**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 №6

по дисциплине «Объектно-ориентированное программирование»

**Выполнил**

студент группы 4210

Нигамадянов Фанис Магефурович

Казань 2021

**Лабораторная работа №6. Использование полиморфных объектных указателей.**

Описание методов, свойств и полей классов:

Для основного класса:

1. Поле для хранения варианта

static int num\_of\_variant = 27;

1. Создание объекта класса Graphics

Graphics gr;

1. Объект класса Random

Random my\_rnd = new Random();

1. Поле для хранения числа пискселей для перемещения

int move\_pixels = 26;

1. Метод для инициализации формы

public Form1()

{

InitializeComponent();

this.WindowState = FormWindowState.Maximized;

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

}

1. Массив, хранящий фигуры

Abstract\_properties[] my\_objects = new Abstract\_properties[num\_of\_variant];

1. Поля для отслеживания, нажата ли кнопка

bool button\_18\_is\_clicked = false;

1. Для добавления фигур в массив

private void button18\_Click(object sender, EventArgs e)

{

if (button\_18\_is\_clicked)

{

MessageBox.Show("Вы уже создали массив фигур!");

return;

}

int my\_rand = my\_rnd.Next(1, 50);

Thread.Sleep(10);

for (int i = 0; i < num\_of\_variant; i++)

{

if (my\_rand >= 1 && my\_rand < 11)

{

my\_objects[i] = new circle();

}

else if (my\_rand >= 11 && my\_rand < 21)

{

my\_objects[i] = new ellipse();

}

else if (my\_rand >= 21 && my\_rand < 31)

{

my\_objects[i] = new rectangle();

}

else if (my\_rand >= 31 && my\_rand < 41)

{

my\_objects[i] = new square();

}

else if (my\_rand >= 41 && my\_rand < 51)

{

my\_objects[i] = new rhomb();

}

my\_rand = my\_rnd.Next(1, 50);

Thread.Sleep(10);

}

button\_18\_is\_clicked = true;

MessageBox.Show("Массив фигур создан!");

}

1. Метод для отрисовки фигур

private void button19\_Click(object sender, EventArgs e)

{

if (!button\_18\_is\_clicked)

{

MessageBox.Show("Вы еще не создали массив фигур!");

return;

}

foreach (object item in my\_objects)

{

if (item is circle circ\_type)

{

circ\_type.circle\_is\_visible = true;

circ\_type.Show(gr = CreateGraphics(), Color.Aquamarine);

}

else if (item is ellipse ell\_type)

{

ell\_type.ellipse\_is\_visible = true;

ell\_type.Show(gr = CreateGraphics(), Color.Aquamarine);

}

else if (item is rectangle rect\_type)

{

rect\_type.rectangle\_is\_visible = true;

rect\_type.Show(gr = CreateGraphics(), Color.Aquamarine);

}

else if (item is rhomb rh\_type)

{

rh\_type.rhomb\_is\_visible = true;

rh\_type.Show(gr = CreateGraphics(), Color.Aquamarine);

}

else if (item is square square\_type)

{

square\_type.square\_is\_visible = true;

square\_type.Show(gr = CreateGraphics(), Color.Aquamarine);

}

}

} //

1. Очистка холста

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

{

if (!button\_18\_is\_clicked)

{

MessageBox.Show("Вы еще не создали массив фигур!");

return;

}

gr.Clear(Color.White);

foreach (object item in my\_objects)

{

if (item is ellipse ell\_type)

{

ell\_type.Delete();

}

else if(item is circle circ\_type)

{

circ\_type.Delete();

}

else if (item is rectangle rect\_type)

{

rect\_type.Delete();

}

else if (item is rhomb rh\_type)

{

rh\_type.Delete();

}

else if (item is square square\_type)

{

square\_type.Delete();

}

}

}// ОЧИСТКА

1. Поворот эллипса

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

{

if (!button\_18\_is\_clicked)

{

MessageBox.Show("Вы еще не создали массив фигур!");

return;

}

gr.Clear(Color.White);

foreach (object item in my\_objects)

{

if (item is ellipse ell\_type)

{

ell\_type.rotate\_for\_90();

ell\_type.Show(gr = CreateGraphics(), Color.Aquamarine);

}

else if (item is circle circ\_type)

{

circ\_type.Show(gr = CreateGraphics(), Color.Aquamarine);

}

else if (item is rectangle rect\_type)

{

rect\_type.Show(gr = CreateGraphics(), Color.Aquamarine);

}

else if (item is rhomb rh\_type)

{

rh\_type.Show(gr = CreateGraphics(), Color.Aquamarine);

}

else if (item is square square\_type)

{

square\_type.Show(gr = CreateGraphics(), Color.Aquamarine);

}

}

}// ПОВОРОТ ЭЛЛИПСА

1. Приведение к одной точке

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

{

Random rnd = new Random();

int rnd\_x = rnd.Next(300, 1200);

int rnd\_y = rnd.Next(300, 600);

if (!button\_18\_is\_clicked)

{

MessageBox.Show("Вы еще не создали массив фигур!");

return;

}

gr.Clear(Color.White);

foreach (object item in my\_objects)

{

if (item is ellipse ell\_type)

{

ell\_type.x1 = rnd\_x;

ell\_type.y1 = rnd\_y;

ell\_type.Show(gr = CreateGraphics(), Color.Aquamarine);

}

else if (item is circle circ\_type)

{

circ\_type.x1 = rnd\_x;

circ\_type.y1 = rnd\_y;

circ\_type.Show(gr = CreateGraphics(), Color.Aquamarine);

}

else if (item is rectangle rect\_type)

{

rect\_type.x1 = rnd\_x;

rect\_type.y1 = rnd\_y;

rect\_type.Show(gr = CreateGraphics(), Color.Aquamarine);

}

else if (item is rhomb rh\_type)

{

rh\_type.points[0] = new Point(rnd\_x, rnd\_y - rh\_type.A/2);

rh\_type.points[2] = new Point(rnd\_x, rnd\_y + rh\_type.A/2);

rh\_type.points[1] = new Point(rnd\_x + rh\_type.B/2, rnd\_y);

rh\_type.points[3] = new Point(rnd\_x - rh\_type.B/2, rnd\_y);

rh\_type.Show(gr = CreateGraphics(), Color.Aquamarine);

}

else if (item is square square\_type)

{

square\_type.x1 = rnd\_x;

square\_type.y1 = rnd\_y;

square\_type.Show(gr = CreateGraphics(), Color.Aquamarine);

}

}

} // ПРИВЕДЕНИЕ К ОДНОЙ ТОЧКЕ

1. Метод для перемещения фигур

private void Form1\_KeyDown(object sender, KeyEventArgs e)

{

if (!button\_18\_is\_clicked)

{

MessageBox.Show("Вы еще не создали массив фигур!");

return;

}

gr.Clear(Color.White);

bool circ\_type\_is\_moving = false;

bool square\_type\_is\_moving = false;

if (checkedListBox1.GetItemChecked(0))

{

circ\_type\_is\_moving = true;

if (e.KeyValue == (char)Keys.Right)

{

foreach (object item in my\_objects)

{

if (item is circle circ\_type)

{

circ\_type.Move(move\_pixels, 0);

circ\_type.Show(gr = CreateGraphics(), Color.Aquamarine);

}

}

}

else if (e.KeyValue == (char)Keys.Left)

{

foreach (object item in my\_objects)

{

if (item is circle circ\_type)

{

circ\_type.Move(-move\_pixels, 0);

circ\_type.Show(gr = CreateGraphics(), Color.Aquamarine);

}

}

}

else if (e.KeyValue == (char)Keys.Up)

{

foreach (object item in my\_objects)

{

if (item is circle circ\_type)

{

circ\_type.Move(0, -move\_pixels);

circ\_type.Show(gr = CreateGraphics(), Color.Aquamarine);

}

}

}

else if (e.KeyValue == (char)Keys.Down)

{

foreach (object item in my\_objects)

{

if (item is circle circ\_type)

{

circ\_type.Move(0, move\_pixels);

circ\_type.Show(gr = CreateGraphics(), Color.Aquamarine);

}

}

}

}

if (checkedListBox1.GetItemChecked(1))

{

square\_type\_is\_moving = true;

foreach (object item in my\_objects)

{

if (item is ellipse ell\_type)

{

ell\_type.ellipse\_is\_visible = true;

ell\_type.Show(gr = CreateGraphics(), Color.Aquamarine);

}

else if(item is circle circ\_type)

{

circ\_type.circle\_is\_visible = true;

circ\_type.Show(gr = CreateGraphics(), Color.Aquamarine);

}

}

if (e.KeyValue == (char)Keys.Right)

{

foreach (object item in my\_objects)

{

if (item is square square\_type)

{

square\_type.Move(move\_pixels, 0);

square\_type.Show(gr = CreateGraphics(), Color.Aquamarine);

}

}

}

else if (e.KeyValue == (char)Keys.Left)

{

foreach (object item in my\_objects)

{

if (item is square square\_type)

{

square\_type.Move(-move\_pixels, 0);

square\_type.Show(gr = CreateGraphics(), Color.Aquamarine);

}

}

}

else if (e.KeyValue == (char)Keys.Up)

{

foreach (object item in my\_objects)

{

if (item is square square\_type)

{

square\_type.Move(0, -move\_pixels);

square\_type.Show(gr = CreateGraphics(), Color.Aquamarine);

}

}

}

else if (e.KeyValue == (char)Keys.Down)

{

foreach (object item in my\_objects)

{

if (item is square square\_type)

{

square\_type.Move(0, move\_pixels);

square\_type.Show(gr = CreateGraphics(), Color.Aquamarine);

}

}

}

}

if(!circ\_type\_is\_moving)

{

foreach (object item in my\_objects)

{

if (item is ellipse ell\_type)

{

ell\_type.ellipse\_is\_visible = true;

ell\_type.Show(gr = CreateGraphics(), Color.Aquamarine);

}

else if (item is circle circ\_type)

{

circ\_type.circle\_is\_visible = true;

circ\_type.Show(gr = CreateGraphics(), Color.Aquamarine);

}

}

}

if(!square\_type\_is\_moving)

{

foreach (object item in my\_objects)

{

if (item is rhomb rh\_type)

{

rh\_type.rhomb\_is\_visible = true;

rh\_type.Show(gr = CreateGraphics(), Color.Aquamarine);

}

else if (item is rectangle rect\_type)

{

rect\_type.rectangle\_is\_visible = true;

rect\_type.Show(gr = CreateGraphics(), Color.Aquamarine);

}

else if (item is square square\_type)

{

square\_type.square\_is\_visible = true;

square\_type.Show(gr = CreateGraphics(), Color.Aquamarine);

}

}

}

} // ПЕРЕМЕЩЕНИЕ

1. Опустошение массива

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

{

button\_18\_is\_clicked = false;

gr.Clear(Color.White);

for (int i = 0; i < num\_of\_variant; i++)

{

my\_objects[i] = null;

}

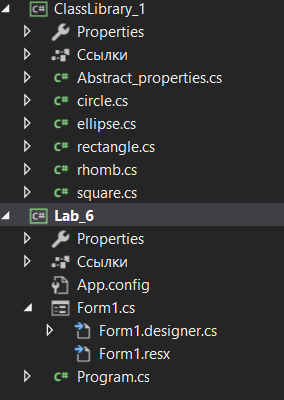
MessageBox.Show("Массив очищен!");

} // ОПУСТОШЕНИЕ МАССИВА

}

}

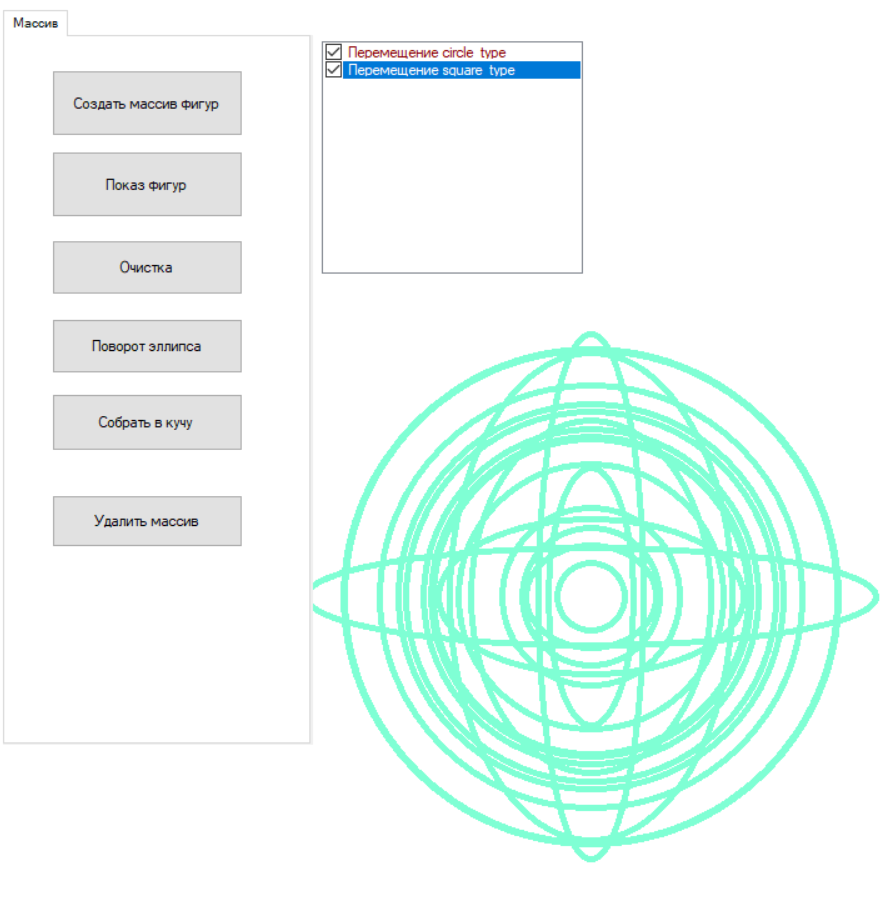
Структура программы:



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:

Имеется вкладка «массив» для его создания и последующей работы с ним. Для начала вам следует создать массив, ибо, не имея массив, вы не сможете применить предоставленный функционал. После создания массива фигур, вы можете показать фигуры, очистить холст, переместить фигуры ( для перемещения отдельно взятого типа фигур перейдите в CheckBox), уничтожить массив, переместить все фигуры в одну точку.



Листинг программы:

public partial class Form1 : Form // пытаюсь все испортить

{

static int num\_of\_variant = 27;

Graphics gr;

Random my\_rnd = new Random();

int move\_pixels = 26;

public Form1()

{

InitializeComponent();

this.WindowState = FormWindowState.Maximized;

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

}

Abstract\_properties[] my\_objects = new Abstract\_properties[num\_of\_variant];

bool button\_18\_is\_clicked = false;

private void button18\_Click(object sender, EventArgs e)

{

if (button\_18\_is\_clicked)

{

MessageBox.Show("Вы уже создали массив фигур!");

return;

}

int my\_rand = my\_rnd.Next(1, 50);

Thread.Sleep(10);

for (int i = 0; i < num\_of\_variant; i++) // добавление элементов в массив

{

if (my\_rand >= 1 && my\_rand < 11)

{

my\_objects[i] = new circle();

}

else if (my\_rand >= 11 && my\_rand < 21)

{

my\_objects[i] = new ellipse();

}

else if (my\_rand >= 21 && my\_rand < 31)

{

my\_objects[i] = new rectangle();

}

else if (my\_rand >= 31 && my\_rand < 41)

{

my\_objects[i] = new square();

}

else if (my\_rand >= 41 && my\_rand < 51)

{

my\_objects[i] = new rhomb();

}

my\_rand = my\_rnd.Next(1, 50);

Thread.Sleep(10);

}

button\_18\_is\_clicked = true;

MessageBox.Show("Массив фигур создан!");

} // СОЗДАЕМ МАССИВ

private void button19\_Click(object sender, EventArgs e)

{

if (!button\_18\_is\_clicked)

{

MessageBox.Show("Вы еще не создали массив фигур!");

return;

}

foreach (object item in my\_objects) // показ фигур

{

if (item is circle circ\_type)

{

circ\_type.circle\_is\_visible = true;

circ\_type.Show(gr = CreateGraphics(), Color.Aquamarine);

}

else if (item is ellipse ell\_type)

{

ell\_type.ellipse\_is\_visible = true;

ell\_type.Show(gr = CreateGraphics(), Color.Aquamarine);

}

else if (item is rectangle rect\_type)

{

rect\_type.rectangle\_is\_visible = true;

rect\_type.Show(gr = CreateGraphics(), Color.Aquamarine);

}

else if (item is rhomb rh\_type)

{

rh\_type.rhomb\_is\_visible = true;

rh\_type.Show(gr = CreateGraphics(), Color.Aquamarine);

}

else if (item is square square\_type)

{

square\_type.square\_is\_visible = true;

square\_type.Show(gr = CreateGraphics(), Color.Aquamarine);

}

}

} // ПОКАЗ ФИГУР

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

{

if (!button\_18\_is\_clicked)

{

MessageBox.Show("Вы еще не создали массив фигур!");

return;

}

gr.Clear(Color.White);

foreach (object item in my\_objects) // очистка фигур с холста

{

if (item is ellipse ell\_type)

{

ell\_type.Delete();

}

else if(item is circle circ\_type)

{

circ\_type.Delete();

}

else if (item is rectangle rect\_type)

{

rect\_type.Delete();

}

else if (item is rhomb rh\_type)

{

rh\_type.Delete();

}

else if (item is square square\_type)

{

square\_type.Delete();

}

}

}// ОЧИСТКА

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

{

if (!button\_18\_is\_clicked)

{

MessageBox.Show("Вы еще не создали массив фигур!");

return;

}

gr.Clear(Color.White);

foreach (object item in my\_objects) // поиск эллипсов и их поворот

{

if (item is ellipse ell\_type)

{

ell\_type.rotate\_for\_90();

ell\_type.Show(gr = CreateGraphics(), Color.Aquamarine);

}

else if (item is circle circ\_type)

{

circ\_type.Show(gr = CreateGraphics(), Color.Aquamarine);

}

else if (item is rectangle rect\_type)

{

rect\_type.Show(gr = CreateGraphics(), Color.Aquamarine);

}

else if (item is rhomb rh\_type)

{

rh\_type.Show(gr = CreateGraphics(), Color.Aquamarine);

}

else if (item is square square\_type)

{

square\_type.Show(gr = CreateGraphics(), Color.Aquamarine);

}

}

}// ПОВОРОТ ЭЛЛИПСА

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

{

Random rnd = new Random();

int rnd\_x = rnd.Next(300, 1200);

int rnd\_y = rnd.Next(300, 600);

if (!button\_18\_is\_clicked)

{

MessageBox.Show("Вы еще не создали массив фигур!");

return;

}

gr.Clear(Color.White);

foreach (object item in my\_objects) // приведение элементов массива к одной точке

{

if (item is ellipse ell\_type)

{

ell\_type.x1 = rnd\_x;

ell\_type.y1 = rnd\_y;

ell\_type.Show(gr = CreateGraphics(), Color.Aquamarine);

}

else if (item is circle circ\_type)

{

circ\_type.x1 = rnd\_x;

circ\_type.y1 = rnd\_y;

circ\_type.Show(gr = CreateGraphics(), Color.Aquamarine);

}

else if (item is rectangle rect\_type)

{

rect\_type.x1 = rnd\_x;

rect\_type.y1 = rnd\_y;

rect\_type.Show(gr = CreateGraphics(), Color.Aquamarine);

}

else if (item is rhomb rh\_type)

{

rh\_type.points[0] = new Point(rnd\_x, rnd\_y - rh\_type.A/2);

rh\_type.points[2] = new Point(rnd\_x, rnd\_y + rh\_type.A/2);

rh\_type.points[1] = new Point(rnd\_x + rh\_type.B/2, rnd\_y);

rh\_type.points[3] = new Point(rnd\_x - rh\_type.B/2, rnd\_y);

rh\_type.Show(gr = CreateGraphics(), Color.Aquamarine);

}

else if (item is square square\_type)

{

square\_type.x1 = rnd\_x;

square\_type.y1 = rnd\_y;

square\_type.Show(gr = CreateGraphics(), Color.Aquamarine);

}

}

} // ПРИВЕДЕНИЕ К ОДНОЙ ТОЧКЕ

private void Form1\_KeyDown(object sender, KeyEventArgs e)

{

if (!button\_18\_is\_clicked)

{

MessageBox.Show("Вы еще не создали массив фигур!");

return;

}

gr.Clear(Color.White);

bool circ\_type\_is\_moving = false;

bool square\_type\_is\_moving = false;

if (checkedListBox1.GetItemChecked(0))

{

circ\_type\_is\_moving = true;

if (e.KeyValue == (char)Keys.Right)

{

foreach (object item in my\_objects) // перемещение элементов массива

{

if (item is circle circ\_type)

{

circ\_type.Move(move\_pixels, 0);

circ\_type.Show(gr = CreateGraphics(), Color.Aquamarine);

}

}

}

else if (e.KeyValue == (char)Keys.Left)

{

foreach (object item in my\_objects)

{

if (item is circle circ\_type)

{

circ\_type.Move(-move\_pixels, 0);

circ\_type.Show(gr = CreateGraphics(), Color.Aquamarine);

}

}

}

else if (e.KeyValue == (char)Keys.Up)

{

foreach (object item in my\_objects)

{

if (item is circle circ\_type)

{

circ\_type.Move(0, -move\_pixels);

circ\_type.Show(gr = CreateGraphics(), Color.Aquamarine);

}

}

}

else if (e.KeyValue == (char)Keys.Down)

{

foreach (object item in my\_objects)

{

if (item is circle circ\_type)

{

circ\_type.Move(0, move\_pixels);

circ\_type.Show(gr = CreateGraphics(), Color.Aquamarine);

}

}

}

}

if (checkedListBox1.GetItemChecked(1))

{

square\_type\_is\_moving = true;

foreach (object item in my\_objects)

{

if (item is ellipse ell\_type)

{

ell\_type.ellipse\_is\_visible = true;

ell\_type.Show(gr = CreateGraphics(), Color.Aquamarine);

}

else if(item is circle circ\_type)

{

circ\_type.circle\_is\_visible = true;

circ\_type.Show(gr = CreateGraphics(), Color.Aquamarine);

}

}

if (e.KeyValue == (char)Keys.Right)

{

foreach (object item in my\_objects)

{

if (item is square square\_type)

{

square\_type.Move(move\_pixels, 0);

square\_type.Show(gr = CreateGraphics(), Color.Aquamarine);

}

}

}

else if (e.KeyValue == (char)Keys.Left)

{

foreach (object item in my\_objects)

{

if (item is square square\_type)

{

square\_type.Move(-move\_pixels, 0);

square\_type.Show(gr = CreateGraphics(), Color.Aquamarine);

}

}

}

else if (e.KeyValue == (char)Keys.Up)

{

foreach (object item in my\_objects)

{

if (item is square square\_type)

{

square\_type.Move(0, -move\_pixels);

square\_type.Show(gr = CreateGraphics(), Color.Aquamarine);

}

}

}

else if (e.KeyValue == (char)Keys.Down)

{

foreach (object item in my\_objects)

{

if (item is square square\_type)

{

square\_type.Move(0, move\_pixels);

square\_type.Show(gr = CreateGraphics(), Color.Aquamarine);

}

}

}

}

if(!circ\_type\_is\_moving)

{

foreach (object item in my\_objects)

{

if (item is ellipse ell\_type)

{

ell\_type.ellipse\_is\_visible = true;

ell\_type.Show(gr = CreateGraphics(), Color.Aquamarine);

}

else if (item is circle circ\_type)

{

circ\_type.circle\_is\_visible = true;

circ\_type.Show(gr = CreateGraphics(), Color.Aquamarine);

}

}

}

if(!square\_type\_is\_moving)

{

foreach (object item in my\_objects)

{

if (item is rhomb rh\_type)

{

rh\_type.rhomb\_is\_visible = true;

rh\_type.Show(gr = CreateGraphics(), Color.Aquamarine);

}

else if (item is rectangle rect\_type)

{

rect\_type.rectangle\_is\_visible = true;

rect\_type.Show(gr = CreateGraphics(), Color.Aquamarine);

}

else if (item is square square\_type)

{

square\_type.square\_is\_visible = true;

square\_type.Show(gr = CreateGraphics(), Color.Aquamarine);

}

}

}

} // ПЕРЕМЕЩЕНИЕ

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

{

button\_18\_is\_clicked = false;

gr.Clear(Color.White);

for (int i = 0; i < num\_of\_variant; i++) // обнуление элементов массива

{

my\_objects[i] = null;

}

MessageBox.Show("Массив очищен!");

} // ОПУСТОШЕНИЕ МАССИВА

}