

Основы программирования

Лабораторная работа №7

WinAPI Графика – Rectangle, Ellipse,
Кисти

Власенко Олег Федосович

Рисование прямоугольника

Простейшая функция, с помощью которой можно нарисовать прямоугольник, называется Rectangle :

```
BOOL WINAPI Rectangle(  
    HDC hdc, // идентификатор контекста отображения  
    int nxTL, // координата x верхнего левого угла  
    int nyTL, // координата y верхнего левого угла  
    int nxBR, // координата x правого нижнего угла  
    int nyBR); // координата y правого нижнего угла
```

Функция Rectangle рисует прямоугольник для контекста отображения hdc, возвращая значение TRUE в случае успеха или FALSE при ошибке.

Назначение остальных параметров иллюстрируется рис. 2.17.

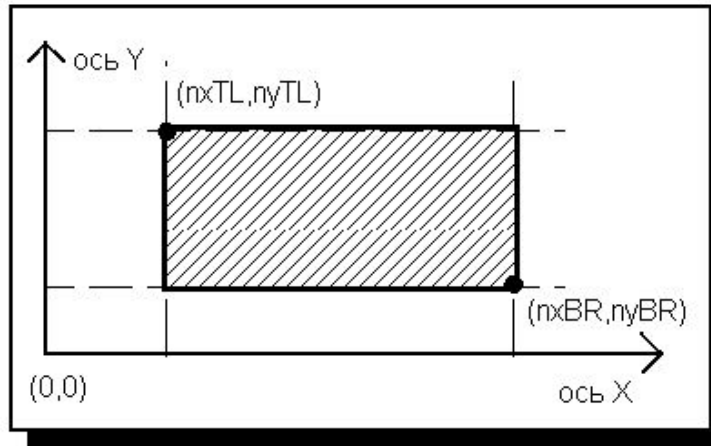


Рис. 2.17. Рисование прямоугольника

https://www.frolov-lib.ru/books/bsp/v14/ch2_3.htm

Функция FillRect закрашивает прямоугольную область окна заданной кистью:

```
int WINAPI FillRect(  
    HDC hdc, // идентификатор контекста отображения  
    const RECT FAR* lprc, // указатель на структуру RECT  
    HBRUSH hbrush); // идентификатор кисти для закрашивания
```

Параметр lprc должен указывать на структуру типа RECT, в которую следует записать координаты закрашиваемой прямоугольной области. Правая и нижняя граница указанной области не закрашивается.

Независимо от того, какая кисть выбрана в контекст отображения, функция FillRect будет использовать для закрашки кисть, указанную параметром hbrush.

Учтите, что правильная работа функции FillRect гарантируется только в том случае, когда значение поля bottom структуры RECT больше значения поля top, а значение поля right больше значения поля left.

Для закрашивания границы прямоугольной области (т. е. для рисования прямоугольной рамки) можно использовать функцию FrameRect :

```
int WINAPI FrameRect(  
    HDC hdc, // идентификатор контекста отображения  
    const RECT FAR* lprc, // указатель на структуру RECT  
    HBRUSH hbrush); // идентификатор кисти для закрашивания
```

Параметры этой функции аналогичны параметрам функции FillRect.

Ширина пера, используемого для рисования рамки, всегда равна одной логической единице. Структура RECT должна быть подготовлена таким же образом, что и для функции FillRect, т. е. значение поля bottom структуры RECT должно быть больше значения поля top, а значение поля right - больше значения поля left.

Значение, возвращаемое функциями FillRect и FrameRect не используется, приложения должны его игнорировать.

Рисование эллипса

Для рисования эллипса вы можете использовать функцию Ellipse :

```
BOOL WINAPI Ellipse(  
    HDC hdc,    // идентификатор контекста отображения  
    int  nxTL,  // координата x верхнего левого угла  
    int  nyTL,  // координата y верхнего левого угла  
    int  nxBR,  // координата x правого нижнего угла  
    int  nyBR); // координата y правого нижнего угла
```

Первый параметр этой функции указывает идентификатор контекста отображения, остальные - координаты верхнего левого и правого нижнего углов прямоугольника, в который должен быть вписан эллипс (рис. 2.19).

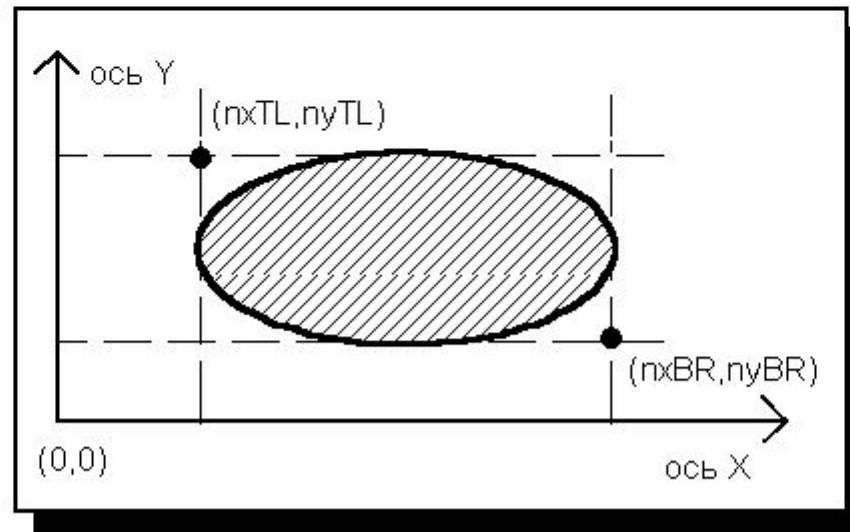
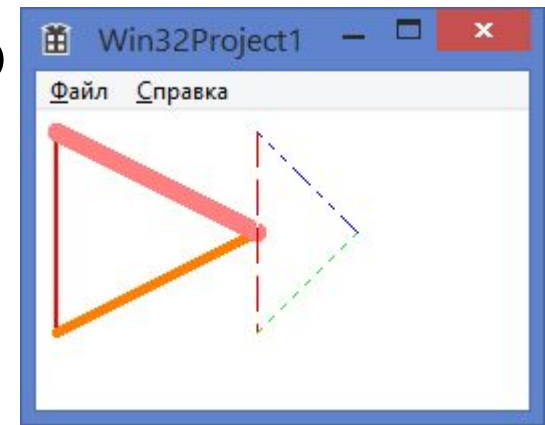


Рис. 2.19. Рисование эллипса

Такое разное перо

```
case WM_PAINT: {  
    PAINTSTRUCT ps;  
    HDC hdc = BeginPaint(hWnd, &ps);  
    HPEN hPen;  
    hPen = CreatePen(PS_SOLID, 2, RGB(255, 0, 0));  
    SelectObject(hdc, hPen);  
    MoveToEx(hdc, 10, 10, NULL);  
    LineTo(hdc, 10, 110);  
  
    hPen = CreatePen(PS_SOLID, 5, RGB(255, 128, 0));  
    SelectObject(hdc, hPen);  
    LineTo(hdc, 110, 60);  
  
    hPen = CreatePen(PS_SOLID, 10, RGB(255, 128, 128));  
    SelectObject(hdc, hPen);  
    LineTo(hdc, 10, 10);
```



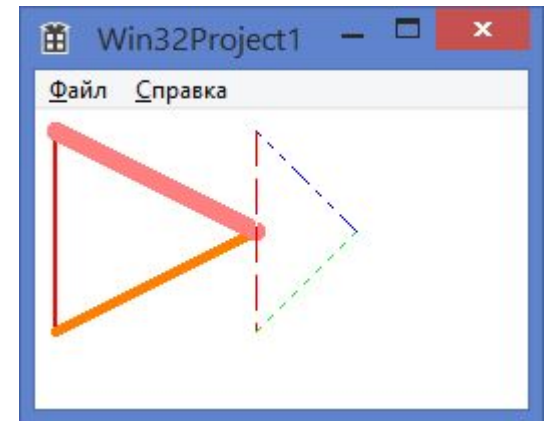
Такое разное перо (2)

```
hPen = CreatePen(PS_DASH, 1, RGB(255, 0, 0));  
SelectObject(hdc, hPen);  
MoveToEx(hdc, 110, 10, NULL);  
LineTo(hdc, 110, 110);
```

```
hPen = CreatePen(PS_DOT, 1, RGB(0, 255, 0));  
SelectObject(hdc, hPen);  
LineTo(hdc, 160, 60);
```

```
hPen = CreatePen(PS_DASHDOTDOT, 1, RGB(0, 0, 255));  
SelectObject(hdc, hPen);  
LineTo(hdc, 110, 10);  
DeleteObject(hPen);
```

```
EndPoint(hWnd, &ps);  
}
```



Кисть и FillRect

```
case WM_PAINT: {  
    PAINTSTRUCT ps;  
    HDC hdc = BeginPaint(hWnd, &ps);
```

```
    HBRUSH hBrush;
```

```
    hBrush = CreateSolidBrush(RGB(0, 0, 255));
```

```
    SelectObject(hdc, hBrush);
```

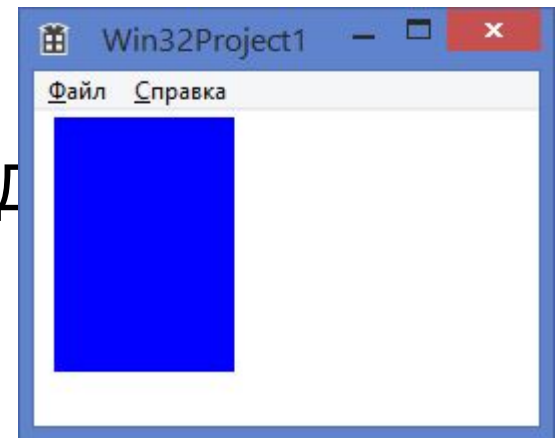
```
    RECT rect = { 10, 3, 100, 130 };
```

```
    FillRect(hdc, &rect, hBrush);
```

```
// СЛЕДУЮЩИЙ КОД ВСТАВИТЬ СЮДА
```

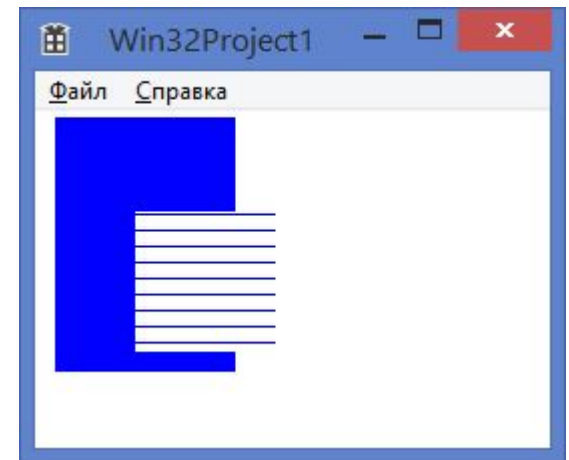
```
    EndPaint(hWnd, &ps);
```

```
}
```



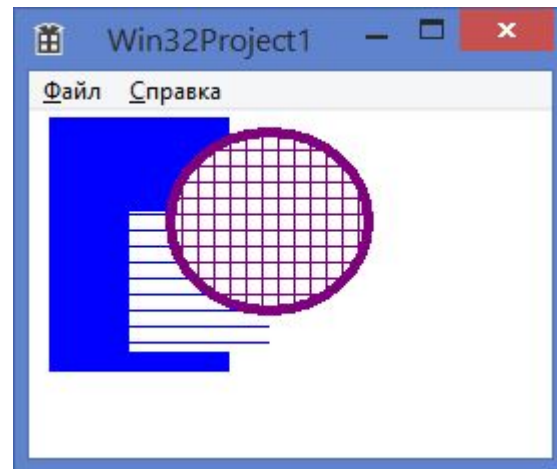
Кисть (2)

```
hBrush = CreateHatchBrush(HS_HORIZONTAL, RGB(0, 0, 255));  
SelectObject(hdc, hBrush);  
RECT rect2 = { 50, 50, 120, 120 };  
FillRect(hdc, &rect2, hBrush);
```



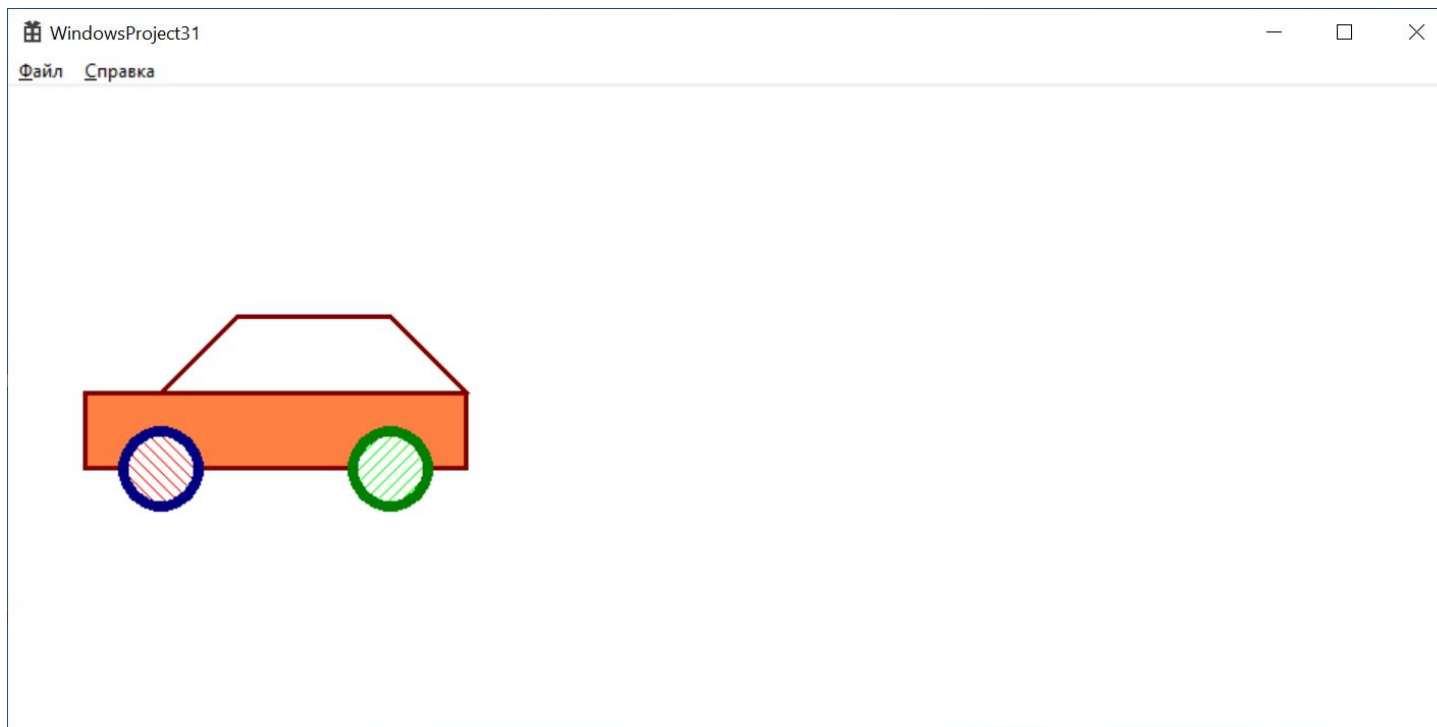
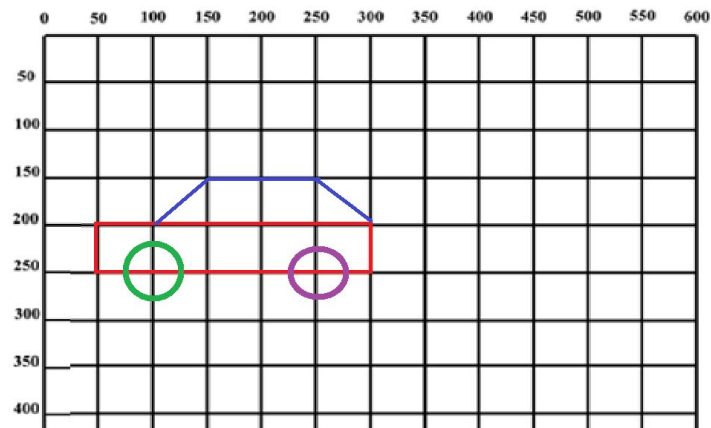
Кисть (3)

```
hBrush = CreateHatchBrush(HS_CROSS, RGB(128, 0, 128));  
SelectObject(hdc, hBrush);  
HPEN hPen;  
hPen = CreatePen(PS_SOLID, 5, RGB(128, 0, 128));  
SelectObject(hdc, hPen);  
Ellipse( hdc, 70, 10, 170, 100);  
  
DeleteObject(hBrush);
```



Задача 1

Нарисовать автомобиль
Использовать 3 разных кисти.
В рисунке есть 2 круга.



Задача 1

Нарисовать автомобиль
Использовать 3 разных кисти.
В рисунке есть 2 круга.

```
case WM_PAINT:
```

```
{
```

```
    PAINTSTRUCT ps;
```

```
    HDC hdc = BeginPaint(hWnd, &ps);
```

```
    // TODO: Добавьте сюда любой код прорисовки, использующий HDC...
```

```
    HPEN hPen = CreatePen(PS_SOLID, 3, RGB(128, 0, 0));
```

```
    SelectObject(hdc, hPen);
```

```
    HBRUSH hBrush; //создаём объект-кисть
```

```
    hBrush = CreateSolidBrush(RGB(255, 128, 67)); //задаём сплошную кисть, закрашенную цветом RGB
```

```
    SelectObject(hdc, hBrush);
```

```
    Rectangle(hdc, 50, 200, 300, 250);
```

```
    MoveToEx(hdc, 100, 200, NULL);
```

```
    LineTo(hdc, 150, 150);
```

```
    LineTo(hdc, 250, 150);
```

```
    LineTo(hdc, 300, 200);
```

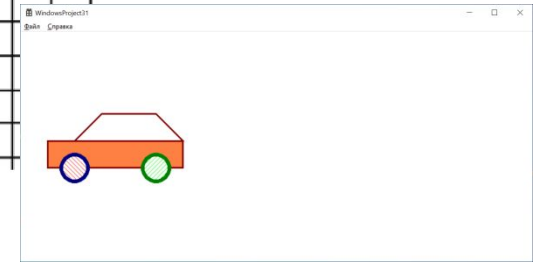
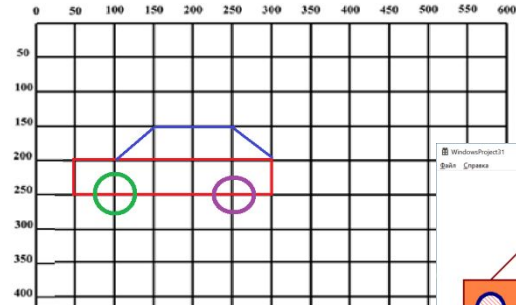
```
    hPen = CreatePen(PS_SOLID, 7, RGB(0, 0, 128));
```

```
    SelectObject(hdc, hPen);
```

```
    hBrush = CreateHatchBrush(HS_FDIAGONAL, RGB(255, 0, 0)); //задаём диагональную кисть, закрашенную цветом RGB
```

```
    SelectObject(hdc, hBrush);
```

```
    Ellipse(hdc, 75, 225, 125, 275);
```



Задача 2

Нарисовать снежную бабу.

В рисунке должно быть 3 закрашенных эллипса.

В качестве носа – морковка.

Глаза. Рот. Руки.

Источники информации

- **Графический интерфейс GDI в Microsoft Windows** (Александр Фролов, Григорий Фролов
Том 4, М.: Диалог-МИФИ, 1993)

https://www.frolov-lib.ru/books/bsp/v14/ch2_3.htm

- **КАК рисовать в Win32 API?**
-<http://radiofront.narod.ru/htm/prog/htm/winda/api/paint.html>

Домашнее задание

1. К картине, созданной в ходе лабораторных работ 5 и 6, добавить рисунок транспортного средства с колесами.
2. Добавить сооружение, растение, или что-угодно иное, содержащее закрашенные прямоугольники. Нужно использовать максимальное количество разных кистей и перьев.

Домашнее задание – оформление

Для сдачи домашней работы нужно иметь:

- 1) код программы
- 2) Расчет картинки – на бумаге (на отдельном листе или в тетради)
- 3) Скрин шот выполнения программы

Срок выполнения – до следующей встречи на лабораторной работе

Если болел/не мог – это становится «долгом»