

Федеральное государственное бюджетное образовательное  
учреждение «Сибирский государственный университет  
телекоммуникаций и информатики»

кафедра прикладной математики и кибернетики

**Лабораторная работа №5 по  
Визуальному программированию**

Тема: «Перетаскивание графических объектов с помощью мыши»

**Выполнили:** студенты 2 курса  
Зырянов Константин  
Ярыгин Максим

**Проверил:**  
Бублей Дмитрий

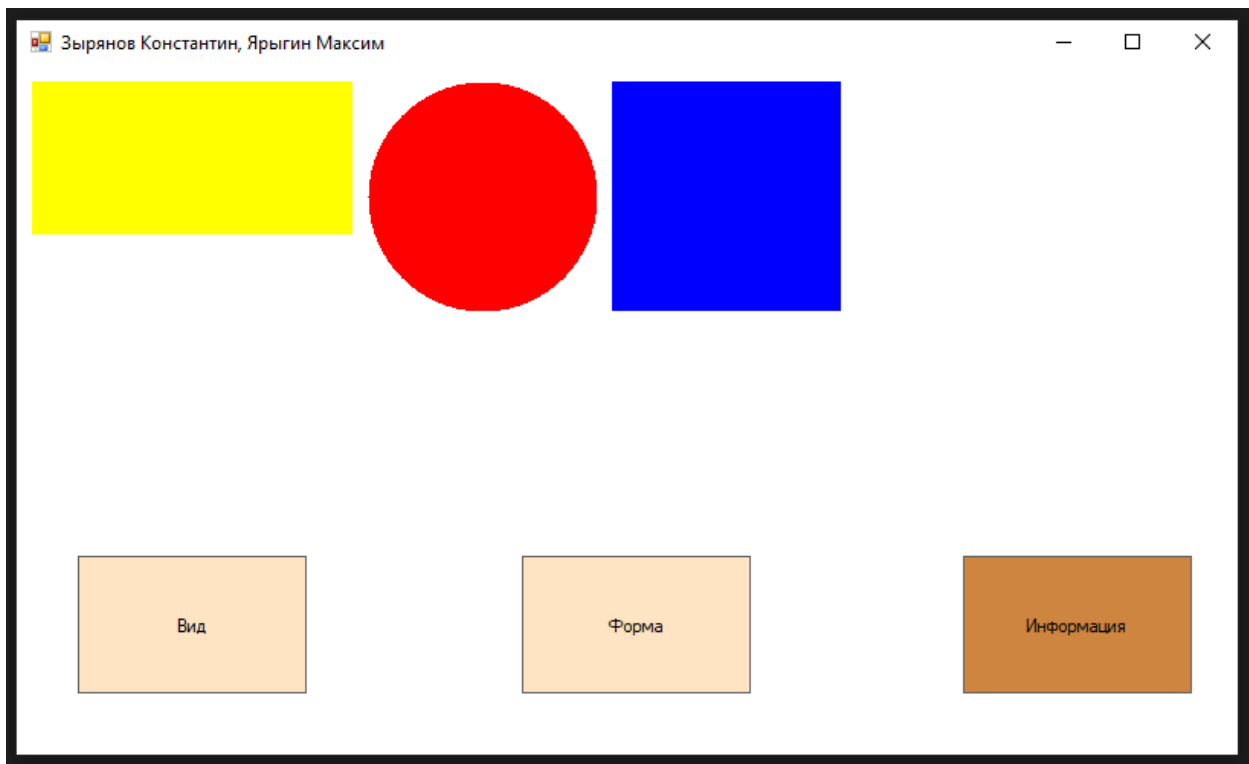
Новосибирск 2021г.

# 1. Цель работы

- 1) На форме, в границах PictureBox, должны быть созданы три графических объекта: квадрат, эллипс и прямоугольник.
- 2) Реализована возможность перетаскивания этих объектов по форме.
- 3) Должно быть выполнено следующее условие: если один из объектов, будет перенесён на Label «Вид», то в третьем Label «Информация», должны появиться данные о том какого цвета и что это за фигура.
- 4) При перетаскивании окружности или квадрата на Label «Форма», объект должен меняться на противоположный (т.е. если перенесли квадрат, то он становится окружностью, а окружность, в свою очередь, становится квадратом и наоборот)

# 2. Выполнение работы

## Созданная форма в MS Visual Studio



## Код программы

```
13 public partial class Form1 : Form
14 {
15     bool DraggingR = false;
16     bool DraggingC = false;
17     bool DraggingS = false;
18     int X, Y, dx, dy;
19     int LastClicked = 0;
20     Rectangle Rectangle = new Rectangle(10, 10, 210, 100);
21     Rectangle Circle = new Rectangle(230, 10, 150, 150);
22     Rectangle Square = new Rectangle(390, 10, 150, 150);
23     int RectangleX = 0;
24     int RectangleY = 0;
25     int CircleX = 0;
26     int CircleY = 0;
27     int SquareX = 0;
28     int SquareY = 0;
29
30     public Form1()
31     {
32         InitializeComponent();
33     }
34
35     private void pictureBox1_MouseDown(object sender, MouseEventArgs e)
36     {
37         if ((e.X < Rectangle.X + Rectangle.Width) && (e.X > Rectangle.X))
38         {
39             if ((e.Y < Rectangle.Y + Rectangle.Height) && (e.Y > Rectangle.Y))
40             {
41                 DraggingR = true;
42                 RectangleX = e.X - Rectangle.X;
43                 RectangleY = e.Y - Rectangle.Y;
44                 LastClicked = 1;
45                 return;
46             }
47         }
48         if ((e.X < Circle.X + Circle.Width) && (e.X > Circle.X))
49         {
50             if ((e.Y < Circle.Y + Circle.Height) && (e.Y > Circle.Y))
51             {
52                 DraggingC = true;
53                 CircleX = e.X - Circle.X;
54                 CircleY = e.Y - Circle.Y;
55                 LastClicked = 2;
56                 return;
57             }
58         }
59         if ((e.X < Square.X + Square.Width) && (e.X > Square.X))
60         {
61             if ((e.Y < Square.Y + Square.Height) && (e.Y > Square.Y))
62             {
63                 DraggingS = true;
64                 SquareX = e.X - Square.X;
65                 SquareY = e.Y - Square.Y;
66                 LastClicked = 3;
67                 return;
68             }
69         }
70     }
```

```
71
72     private void pictureBox1_Paint(object sender, PaintEventArgs e)
73     {
74         e.Graphics.FillRectangle(Brushes.Blue, Square);
75         e.Graphics.FillEllipse(Brushes.Red, Circle);
76         e.Graphics.FillRectangle(Brushes.Yellow, Rectangle);
77     }
78
79     private void pictureBox1_MouseUp(object sender, MouseEventArgs e)
80     {
81         DraggingR = false;
82         DraggingC = false;
83         DraggingS = false;
84     }
85
86     private void pictureBox1_MouseMove(object sender, MouseEventArgs e)
87     {
88         label3.Text = "Информация";
89         if (DraggingR)
90         {
91             Rectangle.X = e.X - RectangleX;
92             Rectangle.Y = e.Y - RectangleY;
93         }
94         if (DraggingC)
95         {
96             Circle.X = e.X - CircleX;
97             Circle.Y = e.Y - CircleY;
98         }
99         if (DraggingS)
100         {
101             Square.X = e.X - SquareX;
102             Square.Y = e.Y - SquareY;
103         }
104         if ((label1.Location.X < Rectangle.X + Rectangle.Width) && (label1.Location.X + label1.Width > Rectangle.X))
105         {
106             if ((label1.Location.Y < Rectangle.Y + Rectangle.Height) && (label1.Location.Y + label1.Height > Rectangle.Y))
107             {
108                 if (label3.Text == "Информация")
109                 {
110                     label3.Text = "Жёлтый прямоугольник";
111                 }
112                 else
113                     label3.Text = string.Concat(label3.Text, "\nЖёлтый прямоугольник");
114             }
115         }
116         if ((label1.Location.X < Circle.X + Circle.Width) && (label1.Location.X + label1.Width > Circle.X))
117         {
118             if ((label1.Location.Y < Circle.Y + Circle.Height) && (label1.Location.Y + label1.Height > Circle.Y))
119             {
120                 if (label3.Text == "Информация")
121                 {
122                     label3.Text = "Красный круг";
123                 }
124                 else
125                     label3.Text = string.Concat(label3.Text, "\nКрасный круг");
126             }
127         }
```

```

128 if ((label1.Location.X < Square.X + Square.Width) && (label1.Location.X + label1.Width > Square.X))
129 {
130     if ((label1.Location.Y < Square.Y + Square.Height) && (label1.Location.Y + label1.Height > Square.Y))
131     {
132         if (label3.Text == "Информация")
133         {
134             label3.Text = "Синий квадрат";
135         }
136         else
137             label3.Text = string.Concat(label3.Text, "\nСиний квадрат");
138     }
139 }
140 if (LastClicked == 1)
141 {
142     if ((label2.Location.X < Rectangle.X + Rectangle.Width) && (label2.Location.X + label2.Width > Rectangle.X))
143     {
144         if ((label2.Location.Y < Rectangle.Y + Rectangle.Height) && (label2.Location.Y + label2.Height > Rectangle.Y))
145         {
146             X = Rectangle.X;
147             Y = Rectangle.Y;
148             dx = RectangleX;
149             dy = RectangleY;
150             Rectangle.X = Circle.X;
151             Rectangle.Y = Circle.Y;
152             RectangleX = CircleX;
153             RectangleY = CircleY;
154             Circle.X = X;
155             Circle.Y = Y;
156             CircleX = dx;
157             CircleY = dy;
158             DraggingR = false;
159             DraggingC = true;
160             LastClicked = 0;
161         }
162     }
163 }
164 if (LastClicked == 2)
165 {
166     if ((label2.Location.X < Circle.X + Circle.Width) && (label2.Location.X + label2.Width > Circle.X))
167     {
168         if ((label2.Location.Y < Circle.Y + Circle.Height) && (label2.Location.Y + label2.Height > Circle.Y))
169         {
170             X = Circle.X;
171             Y = Circle.Y;
172             dx = CircleX;
173             dy = CircleY;
174             Circle.X = Square.X;
175             Circle.Y = Square.Y;
176             CircleX = SquareX;
177             CircleY = SquareY;
178             Square.X = X;
179             Square.Y = Y;
180             SquareX = dx;
181             SquareY = dy;
182             DraggingC = false;
183             DraggingS = true;
184             LastClicked = 0;
185         }
186     }
187 }

```

```

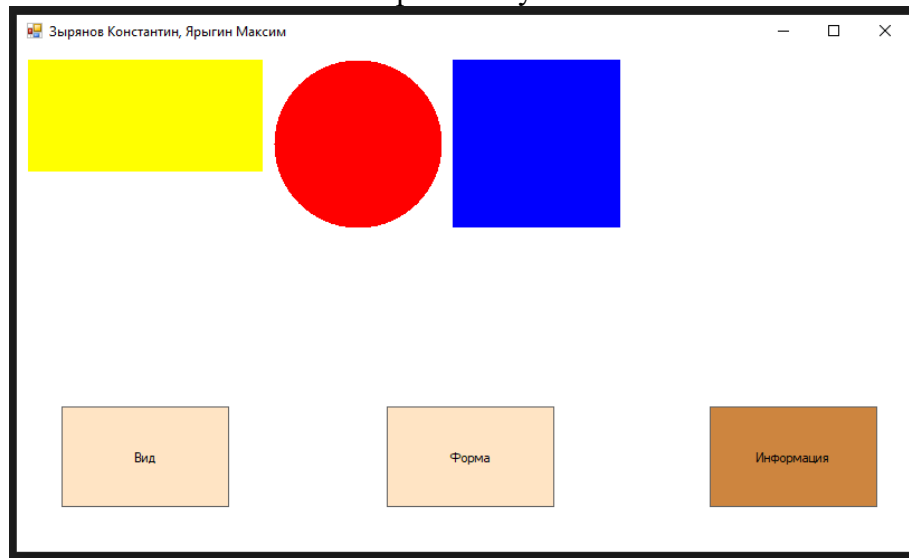
    if (LastClicked == 3)
    {
        if ((label2.Location.X < Square.X + Square.Width) && (label2.Location.X + label2.Width > Square.X))
        {
            if ((label2.Location.Y < Square.Y + Square.Height) && (label2.Location.Y + label2.Height > Square.Y))
            {
                X = Square.X;
                Y = Square.Y;
                dx = SquareX;
                dy = SquareY;
                Square.X = Rectangle.X;
                Square.Y = Rectangle.Y;
                SquareX = RectangleX;
                SquareY = RectangleY;
                Rectangle.X = X;
                Rectangle.Y = Y;
                RectangleX = dx;
                RectangleY = dy;
                DraggingS = false;
                DraggingR = true;
                LastClicked = 0;
            }
        }
    }
    pictureBox1.Invalidate();
}
}

```

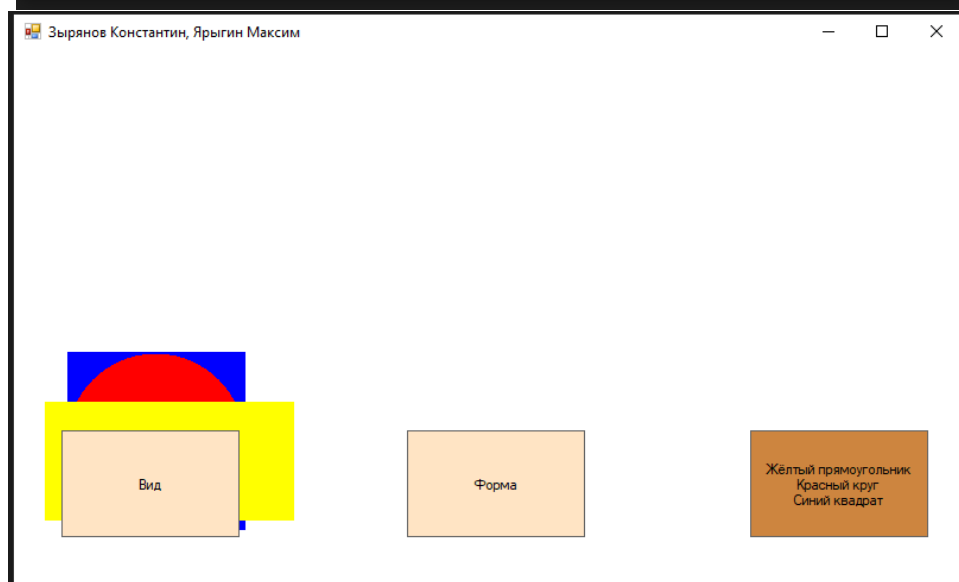
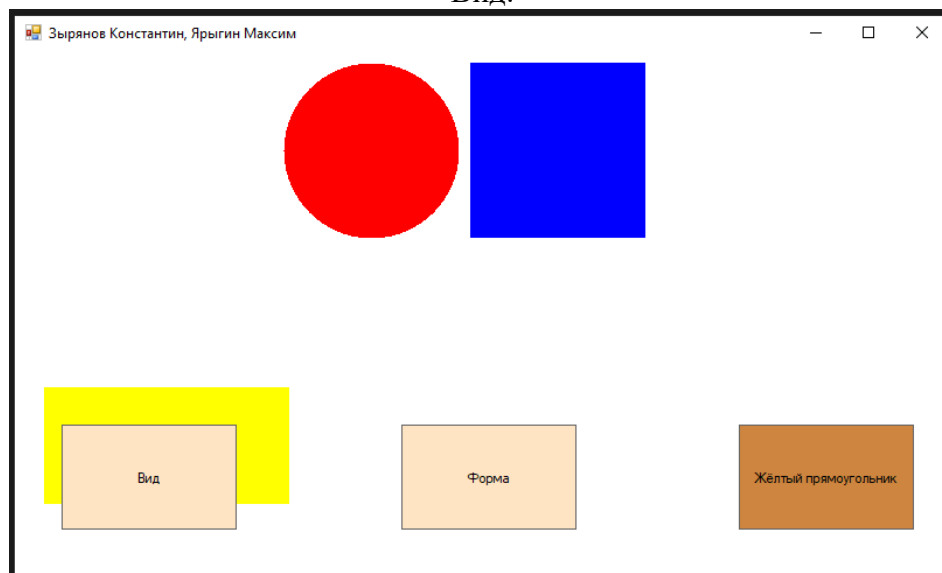
3.

## 4. Результат работы

Первый запуск:



Вид:



Форма:

