

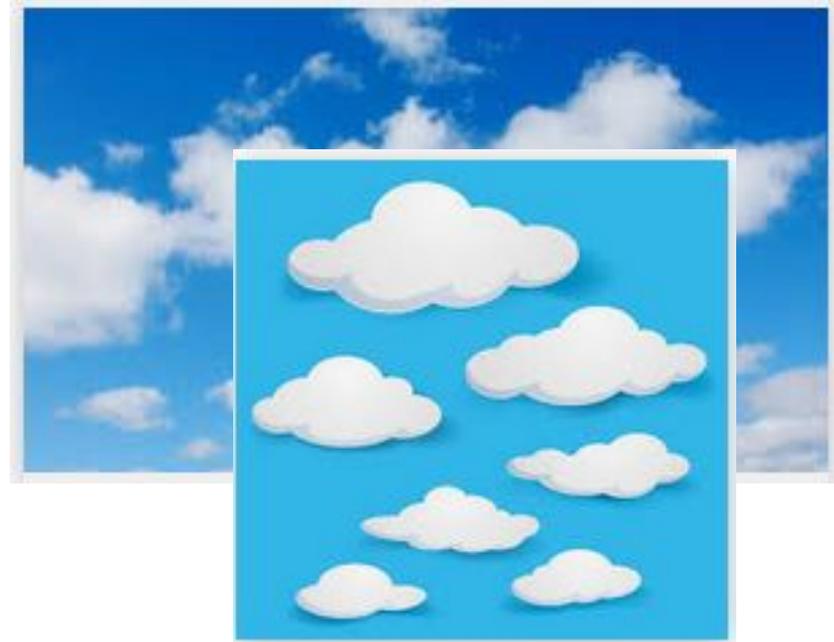
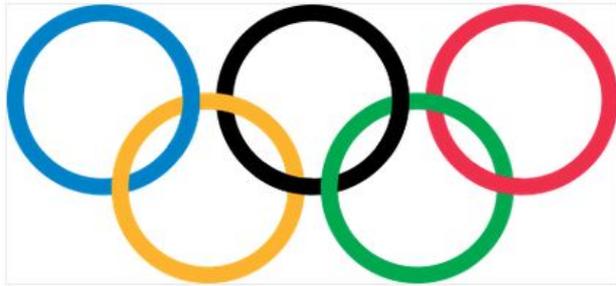
Основы программирования

Лабораторная работа №8

WinAPI Графика – Знакомство с
функциями,
Ellipse, Arc, Chord, Pie

Власенко Олег Федосович

Что будем рисовать



Чем будем рисовать

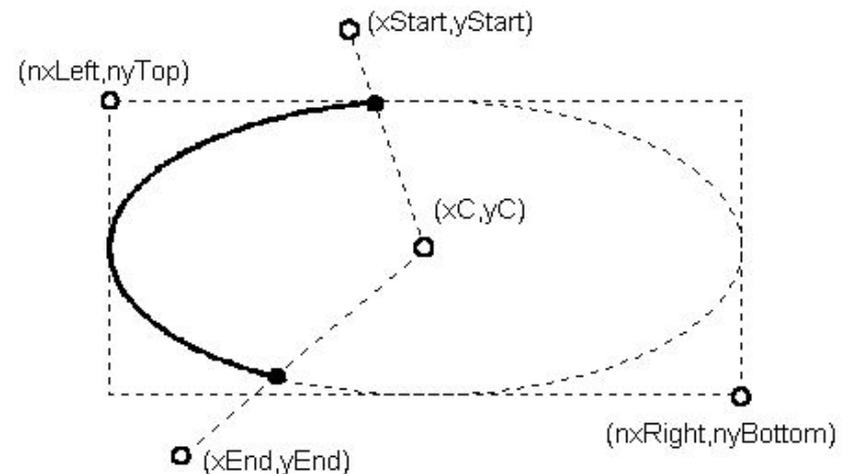
https://www.frolov-lib.ru/books/bsp/v14/ch2_3.htm

Рисование дуги эллипса

функция Arc позволяет нарисовать дугу эллипса или окружности:

```
BOOL WINAPI Arc(  
    HDC hdc, // идентификатор контекста отображения  
    int nXLeft, int nYTop, // верхий левый угол  
    int nXRight, int nYBottom, // правый нижний угол  
    int nXStart, int nYStart, // начало дуги  
    int nXEnd, int nYEnd); // конец дуги
```

Первый параметр этой функции определяет контекст отображения, в котором будет нарисована дуга.



Чем будем рисовать

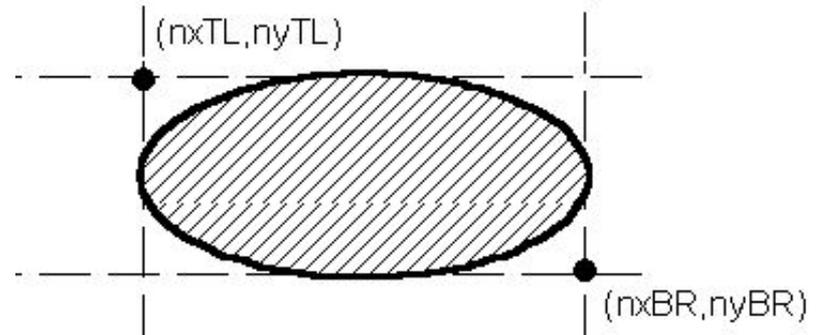
https://www.frolov-lib.ru/books/bsp/v14/ch2_3.htm

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



Чем будем рисовать

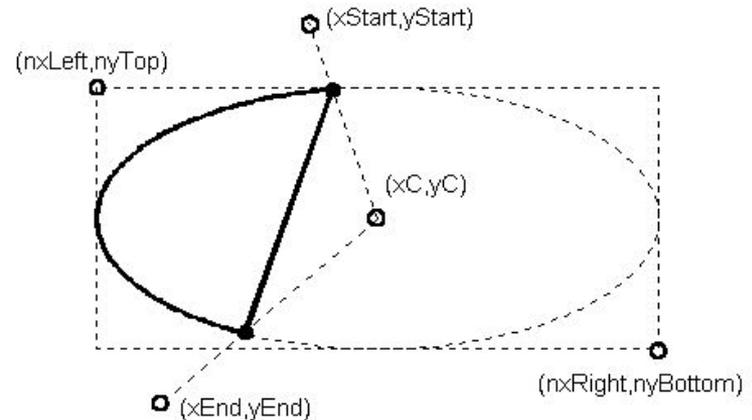
https://www.frolov-lib.ru/books/bsp/v14/ch2_3.htm

Рисование сегмента эллипса

Сегмент эллипса можно нарисовать при помощи функции Chord :

```
BOOL WINAPI Chord(  
    HDC hdc, // идентификатор контекста отображения  
    int nxLeft, int nyTop, // верхий левый угол  
    int nxRight, int nyBottom, // правый нижний угол  
    int nxStart, int nyStart, // начало дуги  
    int nxEnd, int nyEnd); // конец дуги
```

Параметры этой функции аналогичны параметрам рассмотренной нами ранее функции Arc.



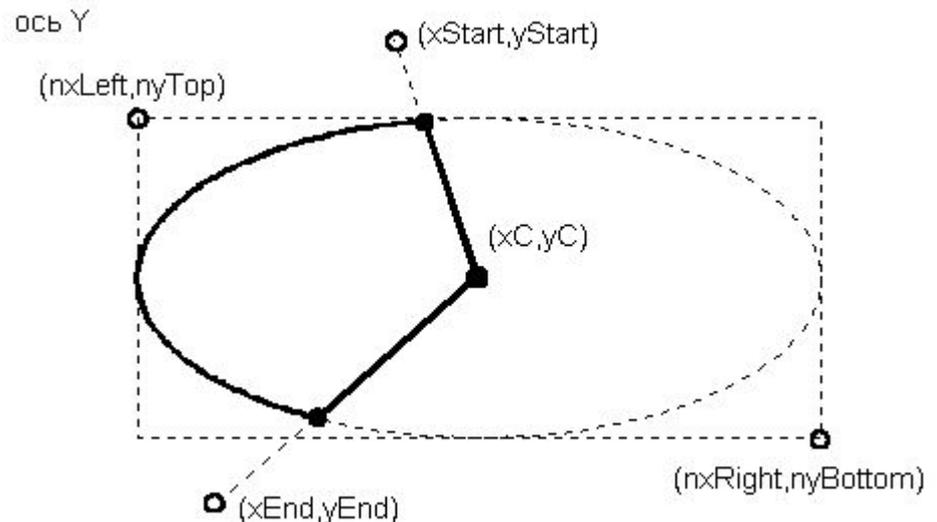
Чем будем рисовать

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

Рисование сектора эллипса

Для рисования сектора эллипса следует использовать функцию Pie ,
аналогичную по своим параметрам функциям Arc и Chord:

```
BOOL WINAPI Pie(  
    HDC hdc, // идентификатор контекста отображения  
    int nxLeft, int nyTop, // верхний левый угол  
    int nxRight, int nyBottom, // правый нижний угол  
    int nxStart, int nyStart, // начало дуги  
    int nxEnd, int nyEnd); // конец дуги
```



Где будем рисовать

Для отрисовки каждого предмета будем использовать отдельную функцию

Задача 1

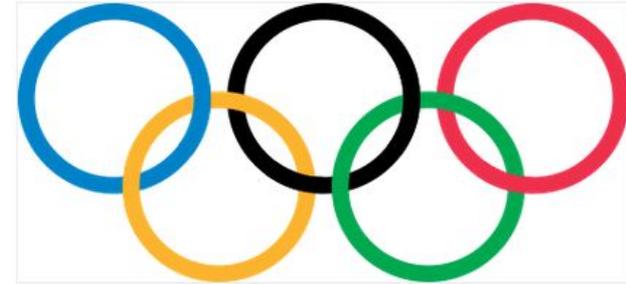
Нарисовать олимпийскую эмблему – настолько близко к оригиналу, насколько это возможно.

Отрисовку эмблемы олимпиады сделать в отдельной функции.



Задача 1

Нарисовать олимпийскую эмблему – настолько близко к оригиналу, насколько это возможно.



```
126 void OlympicEmblem(HDC hdc) {
127
128
129
130 }
131
132
133 //
134 // ФУНКЦИЯ: WndProc(HWND, UINT, WPARAM, LPARAM)
135 //
136 // ЦЕЛЬ: Обрабатывает сообщения в главном окне.
137 //
138 // WM_COMMAND - обработать меню приложения
139 // WM_PAINT   - Отрисовка главного окна
140 // WM_DESTROY - отправить сообщение о выходе и вернуться
141 //
142 //
143 LRESULT CALLBACK WndProc(HWND hWnd, UINT message, WPARAM wParam, LPARAM lParam)
144 {
145     switch (message)
```

Задача 1

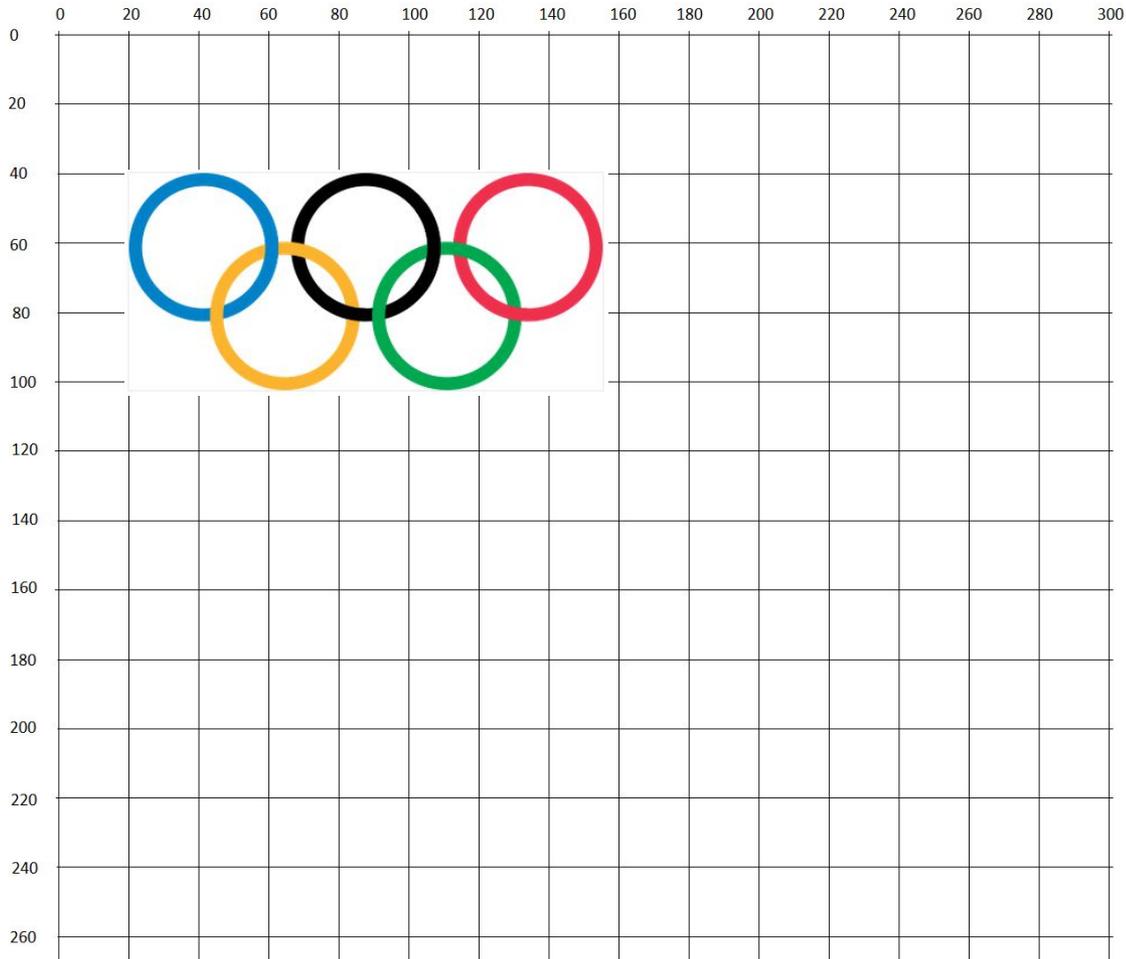
Нарисовать олимпийскую эмблему – настолько близко к оригиналу, насколько это возможно.



```
126 void OlympicEmblem(HDC hdc) {
127
128
129
130 }
131
132
133 //
134 // ФУНКЦИЯ: WndProc(HWND, UINT, WPARAM, LPARAM)
135 //
136 // ЦЕЛЬ: Обрабатывает сообщения в главном окне.
137 //
138 // WM_COMMAND - обработать меню приложения
139 // WM_PAINT   - Отрисовка главного окна
140 // WM_DESTROY - отправить сообщение о выходе и вернуться
141 //
142 //
143 LRESULT CALLBACK WndProc(HWND hWnd, UINT message, WPARAM wParam, LPARAM lParam)
144 {
145     switch (message)
146     {
147     case WM_PAINT:
148     {
149         PAINTSTRUCT ps;
150         HDC hdc = BeginPaint(hWnd, &ps);
151         // TODO: Добавьте сюда любой код прорисовки, использующий HDC...
152
153         OlympicEmblem(hdc);
154
155         EndPaint(hWnd, &ps);
156     }
157     }
158     break;
159 }
```

Задача 1

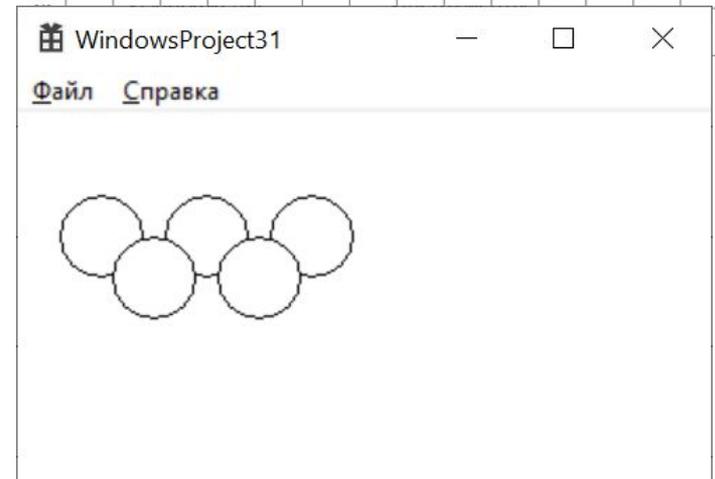
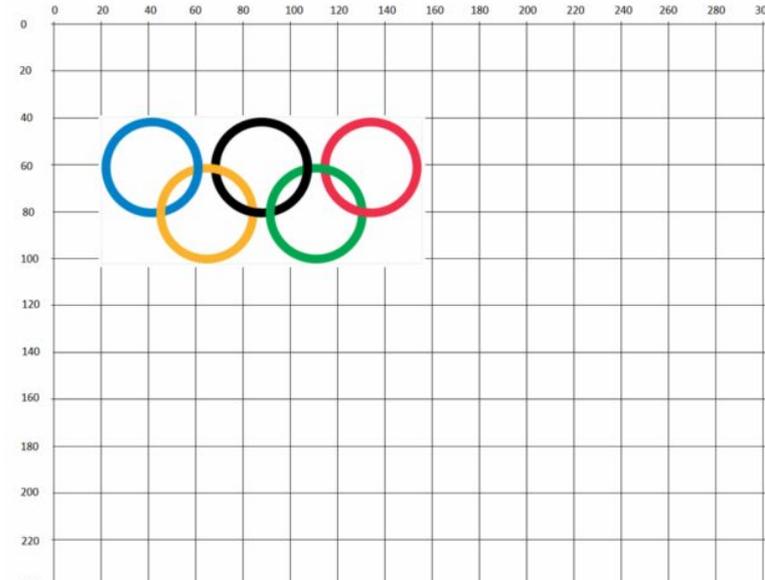
Нарисовать олимпийскую эмблему – настолько близко к оригиналу, насколько это возможно.



Задача 1

Нарисовать олимпийскую эмблему – настолько близко к оригиналу, насколько это возможно.

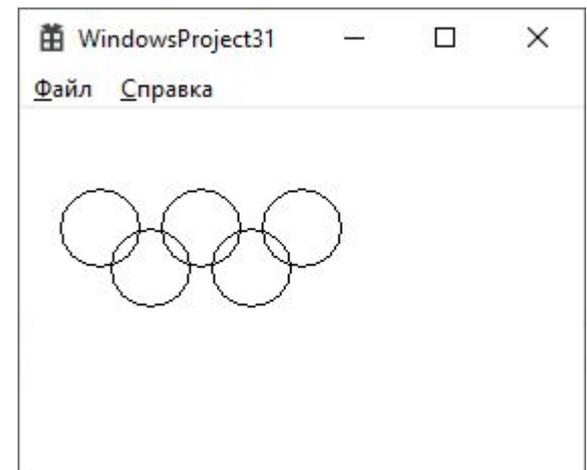
```
126 void OlympicEmblem(HDC hdc) {  
127     // верхний ряд колец  
128     Ellipse(hdc, 20, 40, 60, 80);  
129     Ellipse(hdc, 70, 40, 110, 80);  
130     Ellipse(hdc, 120, 40, 160, 80);  
131  
132     // нижний ряд колец  
133     Ellipse(hdc, 45, 60, 85, 100);  
134     Ellipse(hdc, 95, 60, 135, 100);  
135 }
```



Задача 1

Нарисовать олимпийскую эмблему – настолько близко к оригиналу, насколько это возможно.

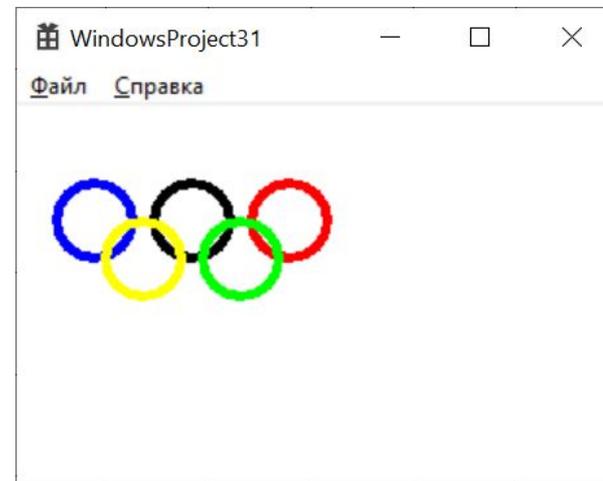
```
126 void OlympicEmblem(HDC hdc) {  
127     // Кисть - прозрачная  
128     SelectObject(hdc, GetStockObject(NULL_BRUSH));  
129  
130     // верхний ряд колец  
131     Ellipse(hdc, 20, 40, 60, 80);  
132     Ellipse(hdc, 70, 40, 110, 80);  
133     Ellipse(hdc, 120, 40, 160, 80);  
134  
135     // нижний ряд колец  
136     Ellipse(hdc, 45, 60, 85, 100);  
137     Ellipse(hdc, 95, 60, 135, 100);  
138  
139 }
```



Задача 1

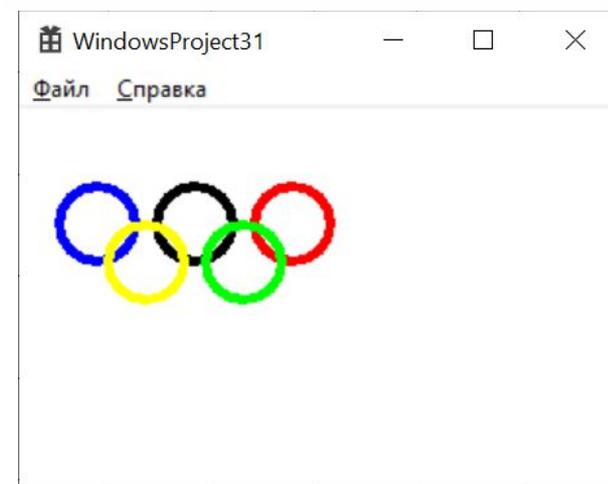
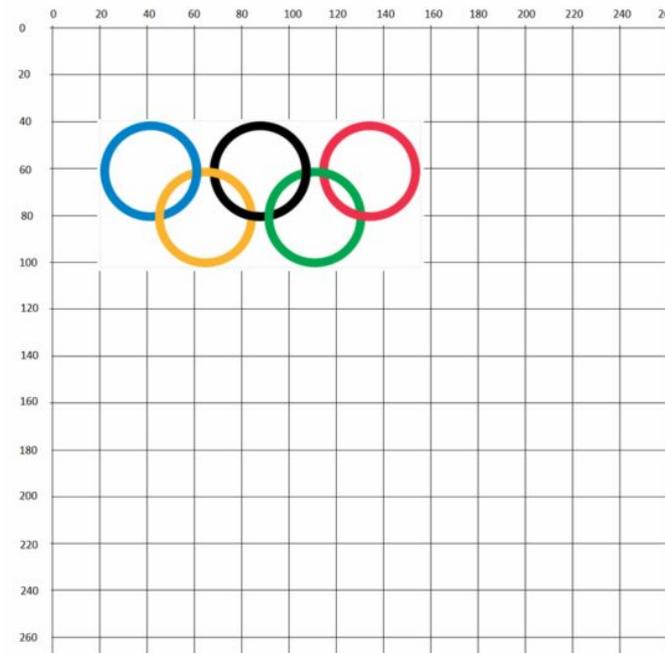
Нарисовать олимпийскую эмблему – настолько близко к оригиналу, насколько это возможно.

```
126 void OlympicEmblem(HDC hdc) {
127     // Кисть - прозрачная
128     SelectObject(hdc, GetStockObject(NULL_BRUSH));
129
130     // верхний ряд колец
131     HPEN hPen = CreatePen(PS_SOLID, 5, RGB(0, 0, 255));
132     SelectObject(hdc, hPen);
133     Ellipse(hdc, 20, 40, 60, 80);
134
135     hPen = CreatePen(PS_SOLID, 5, RGB(0, 0, 0));
136     SelectObject(hdc, hPen);
137     Ellipse(hdc, 70, 40, 110, 80);
138
139     hPen = CreatePen(PS_SOLID, 5, RGB(255, 0, 0));
140     SelectObject(hdc, hPen);
141     Ellipse(hdc, 120, 40, 160, 80);
142
143     // нижний ряд колец
144     hPen = CreatePen(PS_SOLID, 5, RGB(255, 255, 0));
145     SelectObject(hdc, hPen);
146     Ellipse(hdc, 45, 60, 85, 100);
147
148     hPen = CreatePen(PS_SOLID, 5, RGB(0, 255, 0));
149     SelectObject(hdc, hPen);
150     Ellipse(hdc, 95, 60, 135, 100);
151 }
```



Задача 1 - код

```
void OlympicEmblem(HDC hdc) {  
    // Кисть - прозрачная  
    SelectObject(hdc, GetStockObject(NULL_BRUSH));  
  
    // верхний ряд колец  
    HPEN hPen = CreatePen(PS_SOLID, 5, RGB(0, 0, 255));  
    SelectObject(hdc, hPen);  
    Ellipse(hdc, 20, 40, 60, 80);  
  
    hPen = CreatePen(PS_SOLID, 5, RGB(0, 0, 0));  
    SelectObject(hdc, hPen);  
    Ellipse(hdc, 70, 40, 110, 80);  
  
    hPen = CreatePen(PS_SOLID, 5, RGB(255, 0, 0));  
    SelectObject(hdc, hPen);  
    Ellipse(hdc, 120, 40, 160, 80);  
  
    // нижний ряд колец  
    hPen = CreatePen(PS_SOLID, 5, RGB(255, 255, 0));  
    SelectObject(hdc, hPen);  
    Ellipse(hdc, 45, 60, 85, 100);  
  
    hPen = CreatePen(PS_SOLID, 5, RGB(0, 255, 0));  
    SelectObject(hdc, hPen);  
    Ellipse(hdc, 95, 60, 135, 100);  
}
```



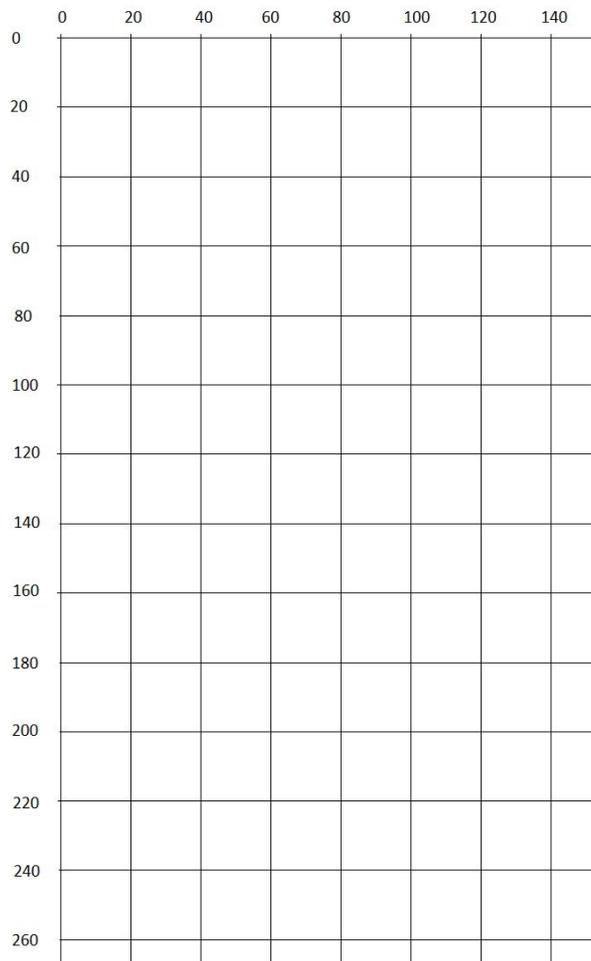
Задача 2

Нарисовать будильник. Отрисовку сделать в отдельной функции.



Задача 2

Нарисовать будильник. Отрисовку сделать в отдельной функции.



Задача 2

Нарисовать будильник. Отрисовку сделать в отдельной функции.

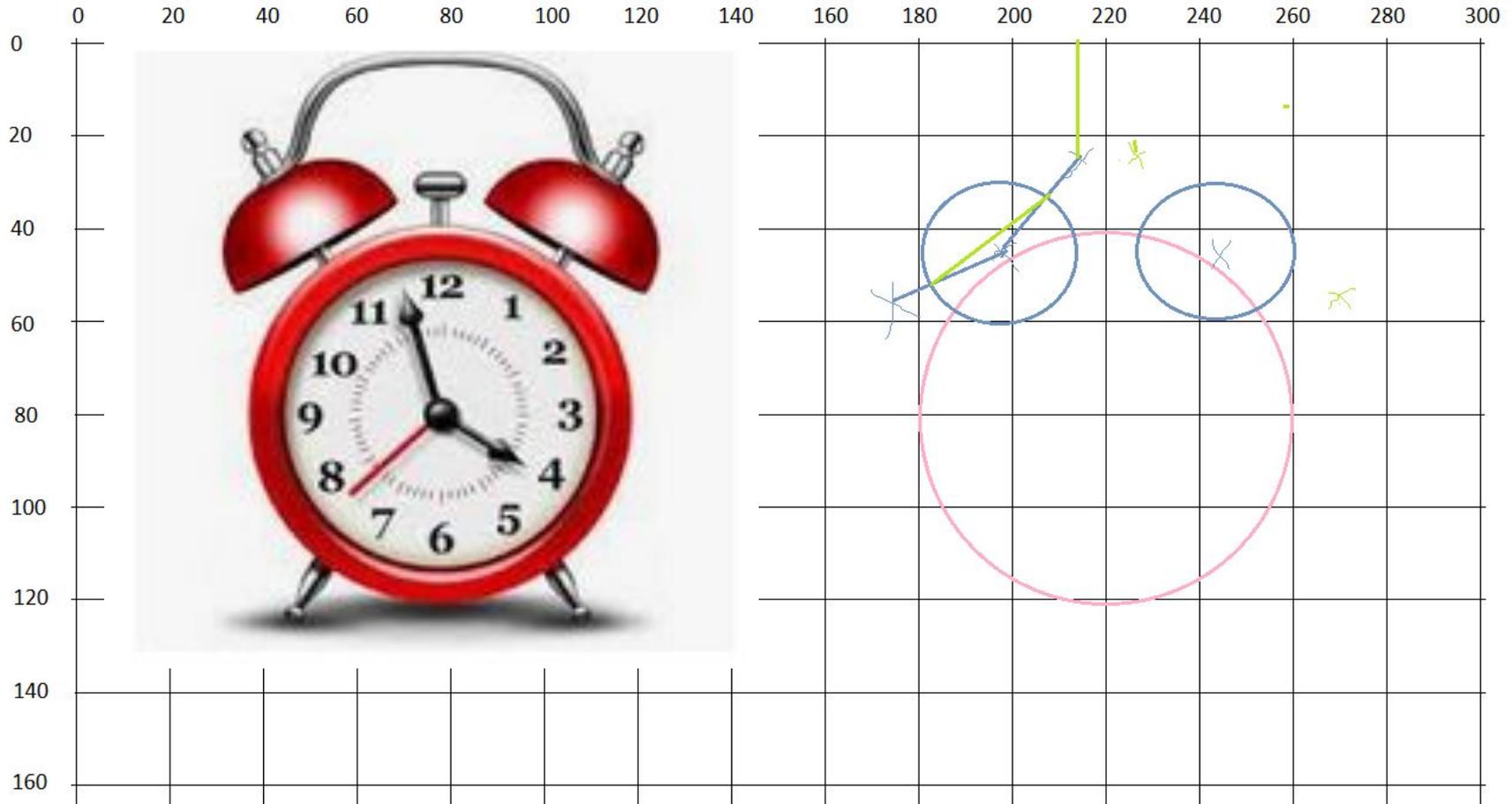
```
124 void AlarmClock(HDC hdc) {  
125     |  
126     }  
127
```

```
128 void OlympicEmblem(HDC hdc) {  
129     // Кисть - прозрачная  
130     SelectObject(hdc, GetStockObject(NULL_BRUSH));  
131
```

```
187     case WM_PAINT:  
188     {  
189         PAINTSTRUCT ps;  
190         HDC hdc = BeginPaint(hWnd, &ps);  
191         // TODO: Добавьте сюда любой код прорисовки, использующий HDC...  
192  
193         OlympicEmblem(hdc);  
194         AlarmClock(hdc);  
195  
196         EndPaint(hWnd, &ps);  
197     }  
198     break;
```

Задача 2

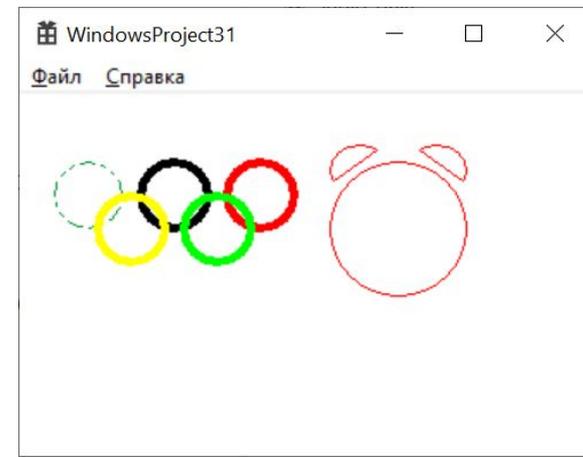
Нарисовать будильник. Отрисовку сделать в отдельной функции.



Задача 2

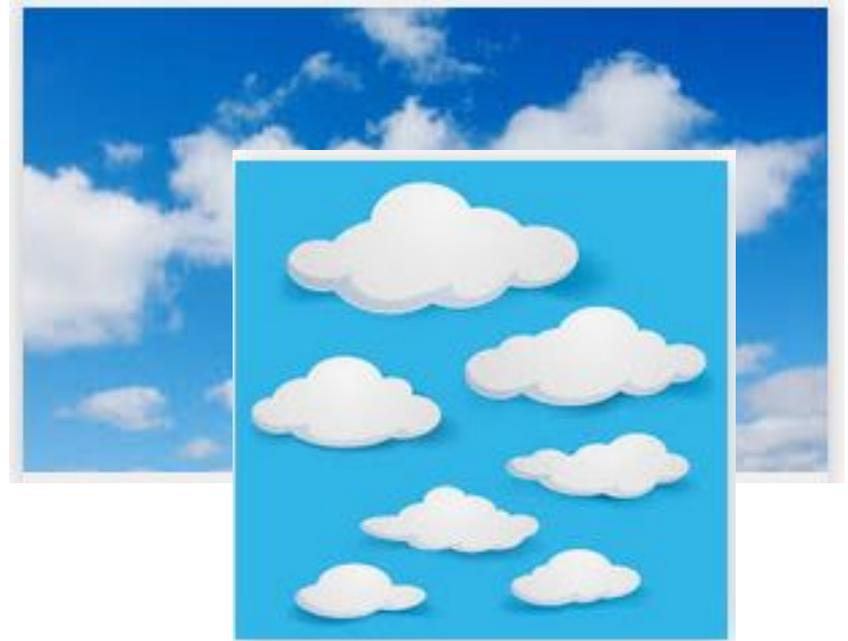
Нарисовать будильник. Отрисовку сделать в отдельной функции.

```
128 void AlarmClock(HDC hdc) {
129     HPEN hPen = CreatePen(PS_SOLID, 1, RGB(0xff, 0x0, 0x0));
130     SelectObject(hdc, hPen);
131
132     // Циферблат
133     Ellipse(hdc, 180, 40, 260, 120);
134
135     // левый колокольчик
136     //Ellipse(hdc, 180, 30, 215, 60); // dx = 35 dy = 30    rx = 17 ry = 15    cx = 197 cy =45
137     //nxStart = 215 nyStart = 25
138     //nxEnd = 175, nyEnd = 55
139
140     Chord(hdc, 180, 30, 215, 60, 215, 25, 175, 55);
141
142     // правый колокольчик
143     //Ellipse(hdc, 225, 30, 260, 60); // dx = 35 dy = 30    rx = 17 ry = 15    cx = 242 cy =45
144     //nxStart = 265 nyStart = 55
145     //nxEnd = 225, nyEnd = 25
146
147     Chord(hdc, 225, 30, 260, 60, 265, 55, 225, 25);
148 }
```



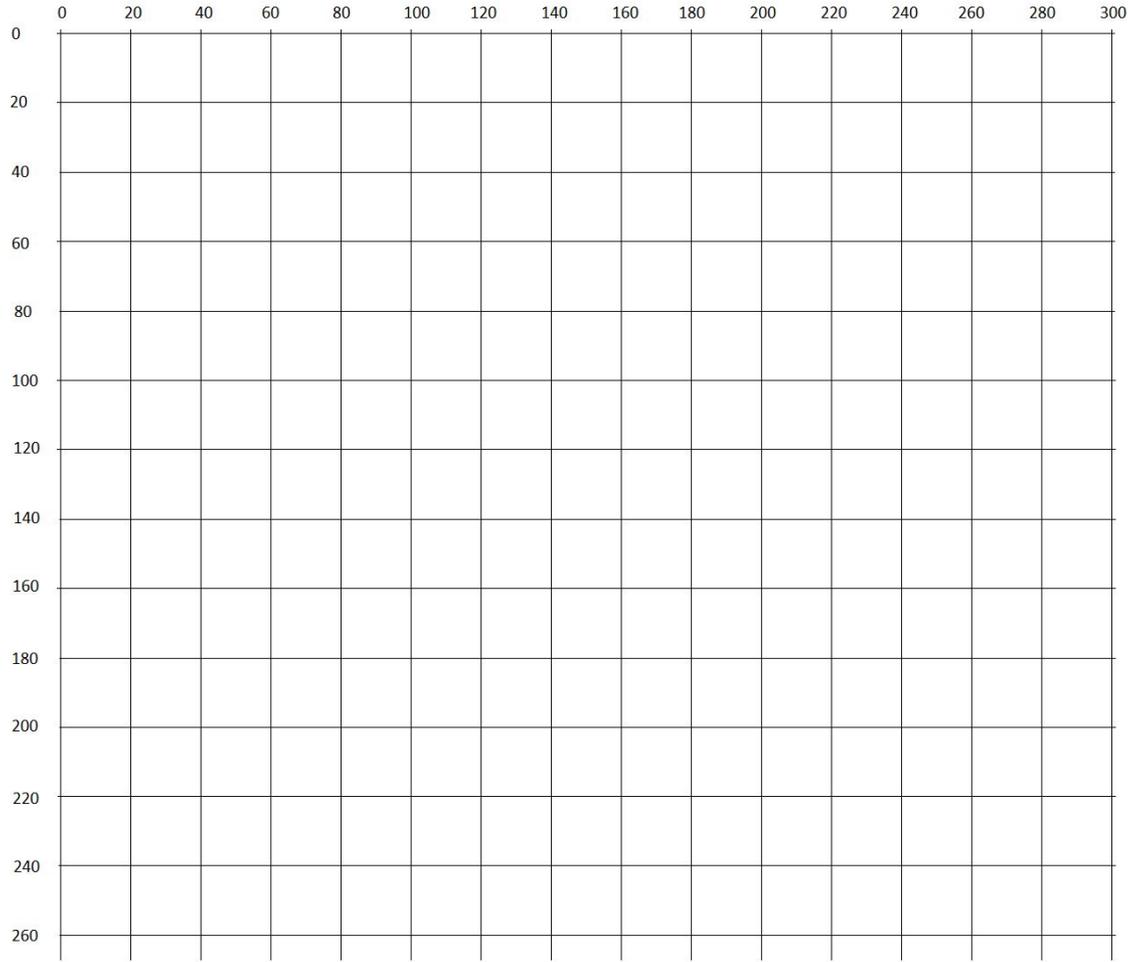
Задача 3

Нарисовать облако. Отрисовку сделать в отдельной функции.



Задача 3

Нарисовать облако. Отрисовку сделать в отдельной функции.



Задача 3

Нарисовать облако. Отрисовку сделать в отдельной функции.

```
124 void Cloud(HDC hdc) {  
125     |  
126     |}  
127  
128 void AlarmClock(HDC hdc) {
```

```
191 case WM_PAINT:  
192 {  
193     PAINTSTRUCT ps;  
194     HDC hdc = BeginPaint(hWnd, &ps);  
195     // TODO: Добавьте сюда любой код прорисовки, использующий HDC...  
196  
197     OlympicEmblem(hdc);  
198     AlarmClock(hdc);  
199     Cloud(hdc);  
200  
201     EndPaint(hWnd, &ps);  
202 }  
203 break;
```

Задача 4*

Нарисовать ????. Отрисовку сделать в отдельной функции.



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

- **Программирование WIN32 API | Функции WIN32 API**

https://life-prog.ru/view_komputernaya_grafika.php?id=10

- **КАК рисовать в Win32 API?**

[-http://radiofront.narod.ru/htm/prog/htm/winda/api/paint.html](http://radiofront.narod.ru/htm/prog/htm/winda/api/paint.html)

Домашнее задание

1. Дорисовать все картинки из презентации, которые не успели нарисовать в классе.
2. Создать свой собственный рисунок, содержащий эллипсы, сегменты и секторы эллипса. Реализовать его в виде отдельной функции.
3. *** Переделать все ранее сделанные рисунки в виде «Один предмет = одна отдельная функция»

Домашнее задание – оформление

Для сдачи домашней работы нужно иметь:

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

Срок выполнения – до следующей встречи на лабораторной работе

Если болел/не мог – это становится «долгом»