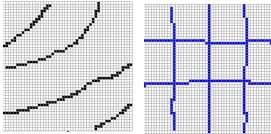


Lab #10

Изобразяване на графична информация, GDI (Graphical Device Interface), стандартни съобщения за прозорец. Обновяване на работната област на прозорец. Извеждане на векторна графика, запълване на области. Примерна графика за бизнес графика.

Solution/Project: нов

1. Създаваме нов проект LabPaintStatic от тип C++/Windows/Desktop/Application (VS2019)
2. Създаваме две битмап изображения с ID IDB_BITMAP1 и IDB_BITMAP2 подобно на:



3. Стандартната обработка на съобщението за „рисуване“ в обработващата функция на приложението е:

```
case WM_PAINT:
{
    PAINTSTRUCT ps;
    HDC hdc = BeginPaint(hWnd, &ps);
    // TODO: Add any drawing code that uses hdc here...
    EndPaint(hWnd, &ps);
}
break;
```

Функцията започва с BeginPaint и завършва с EndPaint, като в случая приключва без действия.

Всички графични действия се извършват между тях!!!

Раздел 1. Извеждане на текст

4. Извеждане на текст се извършва
 - a. В линия, границите на която се определят от правоъгълник – DrawText
 - b. На фиксирана позиция в прозореца на приложението – TextOut
5. Цветове:
 - a. Цвят на текста – SetTextColor
 - b. Цвят на фона – SetBkColor
6. Извеждане в линия

```
TCHAR szHello[] = L"Hello World!"; // текстът, който ще извеждаме

RECT rt, rt1, rt2;
// вземаме в rt текущите граници на прозореца на приложението.
GetClientRect(hWnd, &rt);

// Линия 1 - първа в прозореца с височина 20
rt1 = rt;
rt1.bottom = 20;

// цветовете са по подразбиране
DrawText(hdc, szHello, ARRAYSIZE(szHello), &rt1, DT_CENTER);

// Линия 2 - следваща в прозореца с височина 60
rt2 = rt1;
rt2.top = rt1.bottom + 1;
rt2.bottom = rt2.top + 60;
```

```

        if (rt2.bottom > rt.bottom && rt.bottom > rt2.top)
            rt2.bottom = rt.bottom;

        SetTextColor(hdc, RGB(0xFF, 0x00, 0x00)); // red
        DrawText(hdc, szHello, ARRAYSIZE(szHello), &rt2, NULL);
        SetTextColor(hdc, RGB(0x00, 0xFF, 0x00)); // green
        DrawText(hdc, szHello, ARRAYSIZE(szHello), &rt2, DT_SINGLELINE | DT_RIGHT |
DT_BOTTOM);

```



За да видите ефекта от параметрите за извеждане (4-ти параметър) като `DT_CENTER`, `DT_BOTTOM`, намалете размера на екрана по хоризонтал и вертикал:



Ще визуализираме границите на двете линии като добавим **след съответните редове** функция `Rectangle`

```

        rt1.bottom = 20;
        Rectangle(hdc, rt1.left, rt1.top, rt1.right, rt1.bottom);
и
        if (rt2.bottom > rt.bottom && rt.bottom > rt2.top)
            rt2.bottom = rt.bottom;
        Rectangle(hdc, rt2.left, rt2.top, rt2.right, rt2.bottom);

```

7. Извеждане на фиксирана позиция

След

```
DrawText(hdc, szHello, ARRAYSIZE(szHello), &rt2, DT_SINGLELINE | DT_RIGHT | DT_BOTTOM);
```

Добавяме:

```

// Извеждане на фиксирана позиция
TCHAR szIcan[] = L" I can paint ";
SetBkColor(hdc, RGB(0xFF, 0x66, 0xFF)); // pink
TextOut(hdc, 150, 100, szIcan, ARRAYSIZE(szIcan));
SetBkColor(hdc, RGB(0xFF, 0xFF, 0xFF)); // restore white

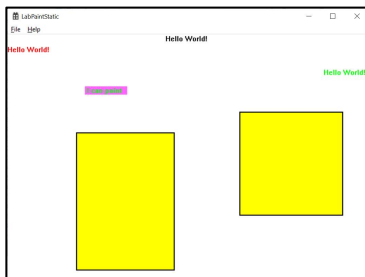
```



Раздел 2. Извеждане на фигури

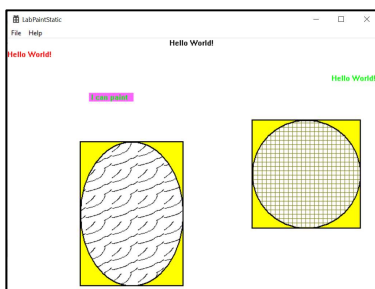
8. При извеждане на фигури се използват следните функции:
 - a. Контур на фигурата: CreatePen
 - b. Запълване на фигурата: CreateSolidBrush, CreatePatternBrush, CreateHatchBrush
 - c. Активиране на всеки елемент: SelectObject
 - d. Всеки активиран елемент е валиден до активиране на друг от същия тип
9. Извеждане на правоъгълник

```
// Установяване на контур
HPEN hPen = CreatePen(PS_SOLID, 2, RGB(0, 0, 0)); // black
SelectObject(hdc, hPen);
// Установяване на запълване
HBRUSH hBrush1 = CreateSolidBrush(RGB(0xFF, 0xFF, 0x00)); // yellow
SelectObject(hdc, hBrush1);
// Painting 2 rectangles
Rectangle(hdc, 134, 456, 324, 190);
Rectangle(hdc, 450, 150, 650, 350);
```



10. Извеждане на кръг/елипса

```
// Запълване на елипса с битмап
HBITMAP hBmp = LoadBitmap(hInst, MAKEINTRESOURCE(IDB_BITMAP1));
HBRUSH hBrush2 = CreatePatternBrush(hBmp);
SelectObject(hdc, hBrush2);
Ellipse(hdc, 134, 456, 324, 190);
//елипса
HBRUSH hBrush3 = CreateHatchBrush(HS_CROSS, RGB(0x88, 0x88, 0x20));
SelectObject(hdc, hBrush3);
Ellipse(hdc, 450, 150, 650, 350);
```



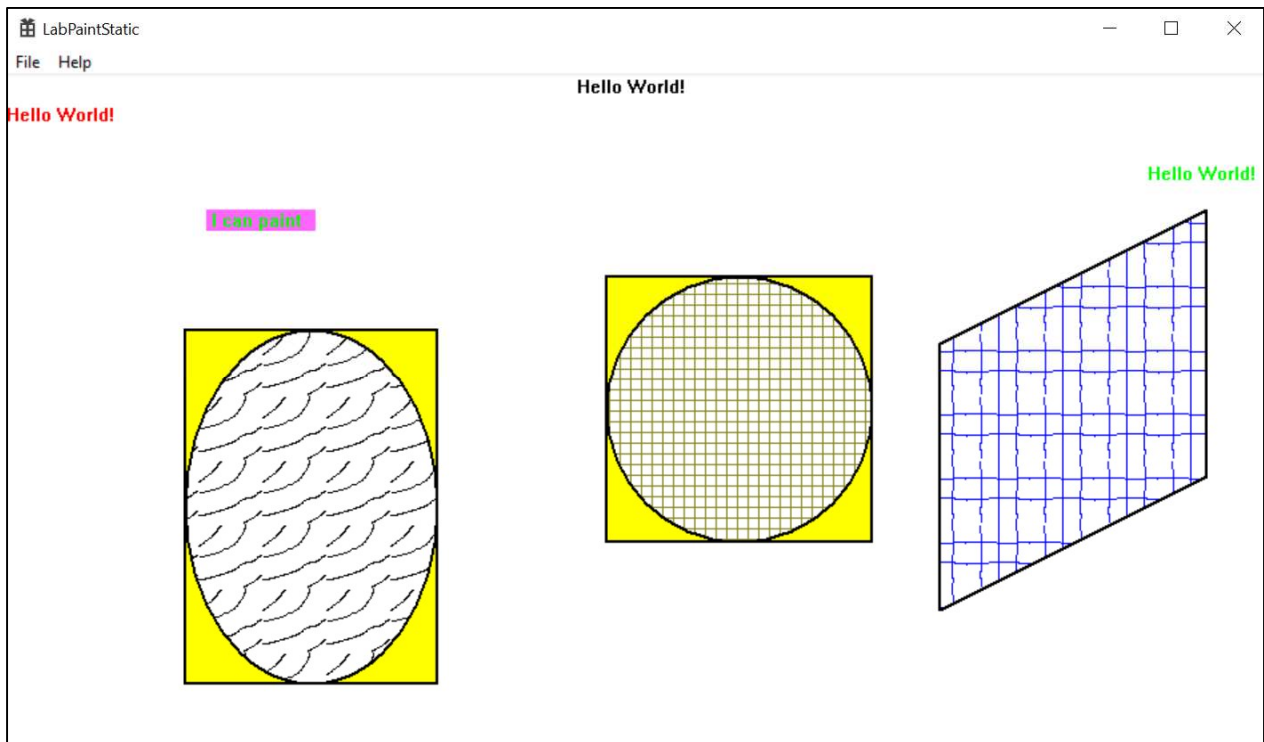
11. Извеждане на многоъгълник

```
//полигон
POINT pp[4];
HBITMAP hBmp1 = LoadBitmap(hInst, MAKEINTRESOURCE(IDB_BITMAP2));
HBRUSH hBrush14 = CreatePatternBrush(hBmp1);
SelectObject(hdc, hBrush14);
pp[0].x = 700; pp[0].y = 200;
```

```

pp[1].x = 900; pp[1].y = 100;
pp[2].x = 900; pp[2].y = 300;
pp[3].x = 700; pp[3].y = 400;
Polygon(hdc, pp, 4);

```



12. Цялостен код на функцията WndProc за лабораторното упражнение

```

LRESULT CALLBACK WndProc(HWND hWnd, UINT message, WPARAM wParam, LPARAM lParam)
{
    switch (message)
    {
        case WM_COMMAND:
        {
            int wmId = LOWORD(wParam);
            // Parse the menu selections:
            switch (wmId)
            {
                case IDM_ABOUT:
                    DialogBox(hInst, MAKEINTRESOURCE(IDD_ABOUTBOX), hWnd, About);
                    break;
                case IDM_EXIT:
                    DestroyWindow(hWnd);
                    break;
                default:
                    return DefWindowProc(hWnd, message, wParam, lParam);
            }
        }
        break;
        case WM_PAINT:
        {
            PAINTSTRUCT ps;
            HDC hdc = BeginPaint(hWnd, &ps);
            // TODO: Add any drawing code that uses hdc here...

            TCHAR szHello[] = L"Hello World!"; // текстът, който ще извеждаме

            RECT rt, rt1, rt2;

```

```

// вземаме в rt текущите граници на прозореца на приложението.
GetClientRect(hWnd, &rt);

// Линия 1 - първа в прозореца с височина 20
rt1 = rt;
rt1.bottom = 20;
//Rectangle(hdc, rt1.left, rt1.top, rt1.right, rt1.bottom);

// цветовете са по подразбиране
DrawText(hdc, szHello, ARRAYSIZE(szHello), &rt1, DT_CENTER);

// Линия 2 - следваща в прозореца с височина 60
rt2 = rt1;
rt2.top = rt1.bottom + 1;
rt2.bottom = rt2.top + 60;
if (rt2.bottom > rt.bottom && rt.bottom > rt2.top)
    rt2.bottom = rt.bottom;
//Rectangle(hdc, rt2.left, rt2.top, rt2.right, rt2.bottom);

SetTextColor(hdc, RGB(0xFF, 0x00, 0x00)); // red
DrawText(hdc, szHello, ARRAYSIZE(szHello), &rt2, NULL);
SetTextColor(hdc, RGB(0x00, 0xFF, 0x00)); // green
DrawText(hdc, szHello, ARRAYSIZE(szHello), &rt2, DT_SINGLELINE | DT_RIGHT |
DT_BOTTOM);

// Извеждане на фиксирана позиция
TCHAR szIcan[] = L" I can paint ";
SetBkColor(hdc, RGB(0xFF, 0x66, 0xFF)); // pink
TextOut(hdc, 150, 100, szIcan, ARRAYSIZE(szIcan));
SetBkColor(hdc, RGB(0xFF, 0xFF, 0xFF)); // restore white

// Установяване на контур
HPEN hPen = CreatePen(PS_SOLID, 2, RGB(0, 0, 0)); // black
SelectObject(hdc, hPen);
// Установяване на запълване
HBRUSH hBrush1 = CreateSolidBrush(RGB(0xFF, 0xFF, 0x00)); // yellow
SelectObject(hdc, hBrush1);
// Painting 2 rectangles
Rectangle(hdc, 134, 456, 324, 190);
Rectangle(hdc, 450, 150, 650, 350);

// Запълване на елипса с битмап
HBITMAP hBmp = LoadBitmap(hInst, MAKEINTRESOURCE(IDB_BITMAP1));
HBRUSH hBrush2 = CreatePatternBrush(hBmp);
SelectObject(hdc, hBrush2);
Ellipse(hdc, 134, 456, 324, 190);
//елипса
HBRUSH hBrush3 = CreateHatchBrush(HS_CROSS, RGB(0x88, 0x88, 0x20));
SelectObject(hdc, hBrush3);
Ellipse(hdc, 450, 150, 650, 350);

//полигон
POINT pp[4];
HBITMAP hBmp1 = LoadBitmap(hInst, MAKEINTRESOURCE(IDB_BITMAP2));
HBRUSH hBrush14 = CreatePatternBrush(hBmp1);
SelectObject(hdc, hBrush14);
pp[0].x = 700; pp[0].y = 200;
pp[1].x = 900; pp[1].y = 100;
pp[2].x = 900; pp[2].y = 300;
pp[3].x = 700; pp[3].y = 400;
Polygon(hdc, pp, 4);

EndPaint(hWnd, &ps);
}
break;
case WM_DESTROY:

```

```
        PostQuitMessage(0);
        break;
default:
    return DefWindowProc(hWnd, message, wParam, lParam);
}
return 0;
}
```