**Лабораторная работа No4**

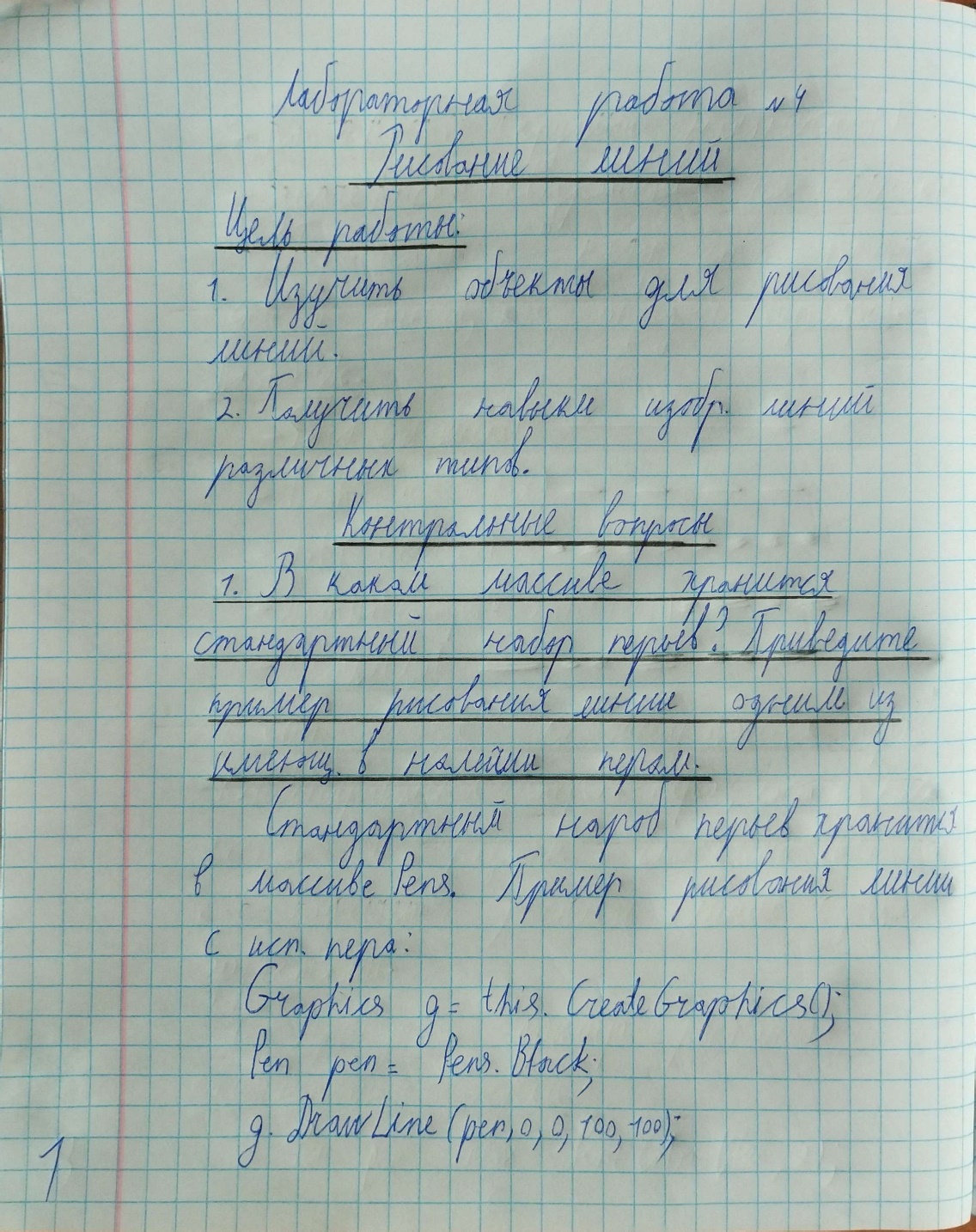
**Рисование линий**

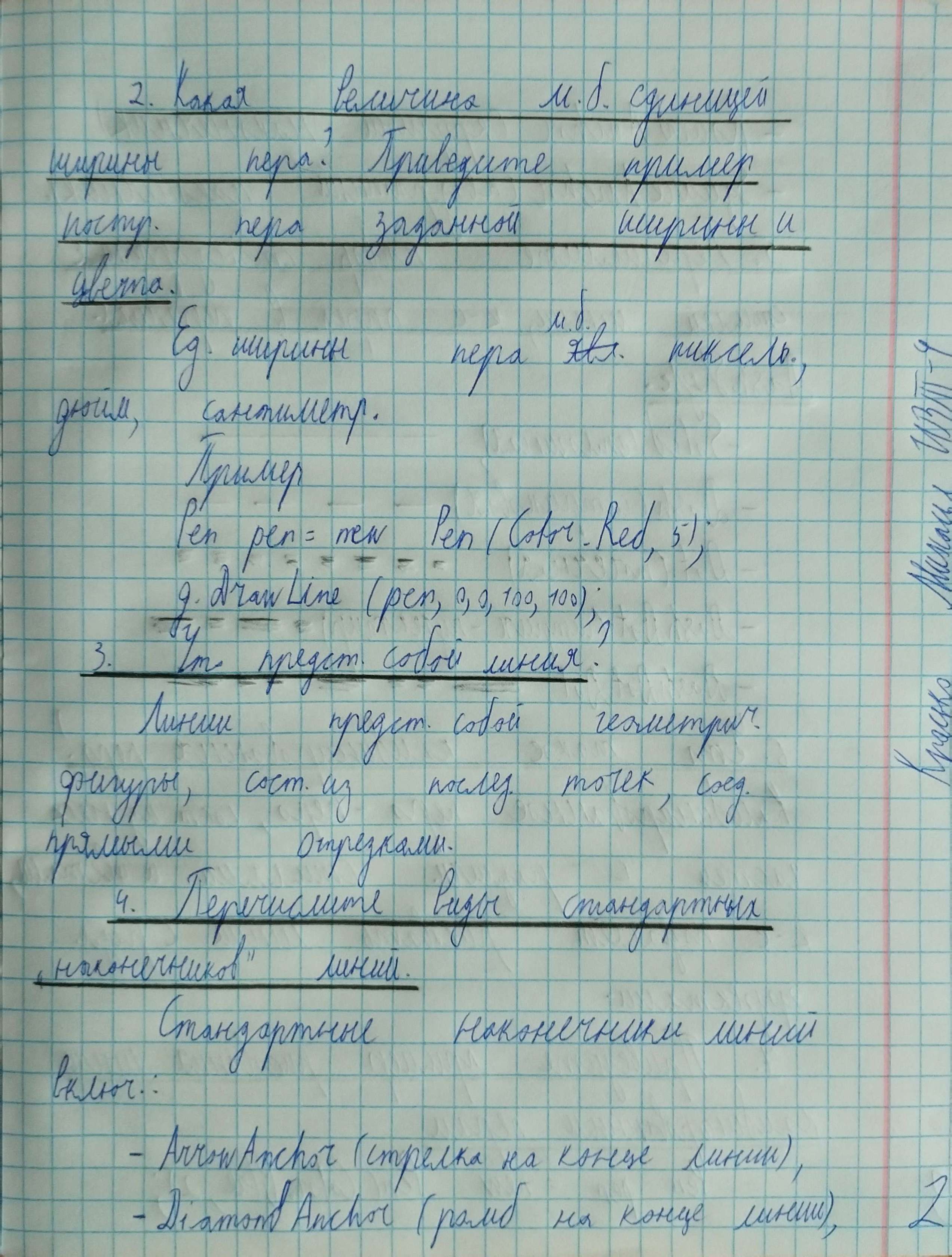
**Цель работы:**

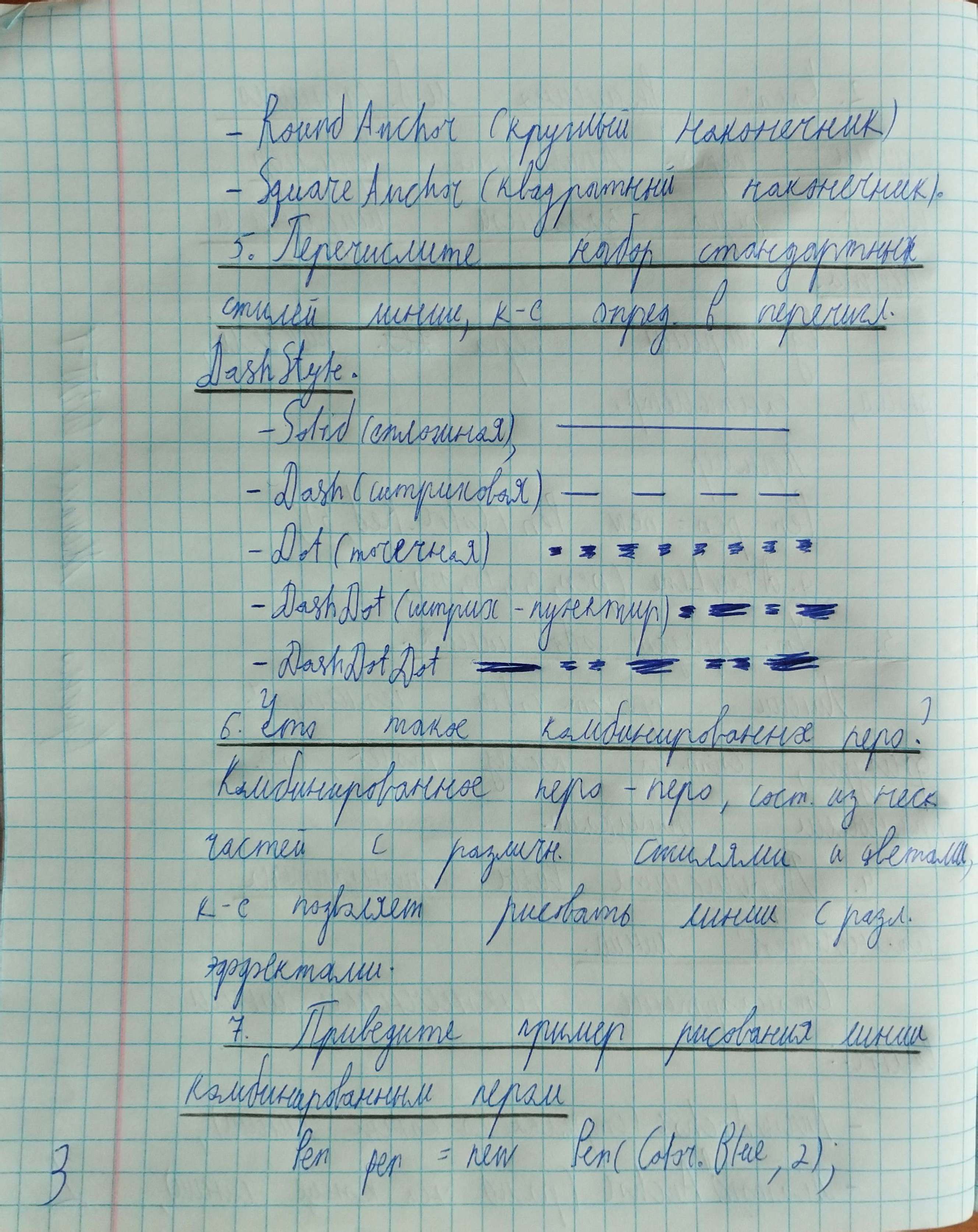
1. Изучить объекты для рисования линий.

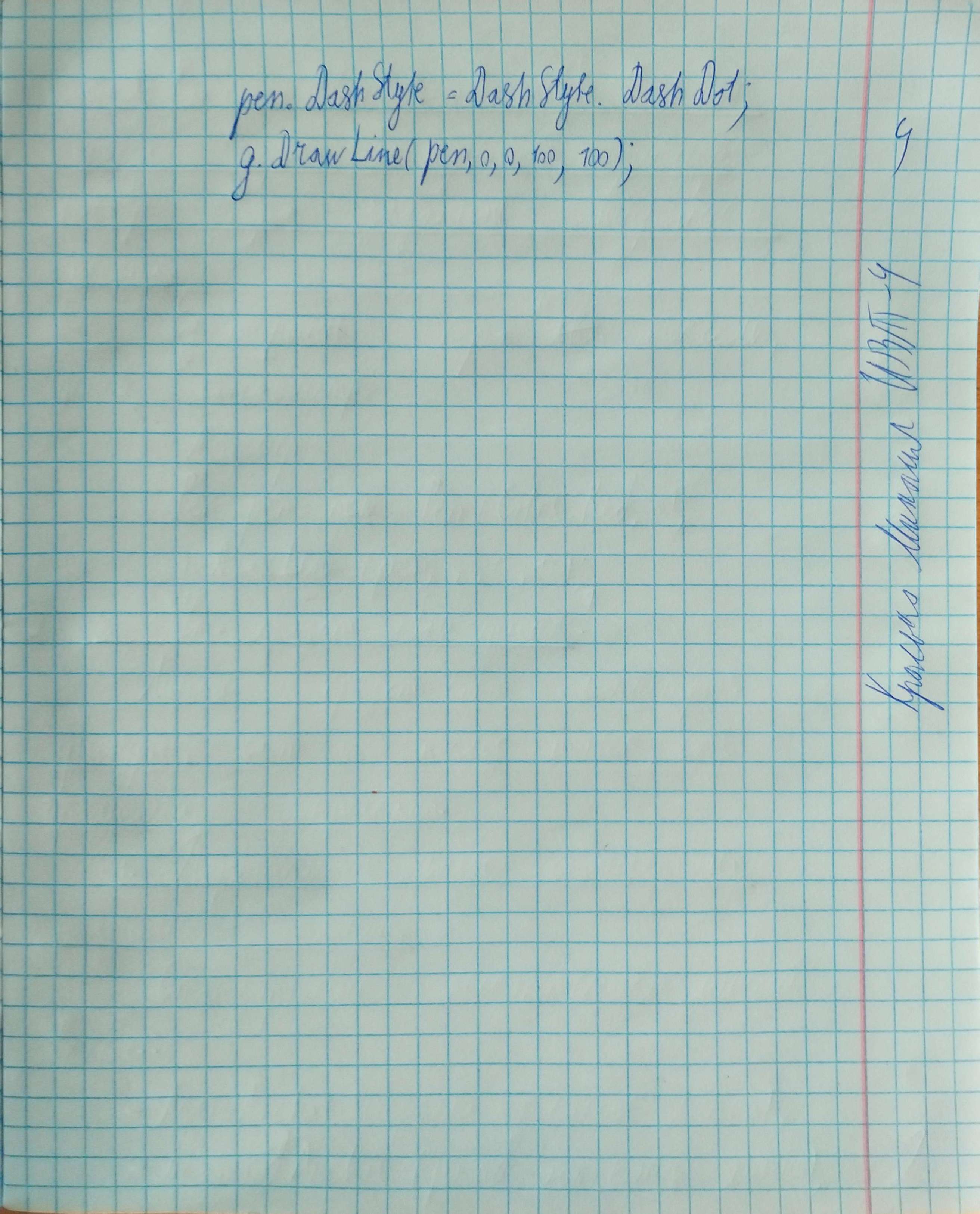
2. Получить навыки изображения линий различных типов.

**Контрольные вопросы**

****

****

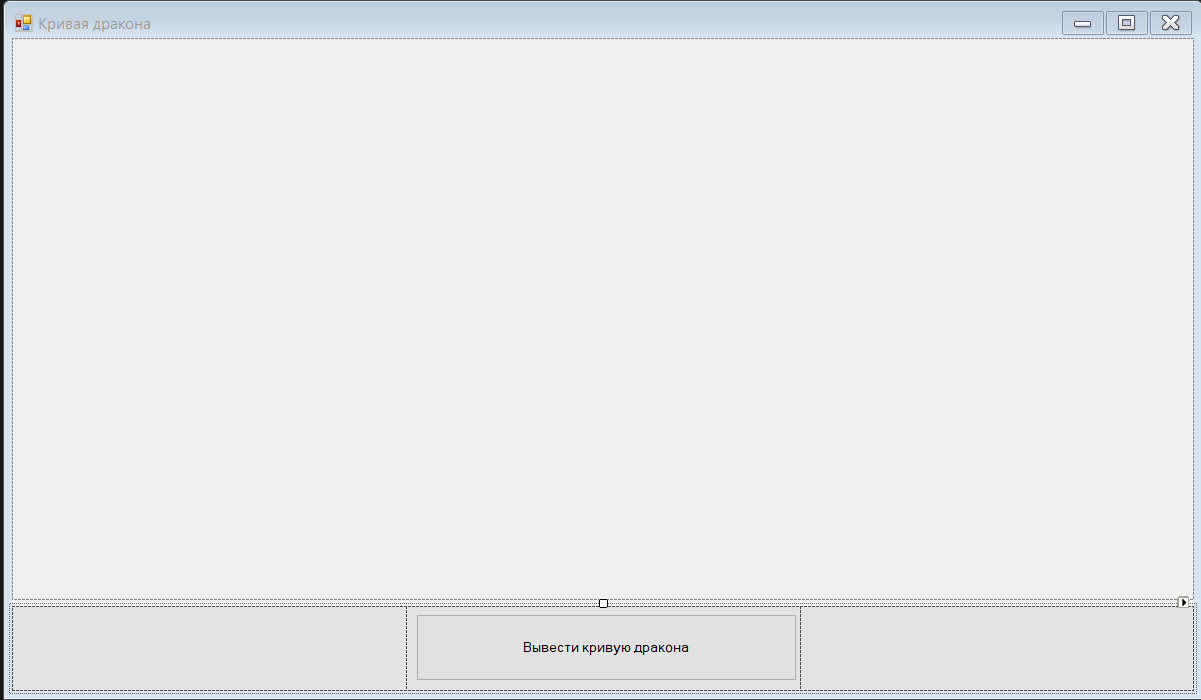
****

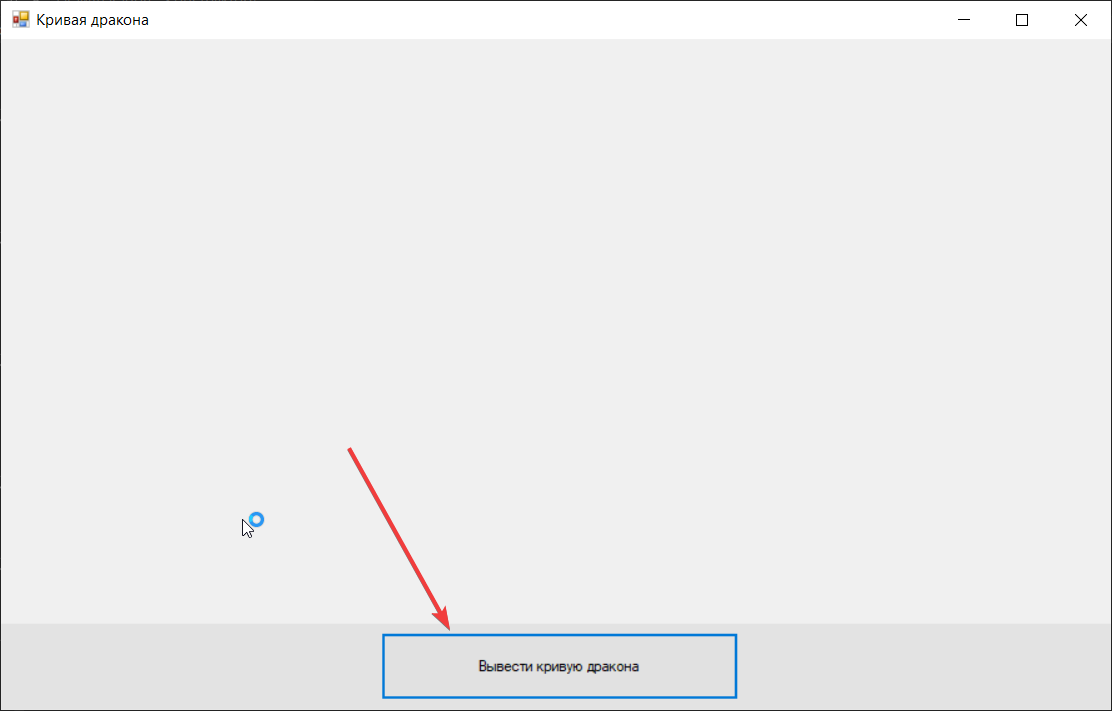
****

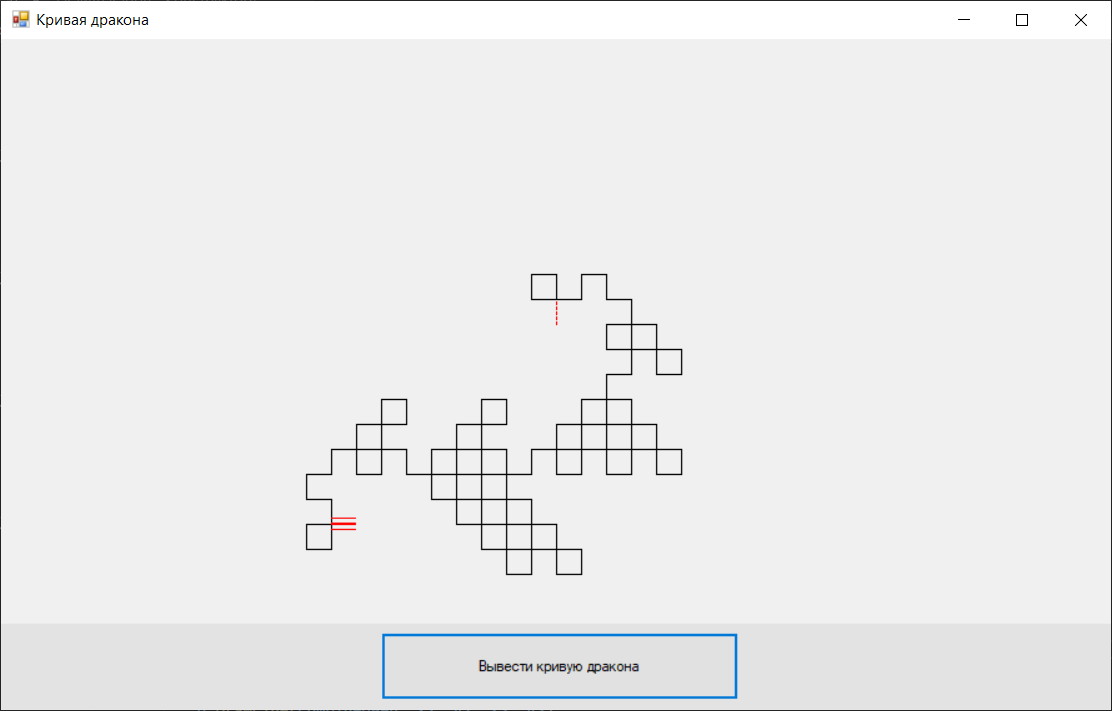
**Ход работы**

**Вариант 5**









**Код проекта:**

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 КГ\_ЛР4\_Красько\_ИВТ\_4\_2курс

{

public partial class DrawADragon : Form

{

public DrawADragon()

{

InitializeComponent();

}

private string CreateDragonLine()

{

// задаем порядок кривой дракона

int orderCurve = 7;

// задаем кривую дракона первого порядка

string str = "1";

for (int i = 1; i < orderCurve; i++)

{ // создаем на базе строки предыдущего порядка изменяемую

//строку символов

StringBuilder sb = new StringBuilder(str);

// находим индекс центральной цифры

int seredina = (int)Math.Floor((double)sb.Length / 2);

// меняем символ с этим индексом на 0

sb[seredina] = '0';

// формируем строку нового порядка кривой дракона

str = str + "1" + sb;

}

return str;

}

private void Btn\_DrawDragon\_Click(object sender, EventArgs e)

{

Graphics g = pictureBox\_FieldForDragon.CreateGraphics();

Pen p = new Pen(Color.Red);

p.DashStyle = System.Drawing.Drawing2D.DashStyle.Dash;

// p.CompoundArray = new float[] { 0.0f, 0.2f, 0.3f, 0.6f,

// 0.7f, 1.0f };

int dx = 20;

// Формируем строку кривой дракона

string str = CreateDragonLine();

// начальная точка первой линии

int x1 = pictureBox\_FieldForDragon.Size.Width / 2;

int y1 = pictureBox\_FieldForDragon.Size.Height / 2;

// конечная точка первой линии

int x2 = pictureBox\_FieldForDragon.Size.Width / 2;

int y2 = pictureBox\_FieldForDragon.Size.Height / 2 - dx;

// сохраняем координаты конечной точки

int x3 = x2; int y3 = y2;

// рисуем линию из начальной в конечную точку

g.DrawLine(p, x1, y1, x3, y3);

// Цикл по всем цифрам кривой дракона

for (int i = 0; i < str.Length; i++)

{

if (y2 - y1 < 0)

{ // рисовали вверх на предыдущем шаге

if (str[i] == '1') x3 = x2 - dx; // поворот налево

else x3 = x2 + dx; // поворот направо

y3 = y2;

}

if (x2 - x1 < 0)

{ // рисовали влево на предыдущем шаге

if (str[i] == '1') y3 = y2 - dx; // поворот налево

else y3 = y2 + dx; // поворот направо

x3 = x2;

}

if (x2 - x1 > 0)

{ // рисовали вправо на предыдущем шаге

if (str[i] == '1') y3 = y2 + dx; // поворот налево

else y3 = y2 - dx; // поворот направо

x3 = x2;

}

if (y2 - y1 > 0)

{ // рисовали вниз на предыдущем шаге

if (str[i] == '1') x3 = x2 + dx; // поворот налево

else x3 = x2 - dx; // поворот направо

y3 = y2;

}

if (i == str.Length - 1)

{

// Создаем перо с комбинированным стилем

Pen combinedPen = new Pen(Color.Red, 10); // тут нужна большая ширина, иначе из-за малого числа пикселей расслоение не будет видно

combinedPen.CompoundArray = new float[] { 0.0f, 0.1f, 0.4f, 0.6f, 0.9f, 1.0f }; // разделение пера на несколько частей

g.DrawLine(combinedPen, x2, y2, x3, y3);

}

else

{ // цвет и стиль остальных линий

g.DrawLine(Pens.Black, x2, y2, x3, y3);

}

// переприсваивание для следующего шага

x1 = x2; y1 = y2;

x2 = x3; y2 = y3;

} // конец цикла по цифрам кривой дракона

} // конец метода обработки события клика на кнопку

}

}

**Вывод:** изучили объекты для рисования линий, получили навыки изображения линий различных типов.