomb\_y, user\_rhomb\_a, user\_rhomb\_b);

}

else

{

m\_rhomb[for\_rhomb] = new rhomb();

}

m\_rhomb[for\_rhomb].Show(gr = CreateGraphics(), Color.Aquamarine);

for\_rhomb++;

MessageBox.Show("Ромб создан!");

row\_name = "Ромб";

checkedListBox1.Items.Add(row\_name + cb\_for\_rhomb);

row\_name = "";

cb\_for\_rhomb++;

}

private void button4\_Click(object sender, EventArgs e)

{

for (int k = 0; k < checkedListBox1.Items.Count; k++) // проверка, что есть отмеченные, чтобы не стереть все

{

if (checkedListBox1.GetItemChecked(k))

{

gr.Clear(Color.White);

break;

}

}

int i = 0;

int counter = 0;

bool square\_exist = false;

foreach (var item in checkedListBox1.Items.OfType<string>().ToList())

{

if (item.Contains("Квадрат"))

{

square\_exist = true;

if (checkedListBox1.GetItemChecked(i))

{

m\_squares[int.Parse(item.Replace("Квадрат", ""))].to\_double();

counter++;

}

m\_squares[int.Parse(item.Replace("Квадрат", ""))].Show(gr, Color.Aquamarine);

}

else if (item.Contains("Круг"))

{

m\_circles[int.Parse(item.Replace("Круг", ""))].Show(gr, Color.Aquamarine);

}

else if (item.Contains("Прямоугольник"))

{

m\_rectangles[int.Parse(item.Replace("Прямоугольник", ""))].Show(gr, Color.Aquamarine);

}

else if (item.Contains("Эллипс"))

{

m\_ellipses[int.Parse(item.Replace("Эллипс", ""))].Show(gr, Color.Aquamarine);

}

else if (item.Contains("Ромб"))

{

m\_rhomb[int.Parse(item.Replace("Ромб", ""))].Show(gr, Color.Aquamarine);

}

i++;

}

if (counter == 0)

{

MessageBox.Show("Не выбрано ни одного квадрата");

}

if (!square\_exist)

{

MessageBox.Show("Квадраты не созданы!");

}

} // удвоение длины квадрата

private void button15\_Click(object sender, EventArgs e)// поворот эллипса

{

for (int k = 0; k < checkedListBox1.Items.Count; k++) // проверка, что есть отмеченные, чтобы не стереть все

{

if (checkedListBox1.GetItemChecked(k))

{

gr.Clear(Color.White);

break;

}

}

int i = 0;

int counter = 0;

bool ellipse\_exist = false;

foreach (var item in checkedListBox1.Items.OfType<string>().ToList())

{

if (item.Contains("Эллипс"))

{

ellipse\_exist = true;

if (checkedListBox1.GetItemChecked(i))

{

m\_ellipses[int.Parse(item.Replace("Эллипс", ""))].rotate\_for\_90();

counter++;

}

m\_ellipses[int.Parse(item.Replace("Эллипс", ""))].Show(gr, Color.Aquamarine);

}

else if (item.Contains("Круг"))

{

m\_circles[int.Parse(item.Replace("Круг", ""))].Show(gr, Color.Aquamarine);

}

else if (item.Contains("Прямоугольник"))

{

m\_rectangles[int.Parse(item.Replace("Прямоугольник", ""))].Show(gr, Color.Aquamarine);

}

else if (item.Contains("Квадрат"))

{

m\_squares[int.Parse(item.Replace("Квадрат", ""))].Show(gr, Color.Aquamarine);

}

else if (item.Contains("Ромб"))

{

m\_rhomb[int.Parse(item.Replace("Ромб", ""))].Show(gr, Color.Aquamarine);

}

i++;

}

if (counter == 0)

{

MessageBox.Show("Не выбрано ни одного эллипса");

}

if (!ellipse\_exist)

{

MessageBox.Show("Эллипсы не созданы!");

}

}

private void button5\_Click(object sender, EventArgs e)

{

for (int k = 0; k < checkedListBox1.Items.Count; k++) // проверка, что есть отмеченные, чтобы не стереть все

{

if (checkedListBox1.GetItemChecked(k))

{

gr.Clear(Color.White);

break;

}

}

int i = 0;

foreach (var item in checkedListBox1.Items.OfType<string>().ToList())

{

if (item.Contains("Круг"))

{

if (checkedListBox1.GetItemChecked(i))

{

m\_circles[int.Parse(item.Replace("Круг", ""))].circle\_is\_visible = false;

checkedListBox1.Items[i] = "Удалено";

}

m\_circles[int.Parse(item.Replace("Круг", ""))].Show(gr, Color.Aquamarine);

}

else if (item.Contains("Прямоугольник"))

{

if (checkedListBox1.GetItemChecked(i))

{

m\_rectangles[int.Parse(item.Replace("Прямоугольник", ""))].rectangle\_is\_visible = false;

checkedListBox1.Items[i] = "Удалено";

}

m\_rectangles[int.Parse(item.Replace("Прямоугольник", ""))].Show(gr, Color.Aquamarine);

}

else if (item.Contains("Квадрат"))

{

if (checkedListBox1.GetItemChecked(i))

{

m\_squares[int.Parse(item.Replace("Квадрат", ""))].square\_is\_visible = false;

checkedListBox1.Items[i] = "Удалено";

}

m\_squares[int.Parse(item.Replace("Квадрат", ""))].Show(gr, Color.Aquamarine);

}

else if (item.Contains("Эллипс"))

{

if (checkedListBox1.GetItemChecked(i))

{

m\_ellipses[int.Parse(item.Replace("Эллипс", ""))].ellipse\_is\_visible = false;

checkedListBox1.Items[i] = "Удалено";

}

m\_ellipses[int.Parse(item.Replace("Эллипс", ""))].Show(gr, Color.Aquamarine);

}

else if (item.Contains("Ромб"))

{

if (checkedListBox1.GetItemChecked(i))

{

m\_rhomb[int.Parse(item.Replace("Ромб", ""))].rhomb\_is\_visible = false;

checkedListBox1.Items[i] = "Удалено";

}

m\_rhomb[int.Parse(item.Replace("Ромб", ""))].Show(gr, Color.Aquamarine);

}

i++;

}

} // очистка

private void button8\_Click(object sender, EventArgs e)

{

for (int k = 0; k < checkedListBox1.Items.Count; k++) // проверка, что есть отмеченные, чтобы не стереть все

{

if (checkedListBox1.GetItemChecked(k))

{

gr.Clear(Color.White);

break;

}

}

int i = 0;

foreach (var item in checkedListBox1.Items.OfType<string>().ToList())

{

if (item.Contains("Круг"))

{

if (checkedListBox1.GetItemChecked(i))

{

m\_circles[int.Parse(item.Replace("Круг", ""))].Move(move\_pixels, 0);

}

m\_circles[int.Parse(item.Replace("Круг", ""))].Show(gr, Color.Aquamarine);

}

else if (item.Contains("Прямоугольник"))

{

if (checkedListBox1.GetItemChecked(i))

{

m\_rectangles[int.Parse(item.Replace("Прямоугольник", ""))].Move(move\_pixels, 0);

}

m\_rectangles[int.Parse(item.Replace("Прямоугольник", ""))].Show(gr, Color.Aquamarine);

}

else if (item.Contains("Квадрат"))

{

if (checkedListBox1.GetItemChecked(i))

{

m\_squares[int.Parse(item.Replace("Квадрат", ""))].Move(move\_pixels, 0);

}

m\_squares[int.Parse(item.Replace("Квадрат", ""))].Show(gr, Color.Aquamarine);

}

else if (item.Contains("Эллипс"))

{

if (checkedListBox1.GetItemChecked(i))

{

m\_ellipses[int.Parse(item.Replace("Эллипс", ""))].Move(move\_pixels, 0);

}

m\_ellipses[int.Parse(item.Replace("Эллипс", ""))].Show(gr, Color.Aquamarine);

}

else if (item.Contains("Ромб"))

{

if (checkedListBox1.GetItemChecked(i))

{

m\_rhomb[int.Parse(item.Replace("Ромб", ""))].Move(move\_pixels, 0);

}

m\_rhomb[int.Parse(item.Replace("Ромб", ""))].Show(gr, Color.Aquamarine);

}

i++;

}

} // сдвиг вправо

private void button7\_Click(object sender, EventArgs e)

{

for (int k = 0; k < checkedListBox1.Items.Count; k++) // проверка, что есть отмеченные, чтобы не стереть все

{

if (checkedListBox1.GetItemChecked(k))

gr.Clear(Color.White);

}

int i = 0;

foreach (var item in checkedListBox1.Items.OfType<string>().ToList())

{

if (item.Contains("Круг"))

{

if (checkedListBox1.GetItemChecked(i))

{

m\_circles[int.Parse(item.Replace("Круг", ""))].Move(-move\_pixels, 0);

}

m\_circles[int.Parse(item.Replace("Круг", ""))].Show(gr, Color.Aquamarine);

}

else if (item.Contains("Прямоугольник"))

{

if (checkedListBox1.GetItemChecked(i))

{

m\_rectangles[int.Parse(item.Replace("Прямоугольник", ""))].Move(-move\_pixels, 0);

}

m\_rectangles[int.Parse(item.Replace("Прямоугольник", ""))].Show(gr, Color.Aquamarine);

}

else if (item.Contains("Квадрат"))

{

if (checkedListBox1.GetItemChecked(i))

{

m\_squares[int.Parse(item.Replace("Квадрат", ""))].Move(-move\_pixels, 0);

}

m\_squares[int.Parse(item.Replace("Квадрат", ""))].Show(gr, Color.Aquamarine);

}

else if (item.Contains("Эллипс"))

{

if (checkedListBox1.GetItemChecked(i))

{

m\_ellipses[int.Parse(item.Replace("Эллипс", ""))].Move(-move\_pixels, 0);

}

m\_ellipses[int.Parse(item.Replace("Эллипс", ""))].Show(gr, Color.Aquamarine);

}

else if (item.Contains("Ромб"))

{

if (checkedListBox1.GetItemChecked(i))

{

m\_rhomb[int.Parse(item.Replace("Ромб", ""))].Move(-move\_pixels, 0);

}

m\_rhomb[int.Parse(item.Replace("Ромб", ""))].Show(gr, Color.Aquamarine);

}

i++;

}

} // сдвиг влево

private void button9\_Click(object sender, EventArgs e)

{

for (int k = 0; k < checkedListBox1.Items.Count; k++) // проверка, что есть отмеченные, чтобы не стереть все

{

if (checkedListBox1.GetItemChecked(k))

gr.Clear(Color.White);

}

int i = 0;

foreach (var item in checkedListBox1.Items.OfType<string>().ToList())

{

if (item.Contains("Круг"))

{

if (checkedListBox1.GetItemChecked(i))

{

m\_circles[int.Parse(item.Replace("Круг", ""))].Move(0, move\_pixels);

}

m\_circles[int.Parse(item.Replace("Круг", ""))].Show(gr, Color.Aquamarine);

}

else if (item.Contains("Прямоугольник"))

{

if (checkedListBox1.GetItemChecked(i))

{

m\_rectangles[int.Parse(item.Replace("Прямоугольник", ""))].Move(0, move\_pixels);

}

m\_rectangles[int.Parse(item.Replace("Прямоугольник", ""))].Show(gr, Color.Aquamarine);

}

else if (item.Contains("Квадрат"))

{

if (checkedListBox1.GetItemChecked(i))

{

m\_squares[int.Parse(item.Replace("Квадрат", ""))].Move(0, move\_pixels);

}

m\_squares[int.Parse(item.Replace("Квадрат", ""))].Show(gr, Color.Aquamarine);

}

else if (item.Contains("Эллипс"))

{

if (checkedListBox1.GetItemChecked(i))

{

m\_ellipses[int.Parse(item.Replace("Эллипс", ""))].Move(0, move\_pixels);

}

m\_ellipses[int.Parse(item.Replace("Эллипс", ""))].Show(gr, Color.Aquamarine);

}

else if (item.Contains("Ромб"))

{

if (checkedListBox1.GetItemChecked(i))

{

m\_rhomb[int.Parse(item.Replace("Ромб", ""))].Move(0, move\_pixels);

}

m\_rhomb[int.Parse(item.Replace("Ромб", ""))].Show(gr, Color.Aquamarine);

}

i++;

}

} // сдвиг вниз

private void button6\_Click(object sender, EventArgs e)

{

for (int k = 0; k < checkedListBox1.Items.Count; k++) // проверка, что есть отмеченные, чтобы не стереть все

{

if (checkedListBox1.GetItemChecked(k))

gr.Clear(Color.White);

}

int i = 0;

foreach (var item in checkedListBox1.Items.OfType<string>().ToList())

{

if (item.Contains("Круг"))

{

if (checkedListBox1.GetItemChecked(i))

{

m\_circles[int.Parse(item.Replace("Круг", ""))].Move(0, -move\_pixels);

}

m\_circles[int.Parse(item.Replace("Круг", ""))].Show(gr, Color.Aquamarine);

}

else if (item.Contains("Прямоугольник"))

{

if (checkedListBox1.GetItemChecked(i))

{

m\_rectangles[int.Parse(item.Replace("Прямоугольник", ""))].Move(0, -move\_pixels);

}

m\_rectangles[int.Parse(item.Replace("Прямоугольник", ""))].Show(gr, Color.Aquamarine);

}

else if (item.Contains("Квадрат"))

{

if (checkedListBox1.GetItemChecked(i))

{

m\_squares[int.Parse(item.Replace("Квадрат", ""))].Move(0, -move\_pixels);

}

m\_squares[int.Parse(item.Replace("Квадрат", ""))].Show(gr, Color.Aquamarine);

}

else if (item.Contains("Эллипс"))

{

if (checkedListBox1.GetItemChecked(i))

{

m\_ellipses[int.Parse(item.Replace("Эллипс", ""))].Move(0, -move\_pixels);

}

m\_ellipses[int.Parse(item.Replace("Эллипс", ""))].Show(gr, Color.Aquamarine);

}

else if (item.Contains("Ромб"))

{

if (checkedListBox1.GetItemChecked(i))

{

m\_rhomb[int.Parse(item.Replace("Ромб", ""))].Move(0, -move\_pixels);

}

m\_rhomb[int.Parse(item.Replace("Ромб", ""))].Show(gr, Color.Aquamarine);

}

i++;

}

} // сдвиг вверх

private void textBox1\_TextChanged(object sender, EventArgs e) // сдвиг по пикселям

{

if (textBox1.Text == "")

move\_pixels = 10;

else

{

move\_pixels = int.Parse(textBox1.Text);

}

}

private void button10\_Click(object sender, EventArgs e)

{

try

{

circ\_data = textBox2.Text.Split(' ');

user\_circ\_x = int.Parse(circ\_data[0]);

user\_circ\_y = int.Parse(circ\_data[1]);

user\_circ\_R = int.Parse(circ\_data[2]);

}

catch

{

MessageBox.Show("Возникла ошибка");

}

} // задание значений кругу

private void button11\_Click(object sender, EventArgs e) // задание значений для прямо-ка

{

try

{

rect\_data = textBox3.Text.Split(' ');

user\_rect\_x = int.Parse(rect\_data[0]);

user\_rect\_y = int.Parse(rect\_data[1]);

user\_rect\_a = int.Parse(rect\_data[2]);

user\_rect\_b = int.Parse(rect\_data[3]);

}

catch

{

MessageBox.Show("Возникла ошибка");

}

}

private void button12\_Click(object sender, EventArgs e)

{

try

{

square\_data = textBox4.Text.Split(' ');

user\_square\_x = int.Parse(square\_data[0]);

user\_square\_y = int.Parse(square\_data[1]);

user\_square\_a = int.Parse(square\_data[2]);

}

catch

{

MessageBox.Show("Возникла ошибка");

}

} // задание значений квадрата

private void button14\_Click(object sender, EventArgs e)

{

try

{

ellipse\_data = textBox5.Text.Split(' ');

user\_ellipse\_x = int.Parse(ellipse\_data[0]);

user\_ellipse\_y = int.Parse(ellipse\_data[1]);

user\_ellipse\_a = int.Parse(ellipse\_data[2]);

user\_ellipse\_b = int.Parse(ellipse\_data[3]);

}

catch

{

MessageBox.Show("Возникла ошибка");

}

}// задание значений для эллипса

private void button17\_Click(object sender, EventArgs e ромба

{

try

{

rhomb\_data = textBox6.Text.Split(' ');

user\_rhomb\_x = int.Parse(rhomb\_data[0]);

user\_rhomb\_y = int.Parse(rhomb\_data[1]);

user\_rhomb\_a = int.Parse(rhomb\_data[2]);

user\_rhomb\_b = int.Parse(rhomb\_data[3]);

}

catch

{

MessageBox.Show("Возникла ошибка");

}

}

}