

Государственное автономное профессиональное образовательное учреждение  
Саратовской области «Балаковский политехнический техникум» ГАПОУ СО  
"БПТ"

## ОТЧЕТ ПО УЧЕБНОЙ ПРАКТИКЕ

Сроки прохождения практики: с 29.05. по 10.06г.

Профиль «09.02.07»»

Студент группы 16 Белоусов Денис Александрович

Руководитель практики Фолькина Надежда Артёмовна

Дата сдачи \_\_\_\_\_

Оценка \_\_\_\_\_

Балаково 2023

Вид практики: Учебная

Сроки прохождения практики:

с « 29 » 05.2023 г. по « 10 » 06.2023 г.

Цель практики:

Формирование навыков планирования и анализа собственной деятельности, умения продуктивного распределения времени.

В ходе практики студент решает следующие задачи:

1. Совершенствовать умения анализировать, планировать и реализовывать различные навыки профессиональной деятельности;
2. Совершенствовать аналитические, прогностические, проектировочные, коммуникативные, организаторские умения при работе, развивать профессионально значимые личностные качества;
3. Развивать профессиональную мотивацию, мобильность, креативность;
4. Формировать ответственное отношение к результату профессиональной деятельности.

Практика ориентирована на формирование следующих компетенций:

- способность применять и пополнять имеющиеся знания в процессе структурирования материалов, обеспечивающих образовательный процесс;
- умение изменять и использовать программный код по назначению
- способность оформлять по ГОСТу документы
- умение работать с документами
- способность к созданию многофункциональных форм и приложений при помощи кода и других свойств формы

База практики:

Государственное автономное профессиональное образовательное учреждение Саратовской области «Балаковский политехнический техникум» (ГАПОУ СО "БПТ")

Длительность пребывания на базе практики:

6 часов в один день, 6 дней в неделю.

### **Краткая характеристика учреждения**

Программа производственной практики осуществлялась на базе Государственного бюджетного образовательного учреждения среднего профессионального образования Саратовской области «Балаковский Политехнический техникум».

Государственное образовательное учреждение Саратовской области «Балаковский Политехнический техникум» - многопрофильное разноуровневое образовательное учреждение, в котором вы можете получить образование по самым разным профессиям начального профессионального и среднего профессионального образования, имеющее собственное лицо и пользующееся заслуженным уважением более 50 лет.

В данном образовательном учреждении работают высококвалифицированные педагоги, заслуженные учителя, отличники профессионально-технического образования, которые ведут предметы с использованием современных информационных технологий.

Учащиеся, поступившие на базе 9 классов, получают профессию и среднее (полное) общее образование. По окончании Политехникума выдаются дипломы, или свидетельства государственного образца.

## **Самоанализ деятельности**

В рамках учебной практики были отработаны навыки планирования и реализации профессиональной деятельности, отработаны навыки работы по научно-методическим материалам. Получен опыт разработки программного обеспечения и его анализ, разработки и реализации программной документации на программное обеспечение

В ходе прохождения учебной практики были реализованы следующие направления работы: аналитическое, научно-методическое, управленческое, а также социальные.

Аналитическое направление работы предполагало написание проектной документации на будущую разработку ПО.

Следующим было техническое направление работы. Некоторые затруднения возникли в процессе написания кода. Во многом это было связано с отсутствием необходимых практических навыков.

Для преодоления возникших затруднений и предупреждения их возникновения в будущем необходимо больше практиковать навыки различных направлений работы.

Практические работы с описанием :

### **1)Руководство пользователя**

Здесь я расписывал инструкцию и требования к созданию приложения, для того чтобы пользователю было понятно и мог приступить к работе с данным приложением.

## **Руководство пользователя**

### **Аннотация**

В данном документе представлено руководство пользователя для учебно-практической программы «Приложения для смешения красок».

Руководство пользователя разработано в соответствии с требованиями РД 50-34.698-90 «Информационная технология. Комплекс стандартов на автоматизированные системы. Автоматизированные системы Требования к содержанию документов», ГОСТ 2.105-95 «Единая система конструкторской документации. Общие требования к текстовым документам» и рекомендациями Департамента информационных технологий города Москвы. Все пользователи должны ознакомиться с данным руководством перед началом работы с Модулем.

## **1 ВВЕДЕНИЕ**

### **1.1 Область применения программы**

Учебно-практическая программа «Приложение для смешения красок» может быть использована для повышения автоматизации и эффективности работы пользователя, работающего с красками.

### **1.2 Описание возможностей**

Приложения для смешения красок имеет возможности

- Выбор значения для любого из 3 основных цветов в диапазоне от 0 до 255
- Настройка Альфа-канала
- Вывод результата на экран
- Закрывать программу

## **2 Условия, при соблюдении которых обеспечивается применение средства автоматизации в соответствии с назначением**

### **2.1 Требование к программным средствам**

Программа поддерживает работу с операционными системами Microsoft Windows (32-bit), Microsoft Windows (x64)

### **2.2 Требования к аппаратным средствам**

Оптимальными аппаратными требованиями считается следующие компоненты и выше

- Процессор intel core I9 или AMD Ryzen 9 3950x
- RTX 3080 или RX 6700XT
- Дисковое пространство не менее 1ГБ памяти
- Устройства ввода-вывода

## **3 Начало работы в приложении**

Для начала работы с «приложением для смешения красок», необходимо нажать на файл «Смешение красок.exe». После окончания загрузки программы можно перейти к работе с ней.

### **3.1 Работа с приложением**

Для смешения 3 цветов, красного, зеленого и синего необходимо потянуть за соответствующие ползунки на экране.

GREEN – устанавливает значение зеленого цвета в итоговом цвете

RED – устанавливает значение красного цвета в итоговом цвете

BLUE – устанавливает значение синего цвета в итоговом цвете

ALFA – устанавливает значение альфа-канала итога цвета

Что бы выполнить смешение красок по установленным значениям 3 цветовых параметров – необходимо нажать кнопку, расположенную в нижней части окна программы, под названием «Смешать». После

выполнения смешения, на панели в верхней части экрана появится итоговый цвет.

## 2) Печатная машинка(игра)

Я сделал печатную машинку, которая выводит буквы и пользователь должен, как можно быстрее нажать на клавишу этой буквы, в случае медлительного процесса на нажатие кнопки, игра будет окончена.

```
7 using System.Text;
8 using System.Threading.Tasks;
9 using System.Windows.Forms;
10
11 namespace Печатная_машинка
12 {
13     Ссылка 3
14     public partial class Form1 : Form
15     {
16         Random random = new Random();
17         Stats stats = new Stats();
18         Ссылка 1
19         public Form1()
20         {
21             InitializeComponent();
22             timer1.Enabled = true;
23             timer1.Start();
24         }
25         Ссылка 1
26         private void toolStripStatusLabel1_Click(object sender, EventArgs e)
27         {
28         }
29         Ссылка 1
30         private void timer1_Tick(object sender, EventArgs e)
31         {
32             listBox1.Items.Add((Keys)random.Next(65, 90));
33             if (listBox1.Items.Count > 7)
34             {
35                 listBox1.Items.Clear();
36                 listBox1.Items.Add("Игра окончена!");
37                 timer1.Stop();
38             }
39         }
40         Ссылка 1
41         private void Form1_KeyDown(object sender, KeyEventArgs e)
42         {
43             if (listBox1.Items.Contains(e.KeyCode))
44             {
45                 listBox1.Items.Remove(e.KeyCode);
46                 listBox1.Refresh();
47                 if (timer1.Interval > 400) timer1.Interval -= 10;
48                 if (timer1.Interval > 250) timer1.Interval -= 7;
49                 if (timer1.Interval > 100) timer1.Interval -= 2;
50
51                 toolStripProgressBar2.Value = 800 - timer1.Interval;
52                 stats.Update(true);
53             }
54             else
55             {
56                 stats.Update(false);
57             }
58             correctLabel.Text = "Правильно: " + stats.correct;
59             missedLabel.Text = "Не правильно" + stats.missed;
60             totalLabel.Text = "Всего" + stats.total;
61             accuracyLabel.Text = "Точность" + stats.accuracy + "%";
62         }
63     }
64 }
```



### 3) Программы с формами

На данном скрине мы делали отличие иностранных от русских приложений с формами, сделали характеристическую таблицу, что лучше, а также сделали вывод, что лучше использовать для нашей работы.

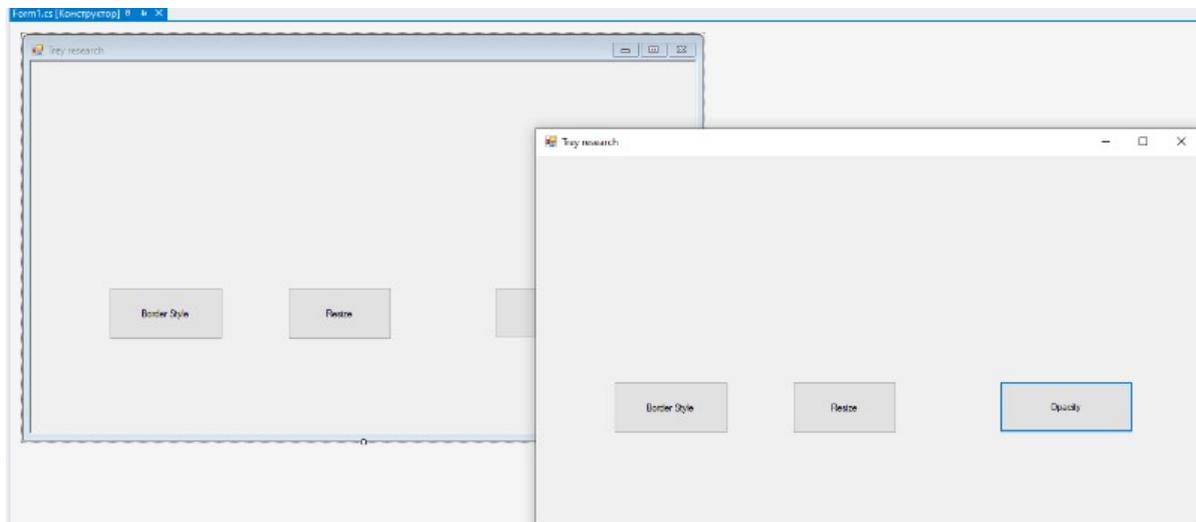
Приложение	Особенности	Ценник	Доступность
<a href="#">YouGile</a>	Легко организовать работу в больших компаниях и связать между собой отделы: есть гибкая настройка прав, доступ к чужим задачам, мониторинг рабочих процессов, прозрачность.	Облачная версия: первые 10 пользователей бесплатно и без ограничений, начиная с 11-го – 299 рублей за пользователя в месяц.	Доступна <a href="#">веб-версия</a> , <a href="#">мобильное приложение</a> и <a href="#">десктоп-версия</a>
Битрикс-24	Удобно использовать в коммерческих целях: вести неограниченное число клиентов и сделок	Облачная версия: есть бесплатная версия, но с ограниченным функционалом. Платные тарифы начинаются от 790 рублей в месяц за 2 пользователей.	Доступно <a href="#">мобильное приложение</a> и <a href="#">приложение на компьютер</a>
<a href="#">Мегаплан</a>	Удобно выставлять счета и проводить документы, выстраивать воронки продаж и обрабатывать заказы	Облачная версия: от 384 рублей за пользователя в месяц.	<a href="#">Веб-версия</a> и <a href="#">приложение для телефона</a>
<a href="#">Рутиз</a>	Много возможностей для учета задач, их классификации и подсчета времени, потраченного сотрудниками на задачи	Облачная версия: от 279 рублей за пользователя в месяц.	<a href="#">Приложение для компьютера</a> и <a href="#">телефона</a>
<a href="#">Яндекс Трекер</a>	Удобно создавать задачи на основе шаблонов, группировать задачи разных отделов, автоматизировать рутинные процессы, согласовывать документы	Облачная версия: от 258 рублей за пользователя (для первых 100 пользователей). Для больших команд стоимость рассчитывается индивидуально.	<a href="#">Веб-версия</a> , <a href="#">приложения для компьютера</a> и <a href="#">телефона</a>

#### Какую систему вы бы выбрали для своего проекта?

Я бы выбрал систему [YouGile](#), потому что в ней легко организовать работу в большой компании, а также из-за [встроенного мессенджера](#) в приложении который облегчает работу в компании. Также из-за ценника, ведь до 10 человек можно работать в нём бесплатно и только с 11 идёт плата за каждого. И конечно же главный плюс, то что можно работать в приложении и с телефона, и с компьютера.

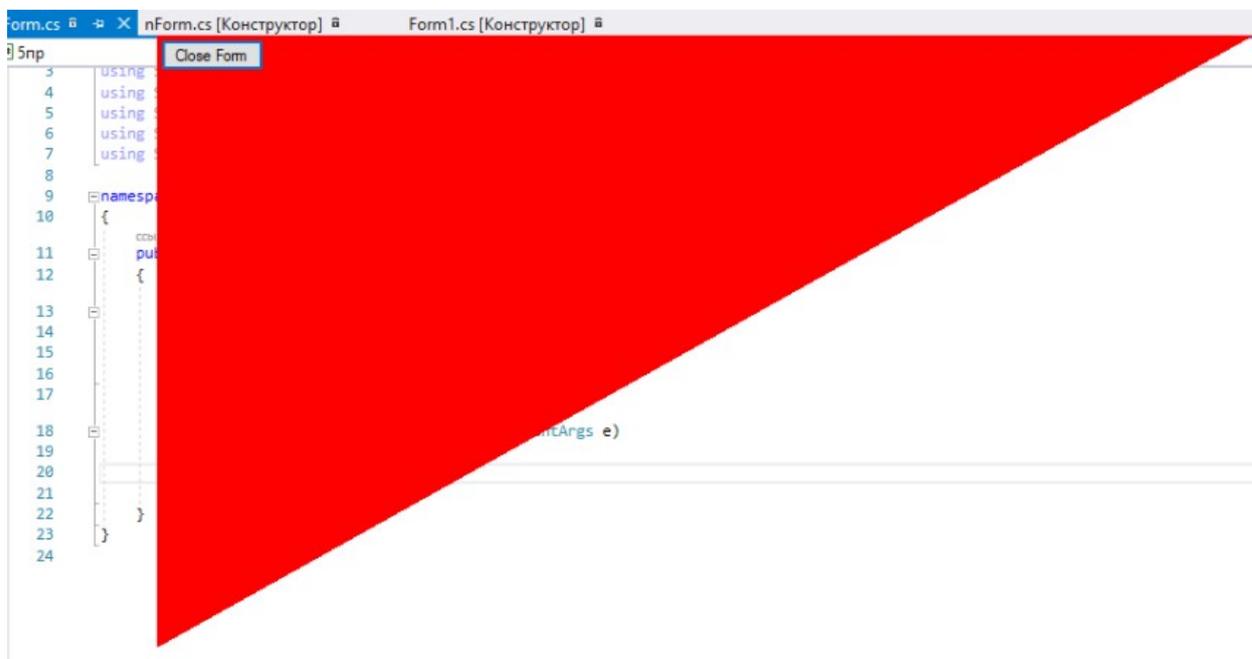
#### 4) Создание формы со свойствами

Создали форму с 3 кнопками, для того чтобы менять расположение и прозрачность самой формы.



#### 5) Форма фигуры

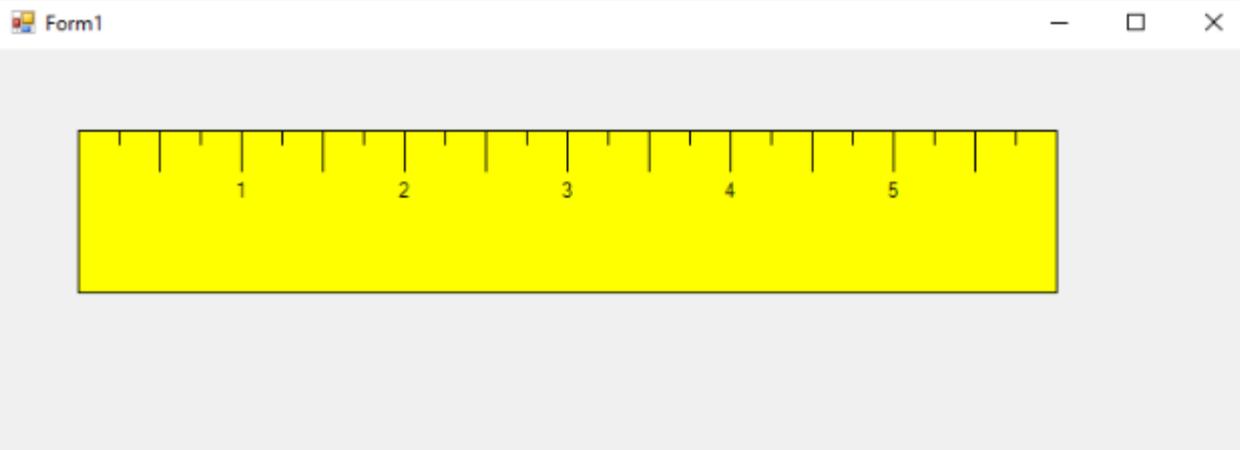
В этой практической мы при помощи кода изменили фигуру формы.



## 6-7)Линейка(Часы)

При помощи кода мы создали в форме линейку(часы), задавали значение pen и рисовали при помощи команды OnPaint. Значение часы добавляли команду DateTime.Now, для того чтобы время совпадала с нашим временем.

```
namespace gh7
{
    ССЫЛКА: 3
    public partial class Form1 : Form
    {
        ССЫЛКА: 1
        public Form1()
        {
            InitializeComponent();
        }
        ССЫЛКА: 0
        protected override void OnPaint(PaintEventArgs e)
        {
            Pen pen = new Pen(Color.Black, 0);
            SolidBrush yellow = new SolidBrush(Color.Yellow);
            SolidBrush black = new SolidBrush(Color.Black);
            e.Graphics.PageUnit = GraphicsUnit.Inch;
            e.Graphics.TranslateTransform(0.5f, 0.5f);
            e.Graphics.FillRectangle(yellow, 0, 0, 6, 1);
            e.Graphics.DrawRectangle(pen, 0, 0, 6, 1);
            for (float x = 0.25f; x < 6.0f; x += 0.25f) e.Graphics.DrawLine(pen, x, 0.0f, x, 0.08f);
            for (float x = 0.5f; x < 6.0f; x += 0.5f) e.Graphics.DrawLine(pen, x, 0.0f, x, 0.25f);
            for (float x = 1.0f; x < 6.0f; x += 1.0f) e.Graphics.DrawLine(pen, x, 0.0f, x, 0.25f);
            StringFormat format = new StringFormat();
            format.Alignment = StringAlignment.Center;
            format.LineAlignment = StringAlignment.Center;
            for (float x = 1.0f; x < 6.0f; x += 1.0f)
            {
                string label = String.Format("{0}", Convert.ToInt32(x));
                RectangleF rect = new RectangleF(x - 0.25f, 0.25f, 0.5f, 0.25f);
                e.Graphics.DrawString(label, Font, black, rect, format);
            }
            pen.Dispose();
            yellow.Dispose();
            black.Dispose();
        }
    }
}
```

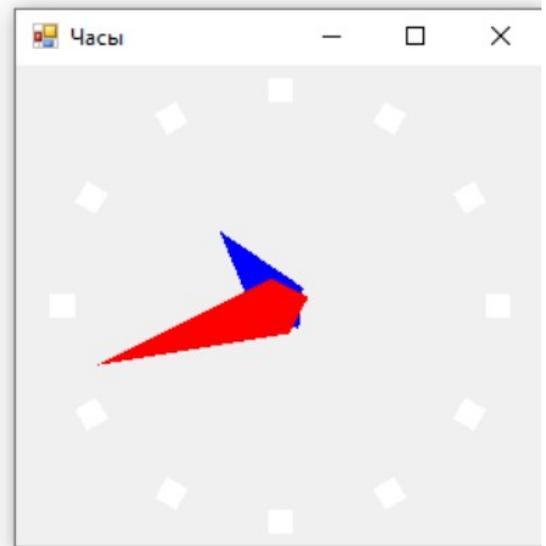


```

namespace np
{
    ССЫЛОК: 3
    public partial class Form1 : Form
    {
        ССЫЛОК: 1
        public Form1()
        {
            InitializeComponent();
            Text = "Часы";
            SetStyle(ControlStyles.ResizeRedraw, true);
        }

        ССЫЛОК: 0
        protected override void OnPaint(PaintEventArgs e)
        {
            SolidBrush red = new SolidBrush(Color.Red);
            SolidBrush white = new SolidBrush(Color.White);
            SolidBrush blue = new SolidBrush(Color.Blue);
            InitializeTransform(e.Graphics);
            for (int i = 0; i < 12; i++)
            {
                e.Graphics.RotateTransform(30.0f);
                e.Graphics.FillRectangle(white, 85, -5, 10, 10);
            }
            DateTime now = DateTime.Now;
            int minute = now.Minute;
            int hour = now.Hour % 12;
            InitializeTransform(e.Graphics);
            e.Graphics.RotateTransform((hour * 30) + (minute / 2));
            DrawHand(e.Graphics, blue, 40);
            InitializeTransform(e.Graphics);
            e.Graphics.RotateTransform(minute * 6);
            DrawHand(e.Graphics, red, 80);
            red.Dispose();
            white.Dispose();
            blue.Dispose();
        }
    }
}

```



Активация Window:  
Чтобы активировать Wind

## 8) Оценка работы другого пользователя

Проверяли работу, составляли критерии, оценивали проект другого пользователя.

### 1 Соответствие оформления документа(шрифт, размер шрифта итд)

Соответствует стандарту. Размер шрифта совпадает, сделано выравнивание по ширине, поставлен междустрочный интервал 1.5.

### 2 Наличие программы или ее прототипа

Программа по базе данных для ювелирной мастерской

### 3 Корректность технических формулировок

Формулировки корректны

### 4 Будет ли это понятно конечному пользователю

По моему мнению данное руководство будет понятно любому пользователю

### 5 Соответствие требованиям стандарта

Руководство пользователя разработано в соответствии с требованиями РД 50-34.698-90 «Информационная технология. Комплекс стандартов на автоматизированные системы. Автоматизированные системы Требования к содержанию документов», ГОСТ 2.105-95 «Единая система конструкторской документации. Общие требования к текстовым документам»

### 6 Поможет ли это руководство пользователю и почему?

Поможет потому что в нём описана сама программа и то как в ней работать

**Вывод:** Оформление соответствует стандартам, программа есть и в ней можно работать, для пользователя руководство понятное ведь в нём описано как работать в приложении.

5/5

## 9) Координаты (Рисовалка)

С помощью кода я создал приложение по типу Paint, а также создал форму для того чтобы узнать координаты мышки.

```
namespace WinFormsApp2
{
    СМЯРКА 2
    class MyForm : Form
    {
        int cx;
        int cy;

        СМЯРКА 1
        MyForm()
        {
            Text = "Mouse Tracker";

            Graphics g = Graphics.FromHwnd(Handle);
            SizeF size = g.MeasureString("MMMM", Font);
            cx = (Convert.ToInt32(size.Width) / 2) + 8;
            cy = (Convert.ToInt32(size.Width) / 2) + 8;
            g.Dispose();
        }

        СМЯРКА 0
        protected override void OnMouseMove(MouseEventArgs e)
        {
            Graphics g = Graphics.FromHwnd(Handle);
            EraseCoordinates(g);
            ShowCoordinates(e.X, e.Y, g);
            g.Dispose();
        }

        СМЯРКА 0
        protected override void OnMouseLeave(EventArgs e)
        {
            Graphics g = Graphics.FromHwnd(Handle);
            EraseCoordinates(g);
            g.Dispose();
        }

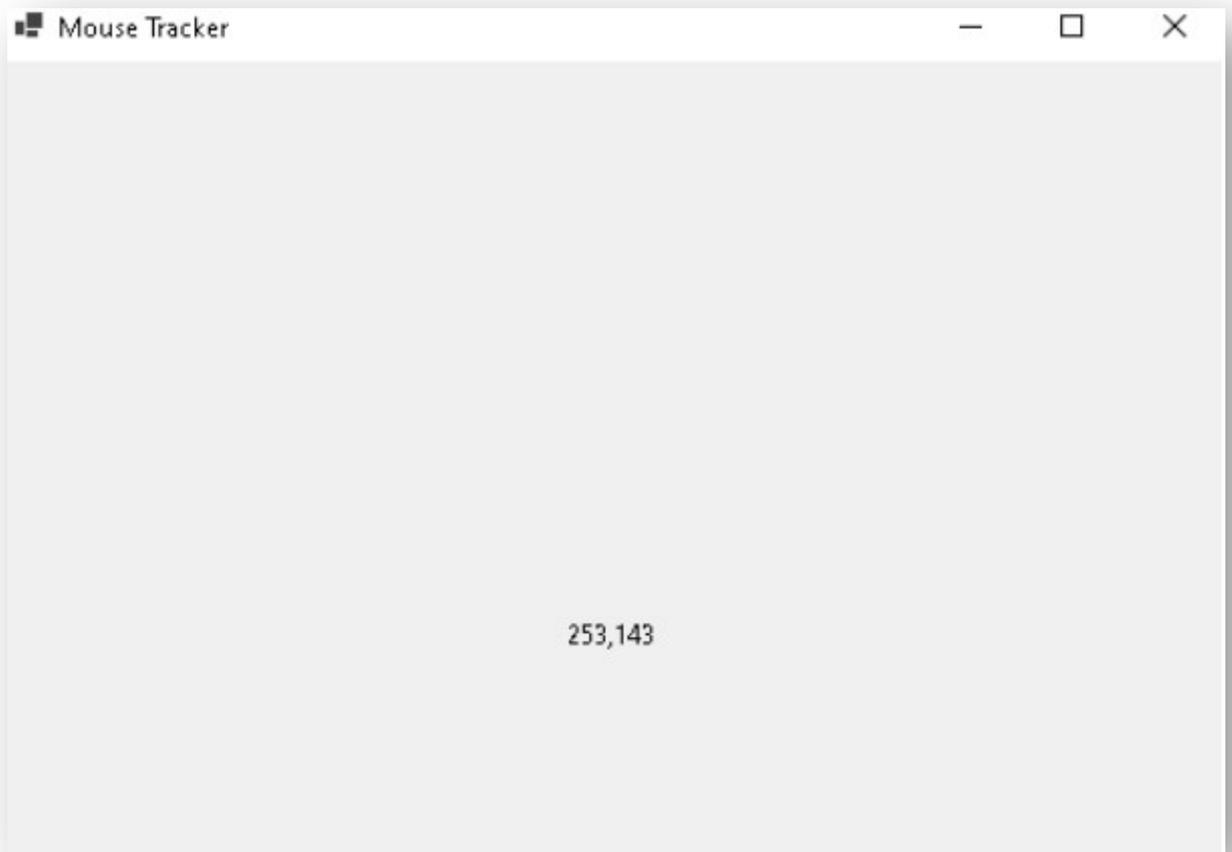
        СМЯРКА 2
        void EraseCoordinates(Graphics g)
        {
            int x = ClientRectangle.Width / 2;
            int y = ClientRectangle.Height / 2;
            SolidBrush brush = new SolidBrush(BackColor);
            g.FillRectangle(brush, x - cx, y - cy, x + cx, y + cy);
            brush.Dispose();
        }

        СМЯРКА 1
        void ShowCoordinates(int x, int y, Graphics g)
        {
            StringFormat format = new StringFormat();
            format.Alignment = StringAlignment.Center;
            format.LineAlignment = StringAlignment.Center;

            string coords = String.Format("{0},{1}", x, y);

            SolidBrush brush = new SolidBrush(Color.Black);
            g.DrawString(coords, Font, brush, ClientRectangle, format);
            brush.Dispose();
        }

        СМЯРКА 0
        static void Main()
        {
            Application.Run(new MyForm());
        }
    }
}
```





```

1  using System;
2  using System.Collections.Generic;
3  using System.ComponentModel;
4  using System.Data;
5  using System.Drawing;
6  using System.Drawing.Drawing2D;
7  using System.Linq;
8  using System.Text;
9  using System.Threading.Tasks;
10 using System.Windows.Forms;
11
12 namespace NetDraw
13 {
14     COMPOSC 2
15     class MyForm : Form
16     {
17         Stroke CurrentStroke = null;
18         System.Collections.ArrayList Strokes = new System.Collections.ArrayList();
19         COMPOSC 1
20         public MyForm()
21         {
22             Text = "NetDraw";
23             InitializeComponent();
24         }
25         COMPOSC 1
26         private void InitializeComponent()
27         {
28         }
29         COMPOSC 0
30         protected override void OnPaint(PaintEventArgs e)
31         {
32             foreach (Stroke stroke in Strokes)
33                 stroke.Draw(e.Graphics);
34         }
35         COMPOSC 0
36         protected override void OnMouseDown(MouseEventArgs e)
37         {
38             if (e.Button == MouseButtons.Left)
39             {
40                 CurrentStroke = new Stroke(e.X, e.Y);
41             }
42         }
43         COMPOSC 0
44         protected override void OnMouseMove(MouseEventArgs e)
45         {
46             if ((e.Button & MouseButtons.Left) != 0 && CurrentStroke != null)
47             {
48                 CurrentStroke.Add(e.X, e.Y);
49                 Graphics g = Graphics.FromHwnd(Handle);
50                 CurrentStroke.DrawLastSegment(g);
51                 g.Dispose();
52             }
53         }
54         COMPOSC 0
55         protected override void OnMouseUp(MouseEventArgs e)
56         {
57         }
58     }
59 }

```

```

51 | protected override void OnMouseUp(MouseEventArgs e)
52 | {
53 |     if (e.Button == MouseButtons.Left && CurrentStroke != null)
54 |     {
55 |         if (CurrentStroke.Count > 1)
56 |             Strokes.Add(CurrentStroke);
57 |         CurrentStroke = null;
58 |     }
59 | }
60 |
61 | COMPILE 0
62 | protected override void OnKeyDown(KeyEventArgs e)
63 | {
64 |     if (e.KeyCode == Keys.Delete)
65 |     {
66 |         Strokes.Clear();
67 |         Invalidate();
68 |     }
69 | }
70 |
71 | COMPILE 4
72 | class Stroke
73 | {
74 |     System.Collections.ArrayList Points = new System.Collections.ArrayList();
75 |
76 |     COMPILE 1
77 |     public int Count
78 |     {
79 |         get { return Points.Count; }
80 |     }
81 |
82 |     COMPILE 1
83 |     public Stroke(int x, int y)
84 |     {
85 |         Points.Add(new Point(x, y));
86 |     }
87 |
88 |     COMPILE 1
89 |     public void Add(int x, int y)
90 |     {
91 |         Points.Add(new Point(x, y));
92 |     }
93 |
94 |     COMPILE 1
95 |     public void Draw(Graphics g)
96 |     {
97 |         Pen pen = new Pen(Color.Black, 8);
98 |         pen.EndCap = LineCap.Round;
99 |         for (int i = 0; i < Points.Count - 1; i++)
100 |             g.DrawLine(pen, (Point)Points[i], (Point)Points[i + 1]);
101 |         pen.Dispose();
102 |     }
103 |
104 |     COMPILE 1
105 |     public void DrawLastSegment(Graphics g)
106 |     {
107 |         Point p1 = (Point)Points[Points.Count - 2];
108 |         Point p2 = (Point)Points[Points.Count - 1];
109 |         Pen pen = new Pen(Color.Black, 8);
110 |         pen.EndCap = LineCap.Round;
111 |         g.DrawLine(pen, p1, p2);
112 |         pen.Dispose();
113 |     }
114 |
115 |     COMPILE 0
116 |     static void Main()
117 |     {
118 |         Application.Run(new MyForm());
119 |     }
120 | }

```

