Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное

образовательное учреждение высшего образования

«Уфимский университет науки и технологий»

Факультет информатики и робототехники

Кафедра вычислительной математики и кибернетики

Лабораторная работа №4.1

“Круги на форме”

По дисциплине “Объектно-ориентированное программирование”

Выполнил:

Студент группы ПРО-232Б

Степанов К.А.

Проверил:

доцент кафедры ВМиК

Котельников В.А.

Уфа-2023

**Цель работы:**

Разработать приложение, позволяющее создавать динамически круги на форме, а также взаимодействовать с ними.

**Задание:**

* Создать простейшее приложение с GUI, содержащее:
* определение простейшего класса CCircle с координатами и постоянным

радиусом;

* форму с объектом для рисования (например, PaintBox);
* два элемента checkbox для управления поведением
* Реализовать следующее поведение:
* при нажатии мышкой на форме создается новый объект CCircle с

координатами нажатия и помещается в контейнер;

* при событии Paint должны отрисовываться на форме все объекты из

контейнера

* Реализовать следующее поведение с выделением объектов:
* хотя бы один объект из существующих на форме всегда является выделенным и отрисовывается отлично от других объектов;
* выделенными могут быть несколько объектов;
* выделение происходит при нажатии ЛКМ (левой клавиши мыши) на объект

CCircle на форме; при нажатии на область формы, где пересекаются несколько кругов, могут выделяться они все или какой-то один из них

* при нажатии на кнопку Del, все выделенные объекты должны удаляться
* при выделении объекта с помощью ЛКМ и удерживаемой клавиши Ctrl,

выделенными становятся несколько объектов

* несколько изменяемых пользователем флагов (элементы checkbox) управляют логикой поведения:
* checkbox для указания, работает ли клавиша Ctrl
* checkbox для указания, выделяется только один объект при нажатии на

их пересечение, или все, в которые попала мышка.

**Ход работы:**

При запуске программы создаётся окно со следующими элементами: “Ctrl” – checkBox1, ‘Выделять по одному кругу – checkBox2, “Delete” – Button1 (рис. 1).

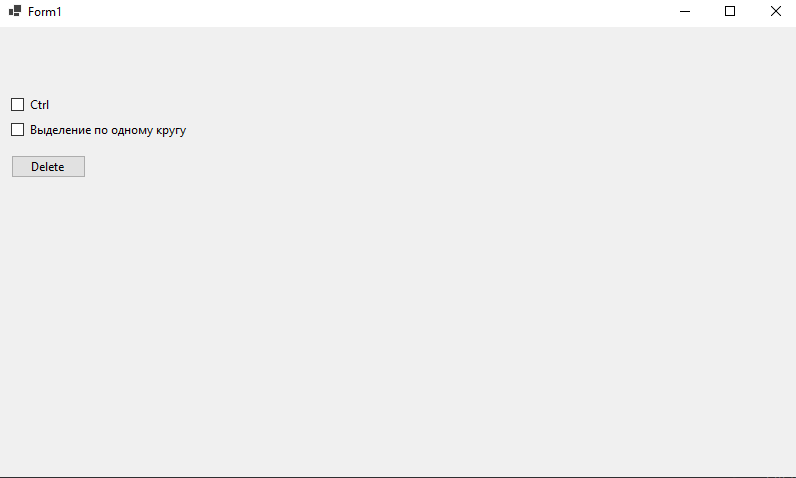


Рис. 1. – Внешний вид программы.

При нажатии левой кнопкой мыши на форму создается новый объект CCircle, который имеет вид круга (рис. 2).

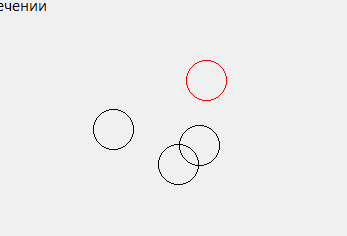


Рис 2 – результат нажатия ЛКМ.

При включенной клавише Control и выключенном флажке ‘Выделять один объект при пересечении’ – checkBox2 отключается возможность создавать новые круги, а при нажатии на пересечении кругов будут выделены все объекты, находящиеся в этом пересечении (рис. 3).

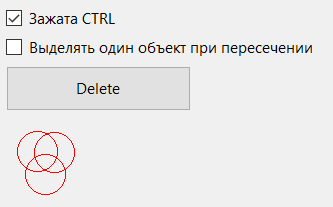


Рис 3 – Результат нажатия ЛКМ на область

пересечении кругов при зажатой Ctrl.

При зажатой клавише Ctrl и включенной ‘Выделять один объект при пересечении’ – checkBox2, круги будут выделяться по одному при каждом нажатии ЛКМ, при этом будет выделяться самый раннее созданный и не выделенный до этого (рис. 4).

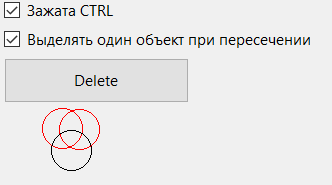


Рис 4 – результат работы с включенными

checkBox1 и checkBox2.

При нажатии на ‘Delete’ – Button1 все выделенные круги будут удалены с формы (рис. 5).

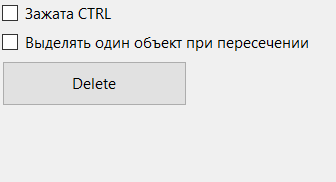


Рис 5 – результат работы при нажатии на

кнопку ‘Delete’ – Button1.

**Вывод:** в результате выполнения работы были получены навыки разработки связанные с динамическим созданием графических объектов на WinForms, а также изучен принцип Инкапсуляции Объектно-ориентированной парадигмы в программировании. Репозиторий работы на GitHub: [KittenKiller29/labaoop4\_1 (github.com)](https://github.com/KittenKiller29/labaoop4_1).

**Приложение 1**

**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;

namespace labaoop4\_1

{

public partial class Form1 : Form

{

private List<CCircle> FCircles = new List<CCircle>();

private int ctrl = 0;

public Form1()

{

InitializeComponent();

}

private void Form1\_Load(object sender, EventArgs e)

{

}

private void Form1\_Paint(object sender, PaintEventArgs e)//описание события Paint

{

foreach (CCircle Circle in FCircles)

{

Circle.drawCircle(e.Graphics);//Рисует все круги из списка

}

}

private void Form1\_MouseClick(object sender, MouseEventArgs e)//описание события нажатия мыши

{

if (ctrl == 0)//не нажат ctrl

{

foreach (CCircle Circle1 in FCircles)

{

Circle1.setColor("Black");//снимает выделение со всех объектов

}

CCircle Circle = new CCircle(e.X, e.Y, 20);//создает новый объект с выделением

FCircles.Add(Circle);

}

if (ctrl == 1)//нажат ctrl

{

foreach (CCircle Circle1 in FCircles)

{

if (Circle1.checkCircle(e) == true && checkBox2.Checked==true)//проверка на массовое выделение

{

break;

}

}

Refresh();

}

Refresh();

}

private void button1\_Click(object sender, EventArgs e)//описание события нажатия на кнопку Delete

{

for(int i = 0; i < FCircles.Count(); i++)

{

if (FCircles[i].getColor() == "Red")//проверка выделения объектов

{

FCircles.RemoveAt(i);//удаление выделенных объектов

i--;

}

}

Refresh();

}

private void Form1\_KeyDown(object sender, KeyEventArgs e)//описание события нажатия клавиши на клавиатуре

{

if (ModifierKeys==Keys.Control)//если нажатая клавиша Control

{

checkBox1.Checked = !checkBox1.Checked;//устанавливает флаг в чекбокс в определенное значение

}

switch (ctrl)//в зависимости от состояния флага

{

case 0:

ctrl++;

foreach (CCircle Circle1 in FCircles)

{

Circle1.setCtrl(true);//если нажат Control устанавливает определенное значение

}

break;

case 1:

ctrl = 0;

foreach (CCircle Circle1 in FCircles)

{

Circle1.setCtrl(false);//если отжат Control устанавливает определенное значение

}

break;

}

}

private void checkBox2\_CheckedChanged(object sender, EventArgs e)

{

}

private void Form1\_MouseEnter(object sender, EventArgs e)

{

}

private void Form1\_MouseUp(object sender, MouseEventArgs e)

{

}

}

public class CCircle//описание объекта круга

{

private int x, y, rad;//координаты и радиус

private string color = "Red";//цвет выделения(изначальный)

private bool ctrl=false;

public CCircle(int xp, int yp, int radp) //конструктор с параметрами

{

x = xp;

y = yp;

rad = radp;

}

public void drawCircle(Graphics Canvas)//метод отрисовки круга

{

if (color == "Red")

{

Canvas.DrawEllipse(new Pen(Color.Red), x - rad, y - rad, rad \* 2, rad \* 2);

}

else

{

Canvas.DrawEllipse(new Pen(Color.Black), x - rad, y - rad, rad \* 2, rad \* 2);

}

}

public void setColor(string Color)//сеттер цвета круга

{

color = Color;

}

public string getColor()//геттер цвета круга

{

return color;

}

public bool checkCircle(MouseEventArgs e)//проверка на попадание курсора мыши во внутрь круга

{

if (ctrl) {

if (Math.Pow(e.X - x, 2) + Math.Pow(e.Y - y, 2) <= Math.Pow(rad, 2) && color!="Red")

{

color = "Red";

return true;

}

}

return false;

}

public void setCtrl(bool a)//сеттер флага выделения

{

ctrl = a;

}

}

}