

[Описание программы](#)

[Код программы](#)

[Код с подробными комментариями](#)

[Советы по улучшению и расширению программы](#)

### **Описание программы**

Данная программа рисует три концентрические окружности и закрашивает пространство между ними. Для этой цели применяется метод рекурсии. Реализован он следующим образом: в качестве аргументов функции передаются координаты точки. Функция проверяет, является ли цвет этой точки цветом фона, и, стало быть, должна ли она быть закрашена. Если да, то точка закрашивается установленным цветом. После этого функция вызывает сама себя для соседних четырех точек. Те же самые действия, следовательно, производятся и над ними. Если эти точки нужно закрасить, то это проделывается, и функция вновь вызывается для точек соседних с этими, и так далее. Таким образом, происходит закрашивание всех точек непрерывной области одного цвета. Ограничителями этой области в данном случае служат те самые окружности. Чтобы соединить кольца между окружностями в непрерывную область, из всех них, кроме внешней, удаляется по одной точке.

Обратите внимание, что при увеличении радиуса кругов и, следовательно, размеров закрашиваемой области, ресурсов системы может не хватить и программа аварийно завершится, так что даже перед первой компиляцией и запуском программы **настоятельно рекомендуется** (как, впрочем, в случае любой программы) сохраниться.

### **Код программы**

```
#include <graphics.h>
#include <stdlib.h>
#include <stdio.h>
#include <conio.h>
#include <dos.h>

void Fill(int X, int Y)
{
    if (getpixel(X, Y) == BLACK)
    {
        delay(1);
        putpixel(X, Y, RED);
        Fill(X - 1, Y);
        Fill(X, Y - 1);
    }
}
```

## Закрашивание области

Автор: Андрей

31.03.2009 15:18 - Обновлено 04.07.2009 13:50

---

```
    Fill(X + 1, Y);
    Fill(X, Y + 1);
  }
}
```

```
void main()
```

```
{
```

```
GraphInit\(\);
```

```
circle(60, 60, 50);
```

```
circle(60, 60, 35);
```

```
circle(60, 60, 20);
```

```
setcolor(BLACK);
```

```
line(11, 60, 60, 60);
```

```
Fill(60, 60);
```

```
getch();
```

```
}
```

### ***Код программы с комментариями***

```
#include <graphics.h> //Подключаем графическую библиотеку
#include <stdlib.h> //Библиотека содержит функцию exit (см. GraphInit)
#include <stdio.h> //Библиотека содержит функцию printf
//(см. GraphInit)
#include <conio.h> //Библиотека содержит функцию getch
#include <dos.h> //Библиотека содержит функцию delay

void Fill(int X, int Y) //Функция, выполняющая основную работу,
//аргументы - координаты точки
{
    if (getpixel(X, Y) == BLACK) //Является ли точка подлежащей
//закрашиванию
    {
        delay(1); //Выполнение программы приостанавливается
//на небольшое время, закрашивание происходит у вас на глазах
putpixel(X, Y, RED); //Закрашиваем точку
Fill(X - 1, Y); //Вызываем эту же функцию для четырех соседних точек
Fill(X, Y - 1);
Fill(X + 1, Y);
Fill(X, Y + 1);
    }
}
```

## Закрашивание области

Автор: Андрей

31.03.2009 15:18 - Обновлено 04.07.2009 13:50

---

```
}

void main()
{
GraphInit\(\); //Текст функции расположен по ссылке.
//Функция инициализирует графику

circle(60, 60, 50); //Чертим три концентрические окружности
circle(60, 60, 35);
circle(60, 60, 20);

setcolor(BLACK); //Устанавливаем цвет, которым будем рисовать,
//на черный, и...
line(11, 60, 60, 60); //...чертим линию фонового цвета,
//чтобы удалить по одной точке из двух окружностей

Fill(60, 60); //Начинаем закрашивание с центра

getch(); //Ждем нажатия клавиши перед завершением программы
}
```

### ***Советы по улучшению и расширению программы***

Прежде всего, область можно закрашивать не монохромным цветом, а, например, градиентом. Для этого необходимо определить функцию, которая будет определять необходимый цвет точки по передаваемым ей координатам и после этого закрашивать ее этим цветом. Вместо `putpixel`, соответственно надо будет вызывать именно эту функцию.

Однако, не забывайте, что такое расширение делает программу еще более требовательной к ресурсам!

Другое направление развития идеи данной программы заключается в том, что можно не только закрашивать какую-то область неправильной формы, но и создавать аналогичным образом выделение, например в графическом редакторе. Для этого необходимо создать альфа-канал - по сути двумерный массив тех же размеров, что и редактируемое изображение (в точках). В этом массиве хранится черно-белое изображение, при этом белый цвет точки массива будет означать, что точка изображения с соответствующими координатами помечена как выделенная, а черный - что точка изображения не выделена.

Рекурсивная функция при этом выделит непрерывную область произвольной формы по заданному алгоритму (например, область изображения с близкими цветами). Для осуществления этого нужно в операторе `if` проверять два условия - соответствует ли данная точка правилу по которым происходит выделение (например, близок ли ее цвет некоему заданному), а также не помечена ли она уже как выделенная (то есть, не

## Закрашивание области

Автор: Андрей

31.03.2009 15:18 - Обновлено 04.07.2009 13:50

---

является ли цвет точки альфа-канала белым). И если необходимые условия выполняются, то точка *принадлежащая альфа-каналу* (не редактируемому изображению!) помечается как выделенная.

Заметим, что альфа-канал может быть не только двухцветным, но и быть в оттенках серого - серый цвет будет соответствовать частичному выделению точки изображения.

Конечно же, вы не будете создавать в среде разработки, выпущенной в 1992м году, сколько-нибудь функциональный графический редактор, в котором будете использовать выделения неправильной формы. Однако, идея, реализованная этой небольшой программой, имеет неплохие перспективы при наличии более мощных инструментов разработки.