**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 № 2

по дисциплине «Объектно-ориентированное программирование»

**Выполнил**

студент группы 4210

Нигамадянов Фанис Магефурович

Казань

2021

**Лабораторная работа №2.** **Создание и использование отдельных классов для графических примитивов**

Описание класса circle:

class square

{

int lx\_edge = 430;

int ly\_edge = 38;

int rx\_edge = 1100;

int ry\_edge = 750;

Random rnd = new Random();

Graphics gr;

public int x1 { get; set; }

public int y1 { get; set; }

public int a {get; set;}

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

public square()

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

public void show(Graphics gc, Color color)

public void to\_double()

public void MoveToRight(int x)

public void MoveToLeft(int x)

public void MoveToUp(int y)

public void MoveToDown(int y)

}

Описание свойств и полей

1. ограничение поля случайности

int lx\_edge = 430;

int ly\_edge = 38;

int rx\_edge = 1100;

int ry\_edge = 750;

1. объект класса рандом

Random rnd = new Random();

1. Объект класса Graphics

Graphics gr;

1. Поля координа

public int x1 { get; set; }

public int y1 { get; set; }

public int x2 { get; set; }

public int y2 { get; set; }

1. условие на видимость

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

1. Конструктор по умолчанию

public circle()

{

this.x1 = rnd.Next(lx\_edge, rx\_edge-100);

this.x2 = rnd.Next(50, 400);

this.y1 = rnd.Next(ly\_edge, ry\_edge-100);

this.y2 = x2;

this.circle\_is\_visible = true;

}

1. Конструктор для создания круга по умолчанию

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

{

this.x1 = x1;

this.x2 = x2;

this.y1 = y1;

this.y2 = x2;

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

}

1. Метод для показа фигуры

public void show(Graphics gc, Color color)

{

if (circle\_is\_visible == true)

{

Pen pen = new Pen(color, 5);

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

}

}

1. Метод для перемещения

public void MoveTo(int x, int y)

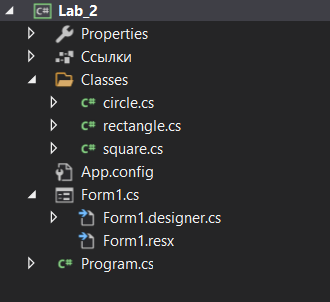
{

this.x += x;

this.y += y;

}

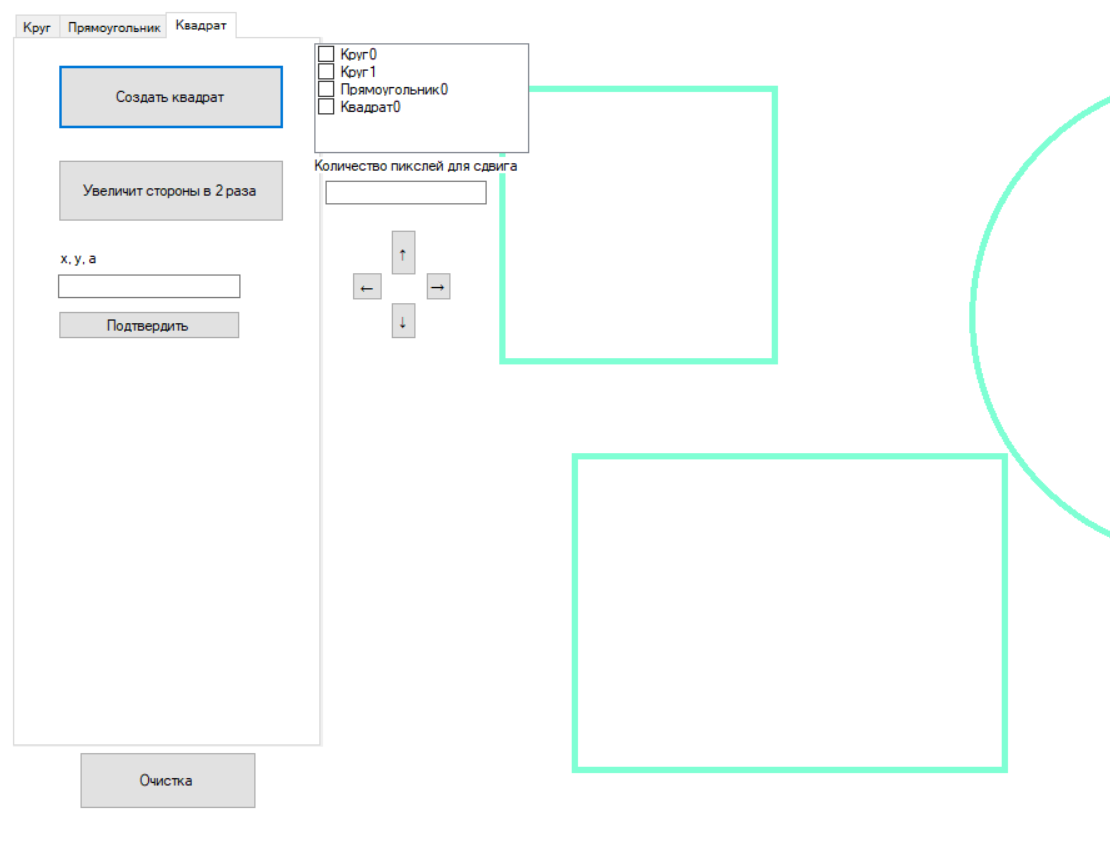
Структура программы:



1. Папка классов, где находятся три фигуры
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 три вкладки 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 нажимайте на соответствующие кнопки (стрелки для перемещения, кнопку очистки и т.д.)



Листинг программы:

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.Threading;

using System.Collections;

namespace MyLab1

{

public partial class Form1 : Form

{

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 move\_pixels = 10;

int for\_circ = 0;

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

int for\_square = 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;

Graphics gr;

static int num\_of\_circles = 20;

static int num\_of\_rectangles = 20;

static int num\_of\_squares = 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];

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\_R, user\_circ\_y - user\_circ\_R,

2\*user\_circ\_R, 2\*user\_circ\_R, true);

}

else

{

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

}

m\_circles[for\_circ].show(gr = CreateGraphics(), Color.Aquamarine);

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);

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

}

else

{

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

}

m\_squares[for\_square].show(gr = CreateGraphics(), Color.Aquamarine);

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 button4\_Click(object sender, EventArgs e)

{

for (int k = 0; k < checkedListBox1.Items.Count; k++)

{

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, .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);

}

i++;

}

if (counter == 0)

{

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

}

if (!square\_exist)

{

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

}

} // удвоение длины квадрата

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

{

for (int k = 0; k < checkedListBox1.Items.Count; k++)

{

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);

}

i++;

}

} // очистка

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

{

for (int k = 0; k < checkedListBox1.Items.Count; k++)

{

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", ""))].MoveToRight(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", ""))].MoveToRight(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", ""))].MoveToRight(move\_pixels);

}

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

}

i++;

}

} // сдвиг вправо

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

{

for (int k = 0; k < checkedListBox1.Items.Count; k++), 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", ""))].MoveToLeft(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", ""))].MoveToLeft(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", ""))].MoveToLeft(move\_pixels);

}

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

}

i++;

}

} // сдвиг влево

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

{

for (int k = 0; k < checkedListBox1.Items.Count; k++)

{

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", ""))].MoveToDown(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", ""))].MoveToDown(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", ""))].MoveToDown(move\_pixels);

}

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

}

i++;

}

} // сдвиг вниз

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

{

for (int k = 0; k < checkedListBox1.Items.Count; k++)

{

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", ""))].MoveToUp(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", ""))].MoveToUp(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", ""))].MoveToUp(move\_pixels);

}

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

}

i++;

}

} // сдвиг вверх

private void textBox1\_TextChanged(object sender, EventArgs e)

{

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)

{

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");

}

}

}

}