

Практическая работа 29

Графика

Количество баллов за работу – 22 (17 баллов за выполнение заданий + 5 баллов за индивидуальное задание)

Теория

Графический интерфейс приложений C# (GDI+), как и других приложений, предназначенных для работы в рамках Microsoft .NET Framework, состоит из набора классов, объединяемых пространством имен.

Одно из основных пространств имен GDI+ языка C# является пространство имен **System.Drawing**.

Классы этого пространства имен определяют перечень объектов и инструментов, предназначенных для «рисования». К наиболее часто используемым классам пространства имен **System.Drawing** относятся:

✓ **Brush** (Brushes, SolidBrush и др.).

Объекты **Brush** (кисть) используются для заполнения пространства внутри геометрических фигур. Тип **Brush** — это абстрактный базовый класс, остальные типы являются производными от **Brush** и определяют разные наборы возможностей.

✓ **Pen** (Pens, SystemPens).

Pen (перо) — это объект класса, при помощи которого можно рисовать прямые и кривые линии. В классе **Pen** определен набор статических свойств, при помощи которых можно получить объект Реп с заданными свойствами (например, с установленным цветом).

✓ **Font** (FontFamily).

Объекты типа **Font** определяют характеристики шрифта (имя, размер, начертание и т. п.). **FontFamily** представляет набор шрифтов, которые относятся к одному семейству, но имеют некоторые небольшие отличия.

Graphics. Этот класс определяет набор свойств и методов для вывода текста, изображений и геометрических фигур на экран монитора. Он позволяет приложению работать с контекстом устройств системы **Windows**.

✓ **Region.**

Этот класс определяет область, занятую геометрической фигурой. **Point** (**PointF**). Эти структуры обеспечивают работу с координатами точки. **Point** работает со значениями типа **int**, а **PointF** — со значениями типа **float**.

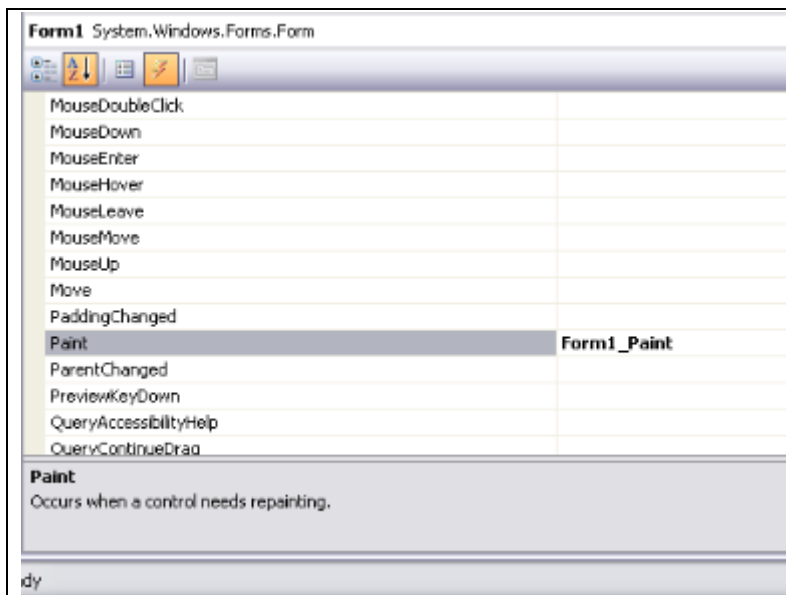
В пространстве имен **System.Drawing** также находятся классы **Icon**, **Image**, **Color**, **Bitmap** и другие классы так или иначе связанные с отображением графической информации на экране монитора.

Основным классом для «рисования» в языке C# является класс **Graphics**. Он предназначен для вывода графической информации в клиентскую часть формы приложения.

Для того чтобы приложение могло что-нибудь нарисовать в окне, оно должно получить или создать для этого окна объект класса **Graphics**.

Далее, пользуясь свойствами и методами этого объекта, приложение может рисовать в окне различные фигуры или текстовые строки. Прежде чем создавать в приложении объект класса **Graphics** необходимо определиться с обработчиком события по «рисованию».

В системе Windows за перемещением и изменением размера окон «следит» специальное сообщение **WM_PAINT**, которое при необходимости извещает приложения, о том, что им следует перерисовать содержимое окна. Любые действия с окном – перемещение его по рабочему столу экрана монитора, изменение его размеров и т.д. сопровождается требованием системы Windows «перерисовать» окно. В приложении обработчик события **WM_PAINT**, получив такое сообщение, должен выполнить перерисовку всего окна или его части, в зависимости от дополнительных данных, полученных вместе с сообщением **WM_PAINT**. Для создания заготовки обработчика сообщения **WM_PAINT** формы необходимо в окне свойств окна формы дважды кликнуть мышкой на пункт **PAINT**.

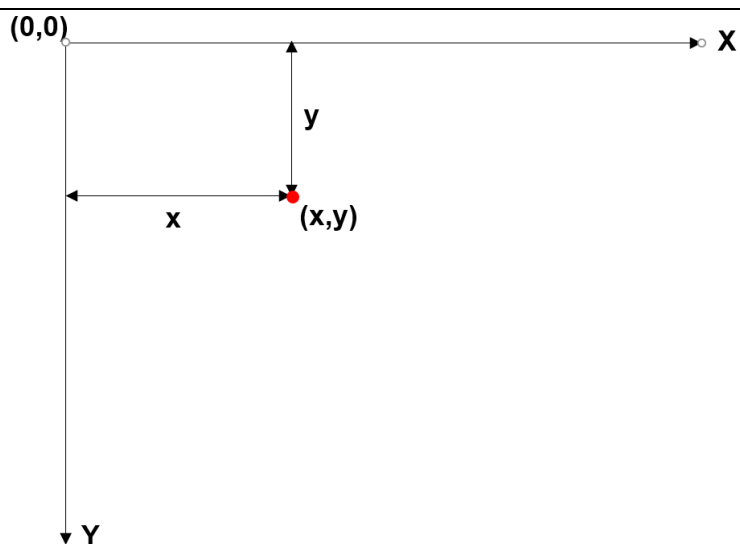


В результате будет создан обработчик события WM_PAINT (точнее Form1_Paint). Этот обработчик будет получать управление всякий раз, когда по тем или иным причинам возникнет необходимость в перерисовке содержимого окна нашего приложения. Вот в каком виде будет создан обработчик события Paint:

```
private void Form1_Paint(object sender, PaintEventArgs e)
{
}
```

Обработчику **Form1_Paint** передаются два параметра. Через первый параметр передается ссылка на объект, вызвавший событие. В нашем случае это будет ссылка на форму **Form1** (где рисовать). Что же касается второго параметра, то через него передается ссылка на объект класса **PaintEventArgs**. Этот объект имеет свойство **ClipRectangle**, доступное только для чтения. Через свойство **ClipRectangle** передаются границы области, которую должен перерисовать обработчик события **Paint**. Эти границы передаются в виде объекта класса **Rectangle**. Свойства этого класса **Left**, **Right**, **Width** и **Height**, наряду с другими свойствами, позволяют определить расположение и размеры области. По умолчанию обработчик события **Paint** игнорирует свойство **ClipRectangle**, перерисовывая содержимое окна полностью.

Система координат имеет значение 0,0 в верхней левой точке



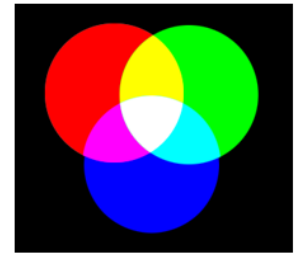
Цвета:

Код	Название
0	BLACK
1	BLUE
2	GREEN
3	CYAN
4	RED
5	MAGENTA
6	BROWN
7	LIGHTGRAY

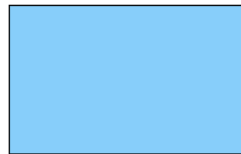
Код	Название
8	DARKGRAY
9	LIGHTBLUE
10	LIGHTGREEN
11	LIGHTCYAN
12	LIGHTRED
13	LIGHTMAGENTA
14	YELLOW
15	WHITE

ЦВЕТ = R + G + B

Red	Green	Blue
красный	зеленый	синий
0..255	0..255	0..255



R = 218
G = 164
B = 32



R = 135
G = 206
B = 250



Сколько разных цветов?

256·256·256 = 16 777 216 (True Color)

Задание 1. Пробуем рисовать точки

Чтобы рисунок отобразился на форме, необходимо обработать событие формы Paint.

1. Создайте новое приложение
2. Зайдите в обработчик событий **void Form1_Paint** и пропишите следующий код:

```
private void Form1_Paint(object sender, PaintEventArgs e)
{
    Random rnd = new Random();
    Pen mypen = new Pen(Color.Red, 2); //объявляем объект "ручка", задаем цвет
    SolidBrush mybrush = new SolidBrush(Color.Red); //объявляем объект - цвет заливки
    Graphics g = e.Graphics; //определяем холст, на котором будем рисовать
    int x = rnd.Next(this.Width); //случайное значение по ширине
    int y = rnd.Next(this.Height); //случайное значение по высоте
    //имитация точки
    g.FillRectangle(mybrush, new Rectangle(x,y,1,1)); //в случайном месте создаем закрашенный прямоугольник стороны которого равны 1 пиксель
}
```

3. Запустите приложение. в случае необходимости, исправьте ошибки*
4. Измените приложение так, чтобы на экране появлялось 1000 точек*
5. Изменим цвет точки:

```
//задаем случайный цвет в формате RGB
Color mycolor= Color.FromArgb(255, rnd.Next(255), rnd.Next(255), rnd.Next(255));
// меняем цвет кисти и пера
mypen.Color = mycolor;
mybrush.Color = mycolor;
```

6. Измените приложение так, чтобы на экране появлялось 1000 точек случайных цветов*
7. Добавьте компонент кнопка на форму. Задайте свойство text компонента значением «Точки»

Необходимо программу в коде события Paint перенести в событие нажатия на кнопку. Для этого очистим программу между фигурными скобками события Paint и перенесем его в событие click кнопки:

```
private void Form1_Paint(object sender, PaintEventArgs e)
{
}

private void button1_Click(object sender, EventArgs e)
{
    Random rnd = new Random();
    Pen mypen = new Pen(Color.Red, 2); //объявляем объект "ручка", задаем цвет
    SolidBrush mybrush = new SolidBrush(Color.Red); //объявляем объект - цвет заливки

    Graphics g = this.CreateGraphics(); //определяем холст, на котором будем рисовать

    //имитация точки
    for (int i = 0; i < 1000; i++)
    {
        //задаем случайный цвет в формате RGB
        Color mycolor = Color.FromArgb(255, rnd.Next(255), rnd.Next(255), rnd.Next(255));
        // меняем цвет кисти и пера
        mypen.Color = mycolor;
        mybrush.Color = mycolor;
        int x = rnd.Next(this.Width); //случайное значение по ширине
        int y = rnd.Next(this.Height); //случайное значение по высоте
        //в случайном месте создаем закрашенный прямоугольник стороны которого равны 1 пиксель
        g.FillRectangle(mybrush, new Rectangle(x, y, 1, 1));
    }
}
```

8. Запустите программу. В случае необходимости, исправьте ошибки*

Задание 2. Пробуем рисовать

1. Добавьте компонент кнопка на форму. Задайте свойство text компонента значением «Линии»
2. Опишите событие нажатия на кнопку следующим образом:

```

private void button2_Click(object sender, EventArgs e)
{
    Random rnd = new Random();
    Pen mypen = new Pen(Color.Red, 2); //объявляем объект "ручка", задаем цвет
    SolidBrush mybrush = new SolidBrush(Color.Red); //объявляем объект - цвет заливки

    Graphics g = this.CreateGraphics(); //определяем холст, на котором будем рисовать
    //задаем случайный цвет в формате RGB
    Color mycolor = Color.FromArgb(255, rnd.Next(255), rnd.Next(255), rnd.Next(255));
    // меняем цвет кисти и пера
    mypen.Color = mycolor;
    mybrush.Color = mycolor;
    int x1 = rnd.Next(this.Width); //случайное значение по ширине
    int y1 = rnd.Next(this.Height); //случайное значение по высоте
    int x2 = rnd.Next(this.Width); //случайное значение по ширине
    int y2 = rnd.Next(this.Height); //случайное значение по высот
    //в случайном месте создаем линию от точки с координатами x1,y1 до точки с координатами x2,y2
    g.DrawLine(mypen, new Point(x1, y1), new Point(x2, y2));
}

```

3. Запустите проект. В случае необходимости, исправьте ошибки*
4. Измените программу так, чтобы по нажатию на кнопку, рисовалась ни одна, а 12 случайных линий, случайного цвета*
5. Добавьте компонент кнопка на форму. Задайте свойство text компонента значением «Ломаная»
6. Опишите событие нажатия на кнопку следующим образом:

```

private void button3_Click(object sender, EventArgs e)
{
    Random rnd = new Random();
    Pen mypen = new Pen(Color.Red, 2); //объявляем объект "ручка", задаем цвет
    SolidBrush mybrush = new SolidBrush(Color.Red); //объявляем объект - цвет заливки

    Graphics g = this.CreateGraphics(); //определяем холст, на котором будем рисовать
    //задаем случайный цвет в формате RGB
    Color mycolor = Color.FromArgb(255, rnd.Next(255), rnd.Next(255), rnd.Next(255));
    // меняем цвет кисти и пера
    mypen.Color = mycolor;
    mybrush.Color = mycolor;
    int x1 = rnd.Next(this.Width); //случайное значение по ширине
    int y1 = rnd.Next(this.Height); //случайное значение по высоте
    int x2 = rnd.Next(this.Width); //случайное значение по ширине
    int y2 = rnd.Next(this.Height); //случайное значение по высот
    // задаем массив точек
    Point[] mypoints = new Point[5];
    // перебираем точки с первой до последней
    for (int i = 0; i < 5; i++)
    {
        //задаем координаты точки случайным образом
        mypoints[i].X = rnd.Next(this.Width);
        mypoints[i].Y = rnd.Next(this.Height);
    }
    g.DrawLines(mypen, mypoints);
}

```

7. Запустите проект. В случае необходимости, исправьте ошибки*

8. Добавьте компонент кнопка на форму. Задайте свойство text компонента значением «Прямоугольник»
9. Опишите событие нажатия на кнопку следующим образом:

```
private void button4_Click(object sender, EventArgs e)
{
    Random rnd = new Random();
    Pen mypen = new Pen(Color.Red, 2); //объявляем объект "ручка", задаем цвет
    SolidBrush mybrush = new SolidBrush(Color.Red); //объявляем объект - цвет заливки

    Graphics g = this.CreateGraphics(); //определяем холст, на котором будем рисовать
    //задаем случайный цвет в формате RGB
    Color mycolor = Color.FromArgb(255, rnd.Next(255), rnd.Next(255), rnd.Next(255));
    // меняем цвет кисти и пера
    mypen.Color = mycolor;
    mybrush.Color = mycolor;
    int x1 = rnd.Next(this.Width); //случайное значение по ширине
    int y1 = rnd.Next(this.Height); //случайное значение по высоте
    int x2 = rnd.Next(this.Width); //случайное значение по ширине
    int y2 = rnd.Next(this.Height); //случайное значение по высот
    // рисуем прямоугольник
    g.DrawRectangle(mypen, new Rectangle(x1, y1, x2, y2));
}
```

10. Запустите проект. В случае необходимости, исправьте ошибки*

```
g.FillRectangle(mybrush, new Rectangle(x1, y1, x2, y2)); //рисуем закрашенный прямоугольник
```

11. Самостоятельно добавьте кнопку «Прямоугольник» и обработчик её события, чтобы по нажатию рисовался закрашенный прямоугольник*
12. Самостоятельно добавьте кнопку «Прямоугольник с контуром» и обработчик её события, чтобы по нажатию рисовался закрашенный прямоугольник с контуром другого цвета*
13. Добавьте компонент кнопка на форму. Задайте свойство text компонента значением «Круг»
14. Опишите событие нажатия на кнопку следующим образом:


```

private void button5_Click(object sender, EventArgs e)
{
    Random rnd = new Random();
    Pen mypen = new Pen(Color.Red, 2); //объявляем объект "ручка", задаем цвет
    SolidBrush mybrush = new SolidBrush(Color.Red); //объявляем объект - цвет заливки

    Graphics g = this.CreateGraphics(); //определяем холст, на котором будем рисовать
    //задаем случайный цвет в формате RGB
    Color mycolor = Color.FromArgb(255, rnd.Next(255), rnd.Next(255), rnd.Next(255));
    // меняем цвет кисти и пера
    mypen.Color = mycolor;
    mybrush.Color = mycolor;
    int x1 = rnd.Next(this.Width); //случайное значение по ширине
    int y1 = rnd.Next(this.Height); //случайное значение по высоте
    int x2 = rnd.Next(this.Width); //случайное значение по ширине
    int y2 = rnd.Next(this.Height); //случайное значение по высот
    // рисуем круг. круг - это эллипс в квадрате
    g.DrawEllipse(mypen, new Rectangle(x1, x1, y1, y1));
}

```

15. Запустите проект. В случае необходимости, исправьте ошибки*
16. Самостоятельно добавьте кнопку «Закрашенный круг» и обработчик её события, чтобы по нажатию рисовался закрашенный прямоугольник*
17. Самостоятельно добавьте кнопку «Круг с контуром» и обработчик её события, чтобы по нажатию рисовался закрашенный прямоугольник с контуром другого цвета*

Задание 3. Рисование по нажатию на кнопку

- 1 Добавьте кнопку на форму
- 2 Добавьте обработку события на нажатие кнопки:

```
private void button1_Click(object sender, EventArgs e)
{
    //Выбираем перо "myPen" черного цвета Black
    //толщиной в 1 пиксель:
    Pen myWind = new Pen(Color.Black, 2);
    //Объявляем объект "g" класса Graphics и предоставляем
    //ему возможность рисования на pictureBox1:
    Graphics g = this.CreateGraphics();
    // Создаем объекты для закрашивания фигур
    SolidBrush myCorp = new SolidBrush(Color.Black);
    SolidBrush myTrum = new SolidBrush(Color.Green);
    SolidBrush myTrub = new SolidBrush(Color.Gray);
    SolidBrush mySe = new SolidBrush(Color.Blue);
    // Закрашиваем фигуры
    g.FillRectangle(myTrub, 300, 125, 75, 75);
    g.FillRectangle(myTrub, 480, 125, 75, 75);
    g.FillPolygon(myCorp, new Point[] {
        new Point(100,300),new Point(700,300),
        new Point(700,300),new Point(600,400),
        new Point(600,400),new Point(200,400),
        new Point(200,400),new Point(100,300)
    });
    g.FillRectangle(myTrum, 250, 200, 350, 100);
    int x = 50;
    int Radius = 50;
}
```

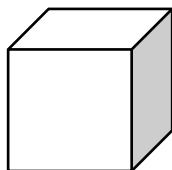
3 Запустите программу на выполнение *

Задание 4. Самостоятельная работа

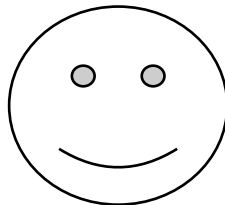
Решите представленные далее задачи.

1 Нарисуйте на форме следующие изображения: **

a.



b.



2 Изобразите на форме флаг России *

Задание 5 (Индивидуальное задание)

Вариант задания определяется порядковым номером студента в журнале преподавателя и является УНИКАЛЬНЫМ в пределах курса!!!

Для группы ИСП-1-21 номер задания = порядковому номеру

Для группы ИСП-2-21 номер задания = порядковому номеру+25

Для группы ИСП-3-21 номер задания = порядковому номеру+50

Для группы ИС-1-20 номер задания = порядковому номеру +75


Задание – проект необходимо сохранить в папке с названием ИНДИВИДУАЛЬНАЯ РАБОТА и номером практической работы

Программа должна иметь комментарии к действиям. Также перед кодом обработчика события нажатия на кнопку необходимо добавить комментарий с личными данными студента (фамилия, имя, группа и текст задания).

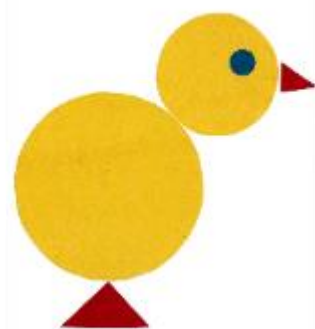
Программа при закрытии должна выдавать диалоговое окно подтверждения закрытия формы.

Не забывайте проверять на правильность ввода данных. Все поля должны быть заполнены. В противном случае, программа должна выдавать ошибку.

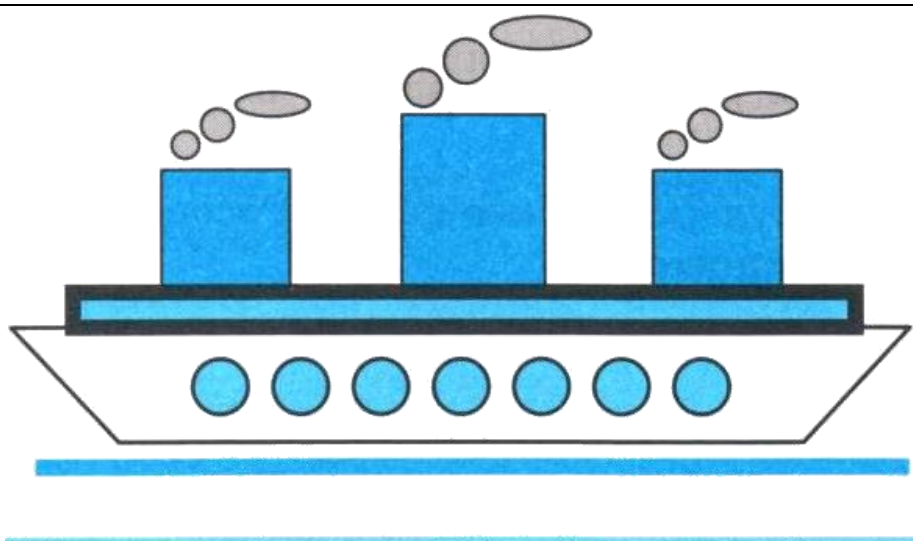
Выполните рисунок по нажатию на кнопку

№	Условие
1)	

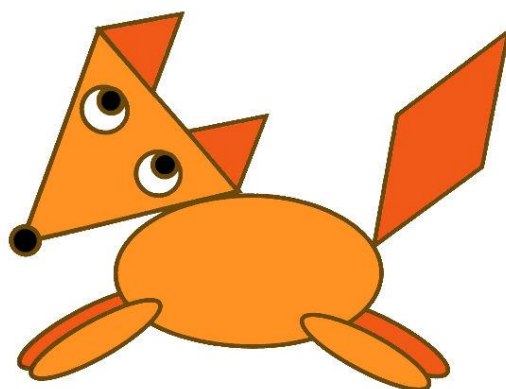
2)



3)



4)



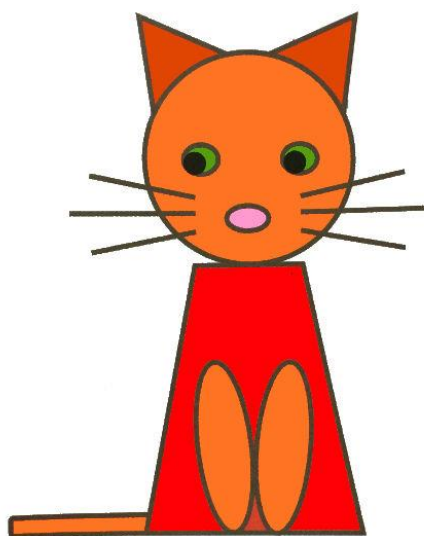
novye-deti.ru

5)



nove-deti.ru

6)



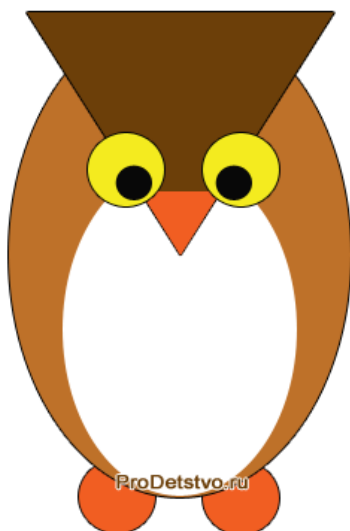
nove-deti.ru

7)

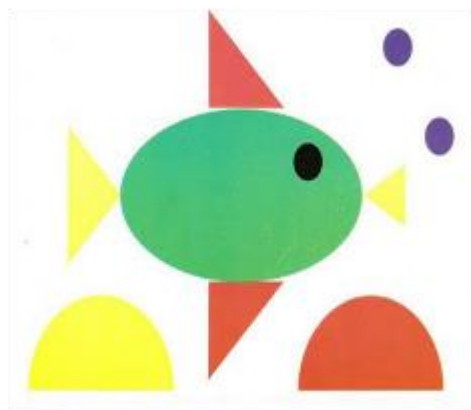


© 2004-2010 e-papa.ru

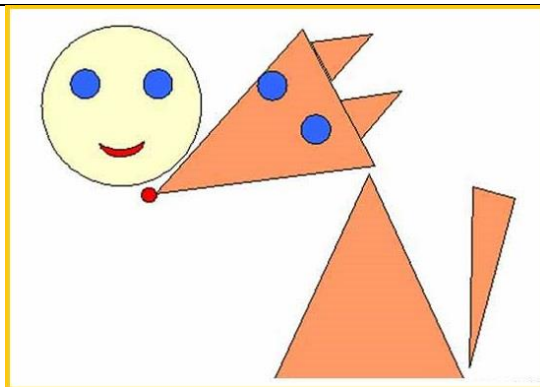
8)

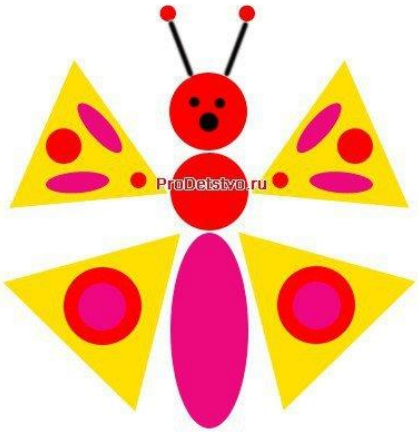
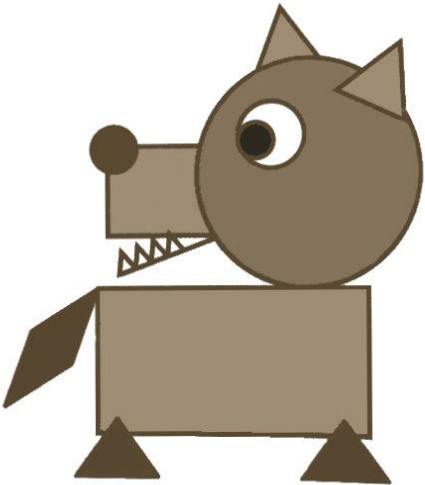

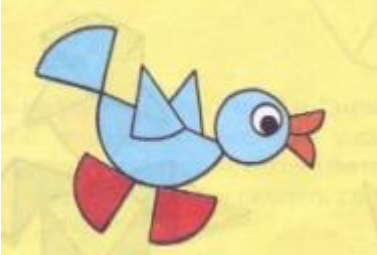





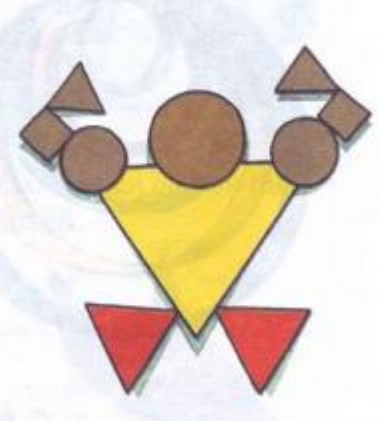
9)

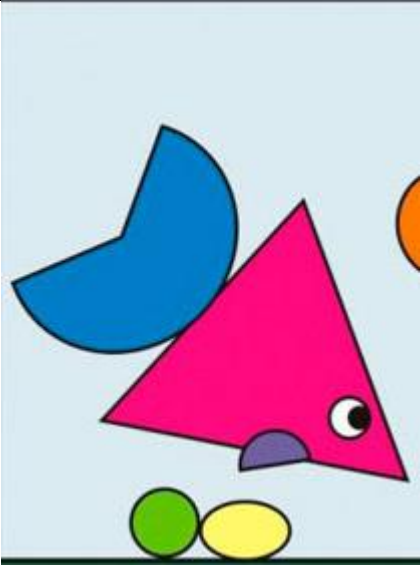
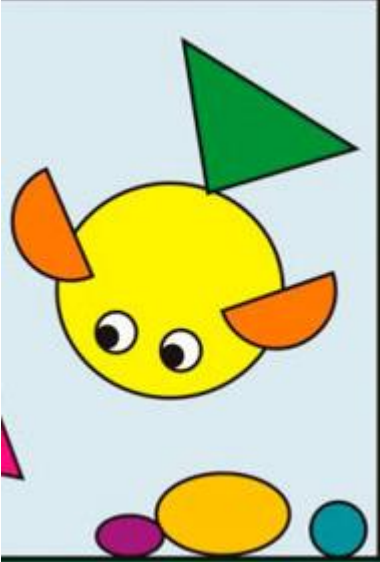
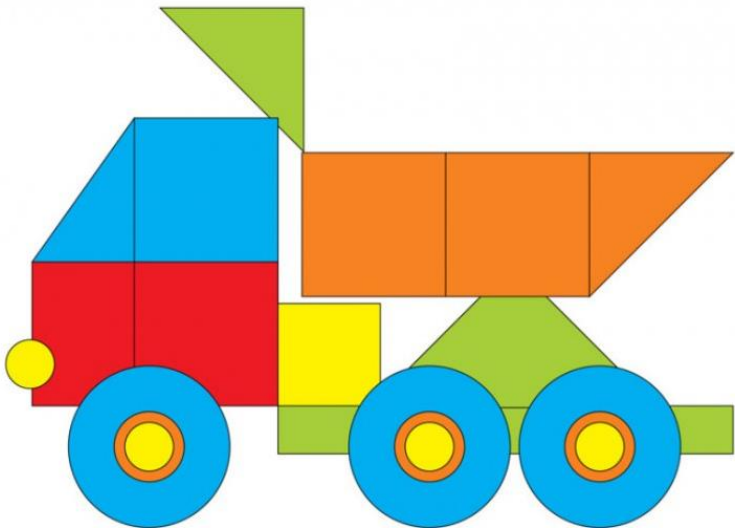


10)

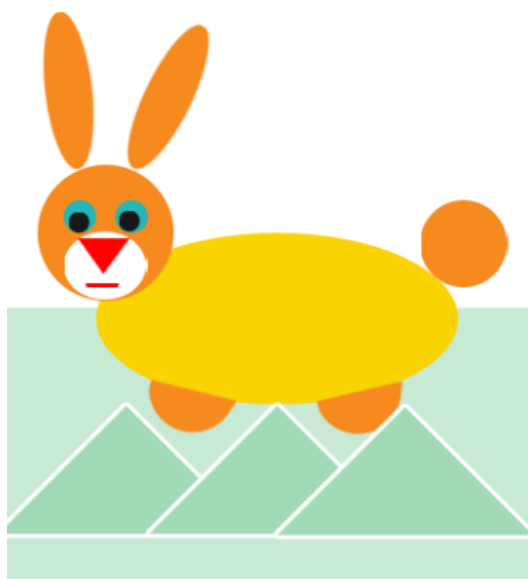


11)	
12)	 <p data-bbox="828 1189 1002 1223">novye-deti.ru</p>
13)	
14)	

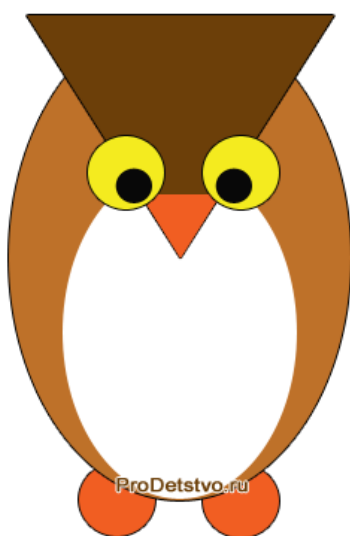
15)	
16)	
17)	
18)	

19)	
20)	
21)	

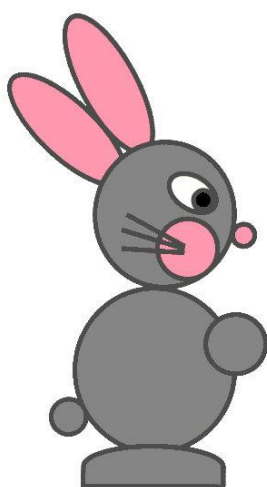
22)



23)



24)

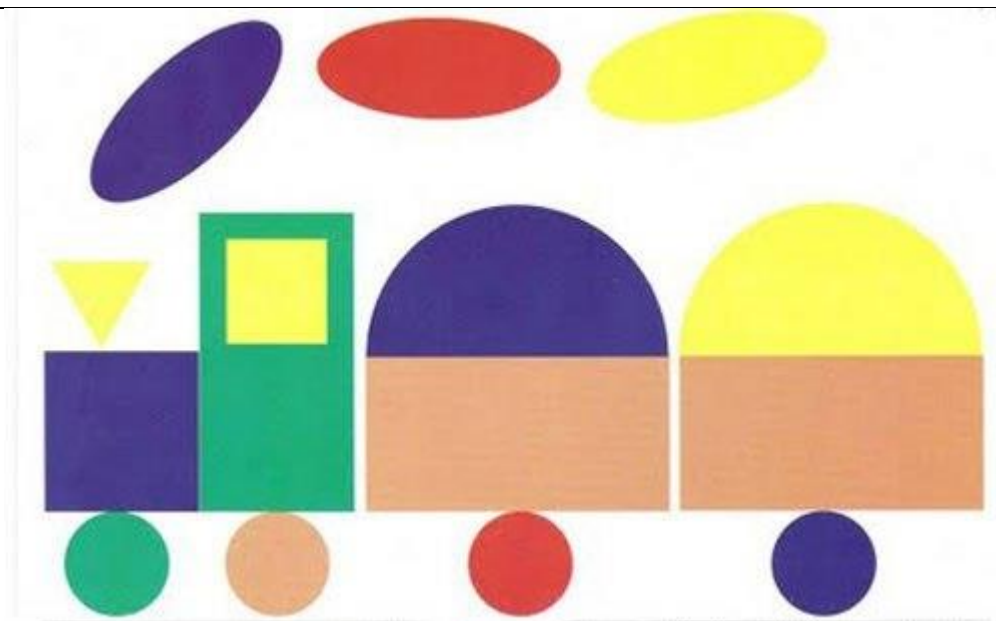


novye-deti.ru

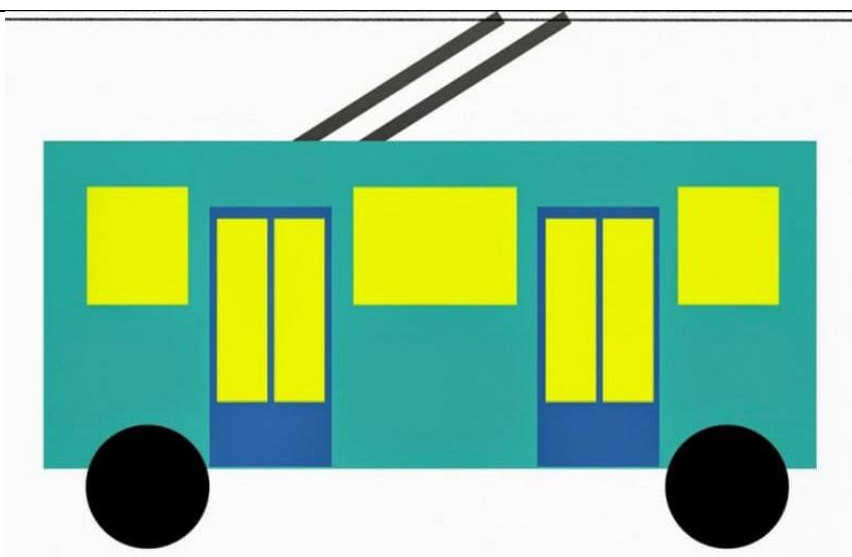
28)



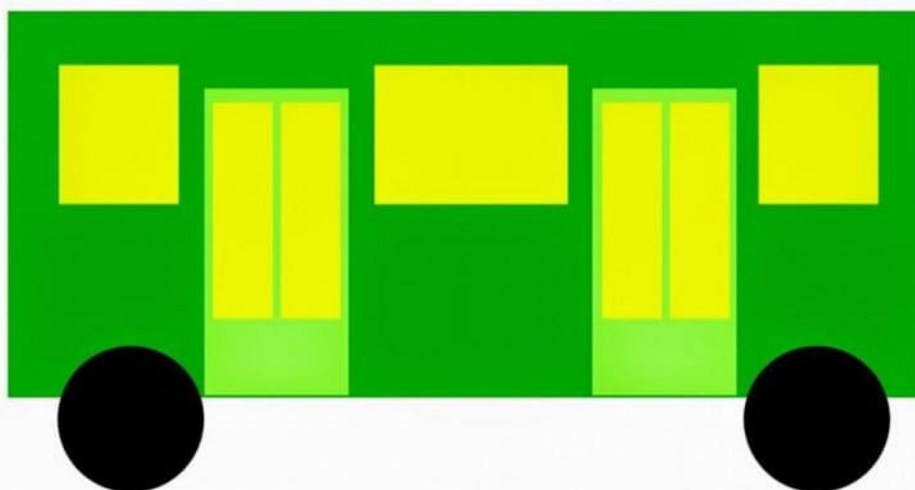
29)



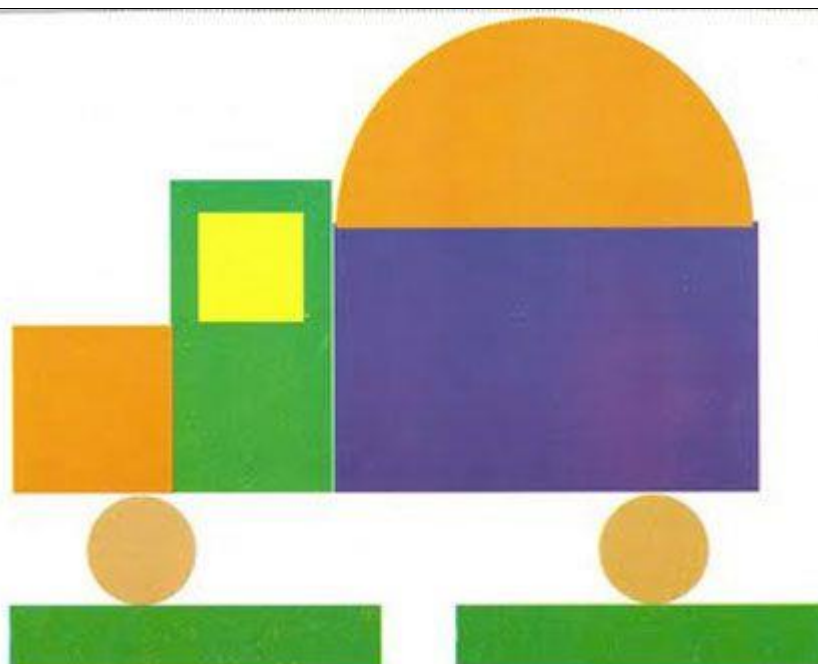
30)



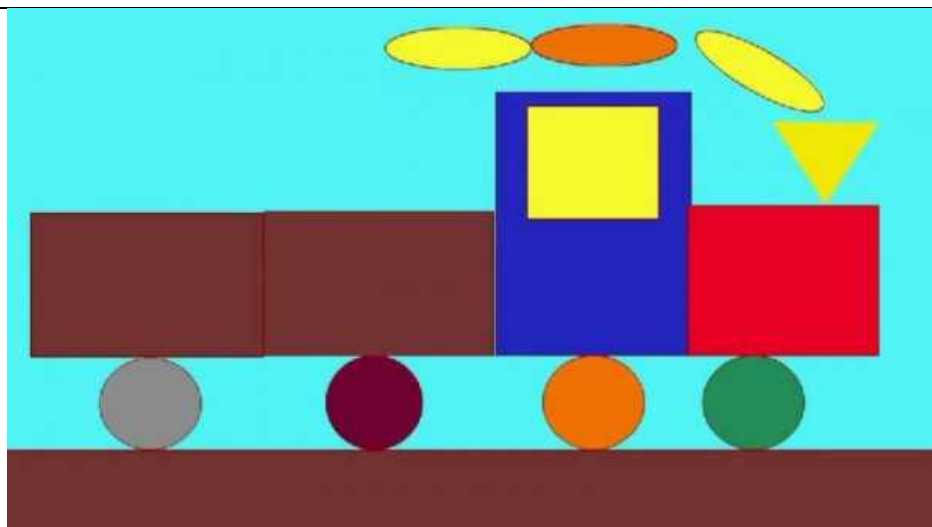
31)

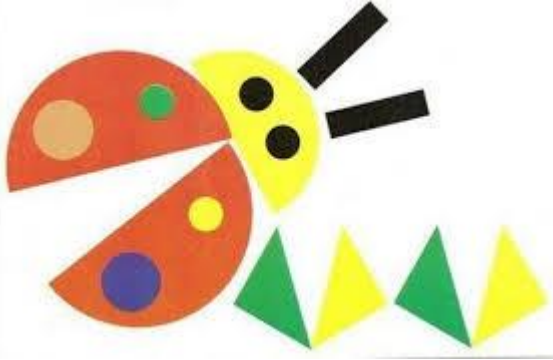
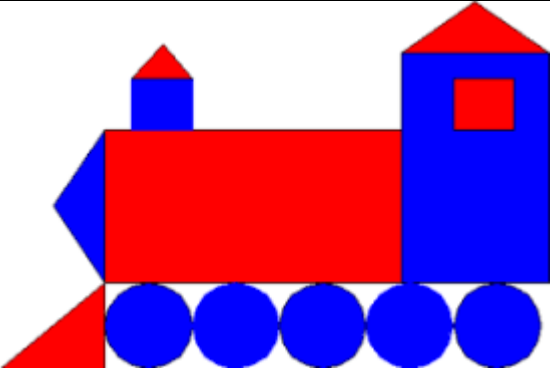



32)



33)

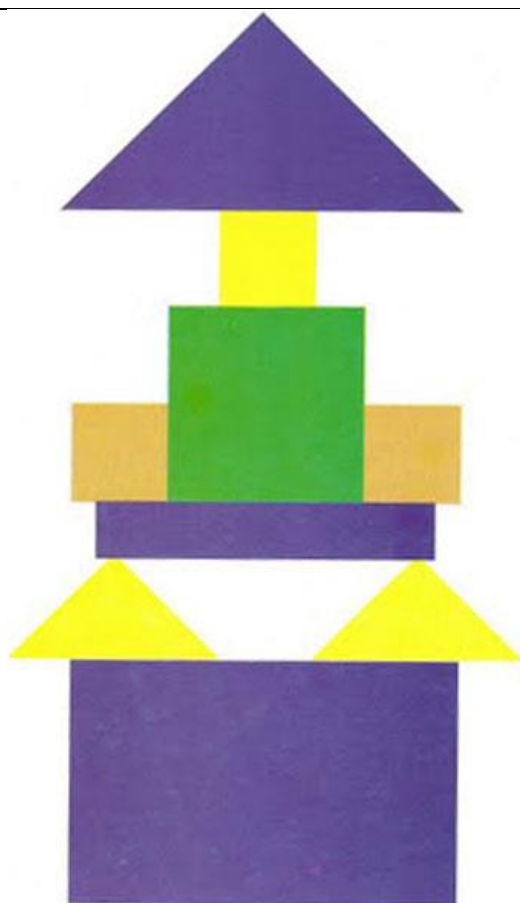


34)	
35)	
36)	<p data-bbox="1069 952 1236 974">Веселая гусеница</p>  <p data-bbox="475 1406 683 1429">http://www.bestkids.ru</p>

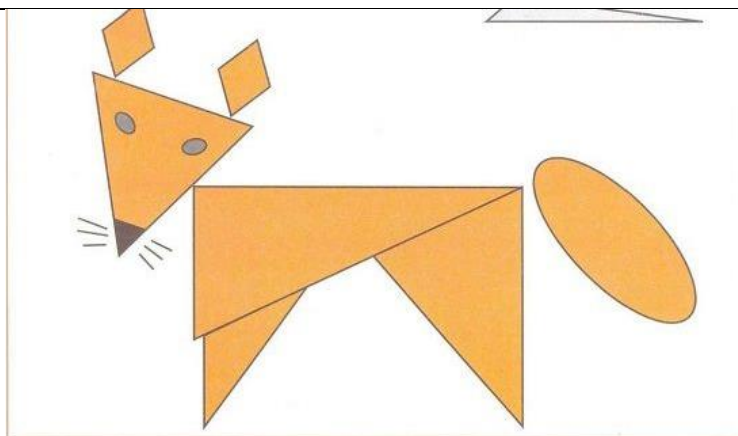
37)



38)



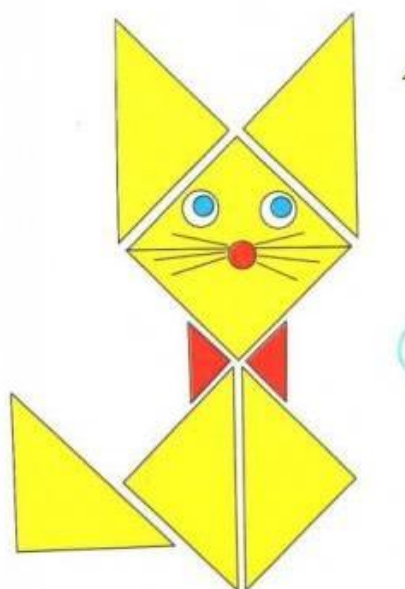
39)



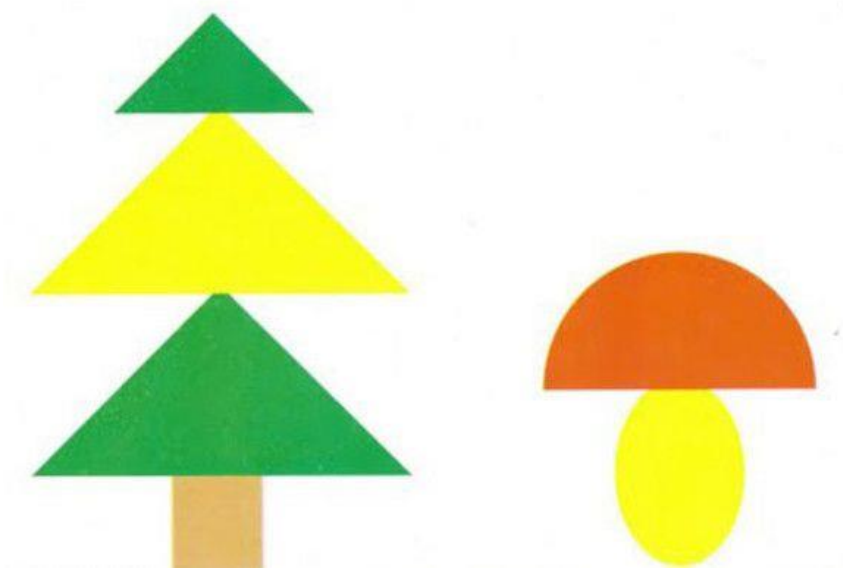


40)



41)

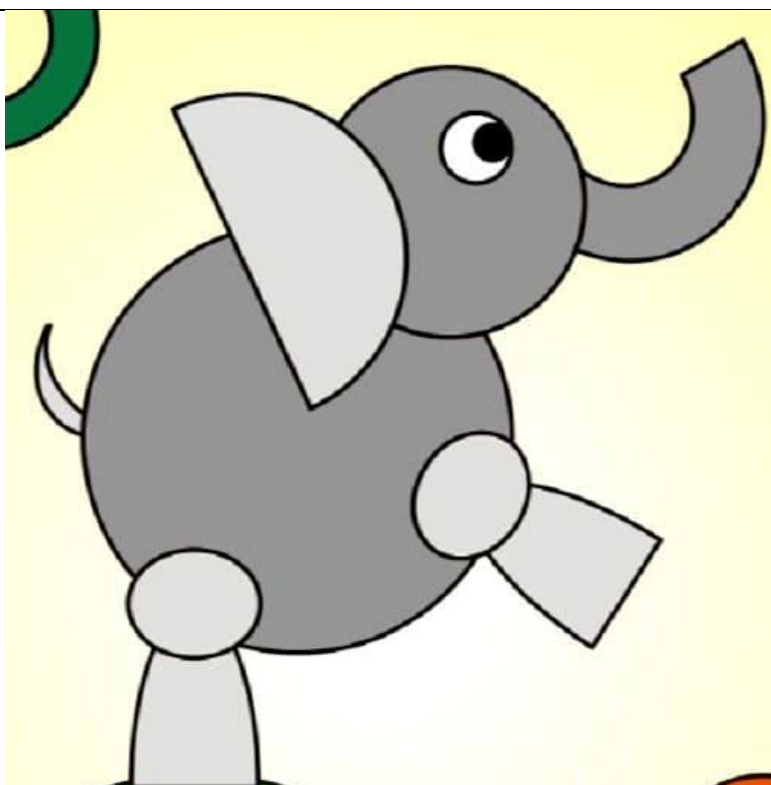


<p>42)</p>	
<p>43)</p>	
<p>44)</p>	

45)



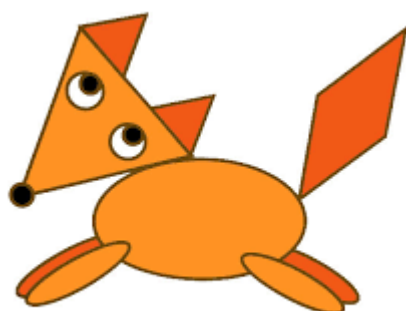
46)



47)



48)

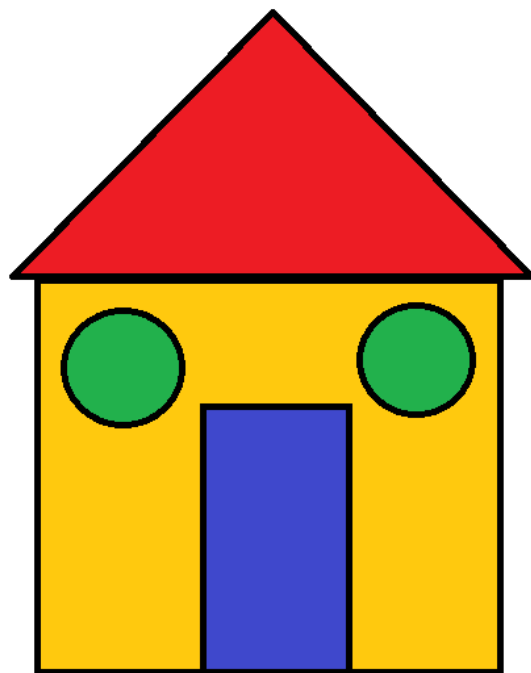


nove-deti.ru

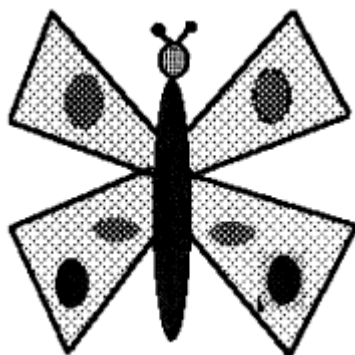
49)



50)



51)



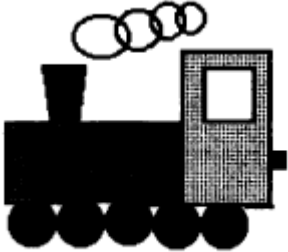

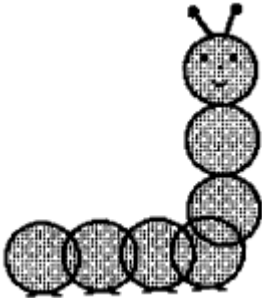



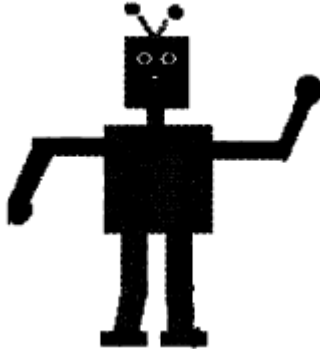
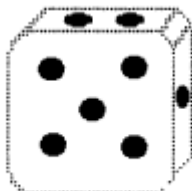
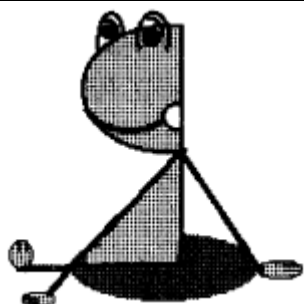
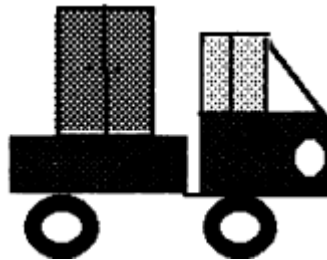
52)

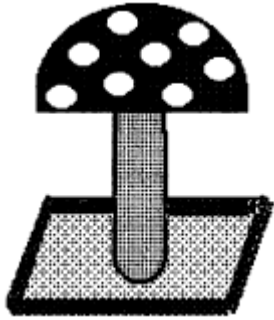


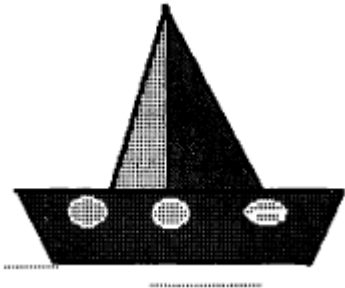



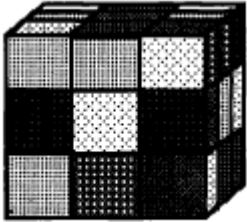
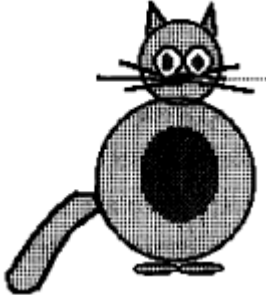


53)



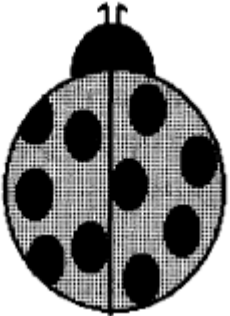


54)	
55)	
56)	
57)	
58)	

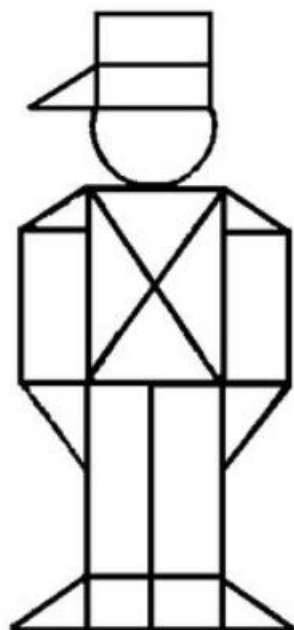
59)	
60)	
61)	
62)	
63)	

64)	
65)	
66)	
67)	
68)	

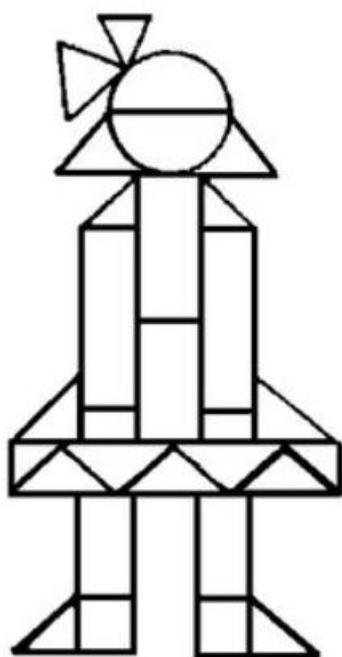
69)	
70)	
71)	
72)	

73)	
74)	
75)	

76)



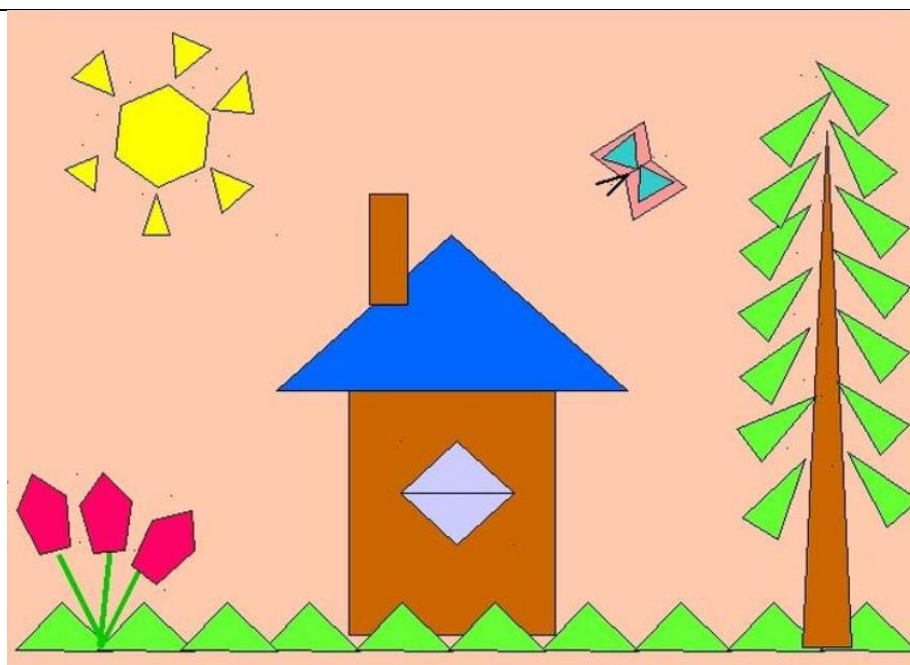
77)



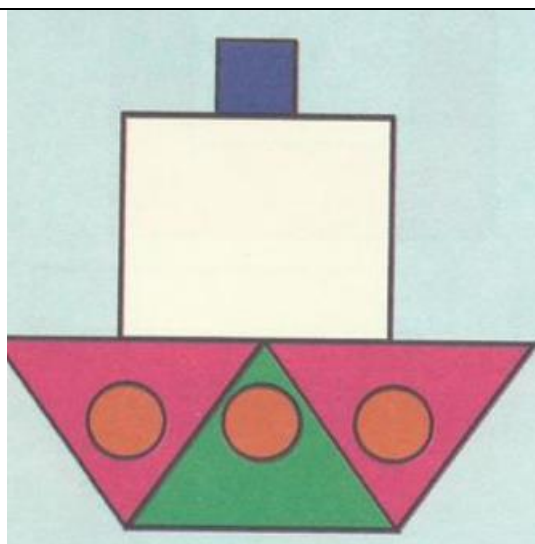
78)



79)



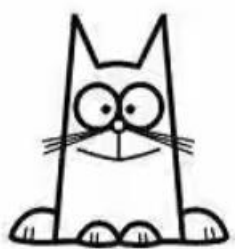
80)



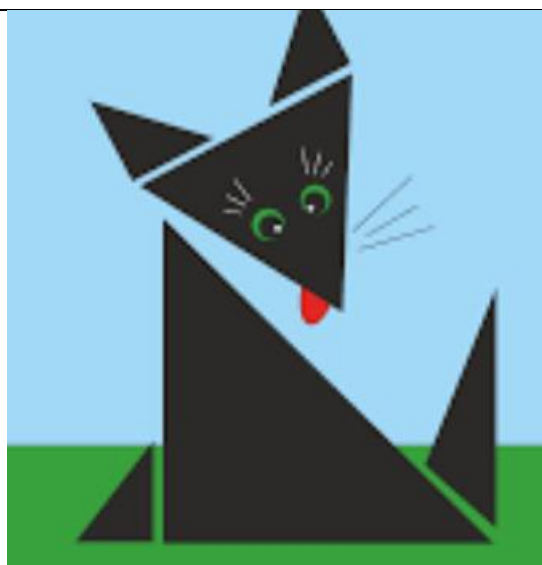
81)

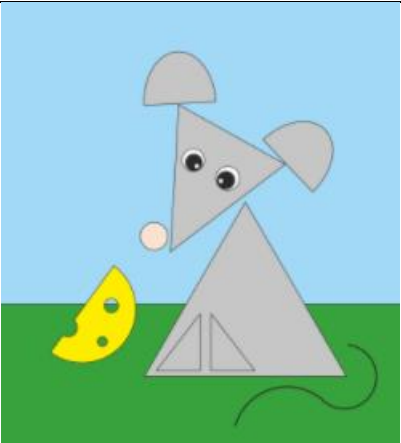






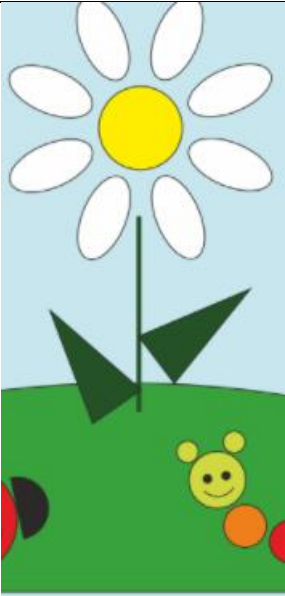
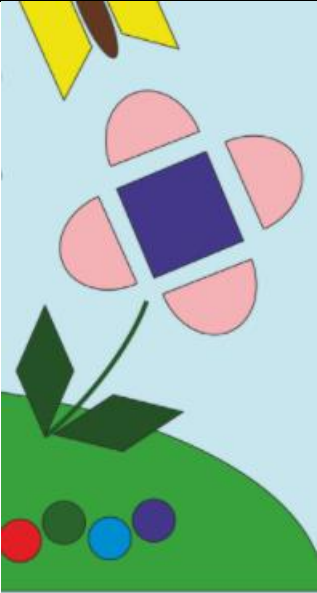
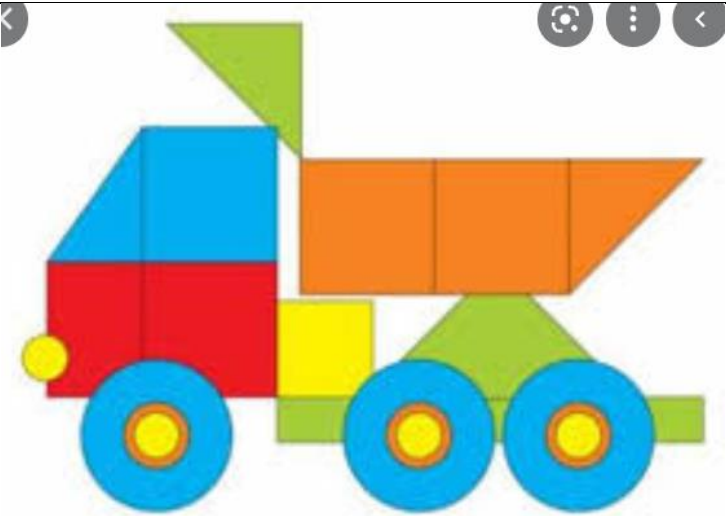
82)

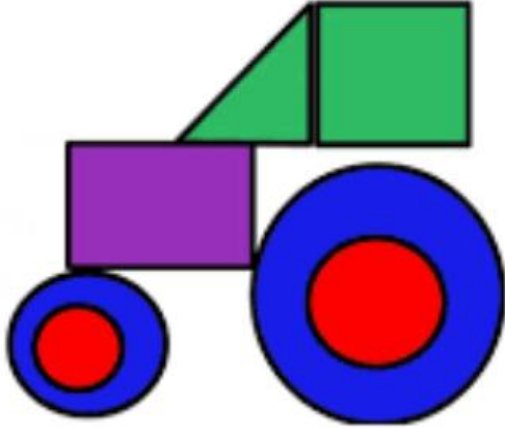


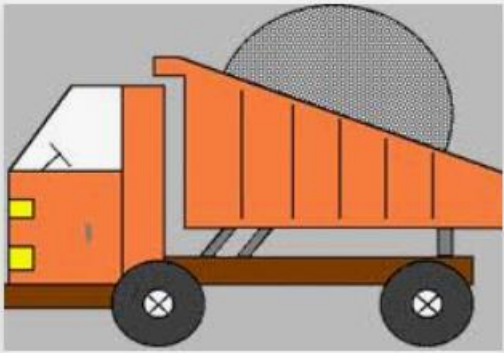






83)



84)		
85)		
86)		
87)		
88)		

89)	
90)	
91)	

92)		
93)		
94)		
95)		

96)	
97)	
98)	
99)	

100)

