

### **Тақырып № 3**

#### **Графикалық ақпараттарды кескіндеу компоненттері**

C++Builder әр түрлі деңгейде Windows GDI функцияларын инкапсуляциялайды. Мұнда ең маңыздысы-графикалық компоненттер монитор экранында өз суреттерін ұсынатын әдіс. GDI функциясын тікелей шақырғанда, сіз таңдаған сурет құралдарын - қауырсын, қаріптер ц қылқаламдарын көрсететін құрылғы контекстінің дескрипторын (device context handle) жіберу керек. Графикалық суреттермен жұмыс істеуді аяқтағаннан кейін, құрылғының контекстін бастапқы қалпына келтіруге және содан кейін ғана одан босауға міндеттісіз.

Оның орнына, осы детализация деңгейінде графикамен жұмыс істеуге мәжбүр ету үшін, C++Builder сізге Canvas (Canvas) графикалық компонент қасиеттері арқылы қарапайым және аяқталған интерфейс ұсынады. Бұл сипат құрылғының дұрыс контекстін іске қосады және сурет салуды тоқтатқанда оны дұрыс уақытта босатады. Windows GDI канва функцияларына ұқсас, қаламның, қылқалам мен қаріптің сипаттарын білдіретін ішкі қасиеттері бар.

Графикалық компоненттермен жұмыс істей отырып, пайдаланушы жасайтын жалғыз нәрсе-қолданылатын сурет құралдарының сипаттамасын анықтау. Құралдарды жасау, таңдау және босату кезінде жүйелік ресурстарды бақылау қажет емес. Канва өзі бұл туралы ойлайды.

Графикамен жұмыс істеу кезінде C++Builder көрсететін артықшылықтардың бірі-жүйенің графикалық ресурстары үшін кэширленген жадты пайдалану. Егер сіздің бағдарлама қайта жасайды, пайдаланады және босатады, айталық, кейбір нақты түрдің қауырсыны, сіз осындай қауырсынды пайдаланған сайын осы қадамдарды қайталауға тура келеді. C++Builder графикалық ресурстарды сақтау үшін кэш-жадыны пайдаланатындықтан, жиі қолданылатын сурет құралы кэш-жадынан қайта таңдалып, әр жолы қайта қалпына келтірілмейді. Әлбетте, осының салдарынан сіздің графикалық

қосымшаңыздың қайталанатын операцияларының тиімділігі айтарлықтай артады.

1 листингі кодтың екі фрагменті бар, ол C++Builder графиканы бағдарламалау жеңілдетеді. Бірінші фрагмент GDI стандартты мүмкіндіктерін Windows үшін Owl қолданбасын салу үшін қолданады. Екінші фрагментте сол тапсырма сурет арқылы шешіледі.

```
void TMyWindow::Paint(TDC& PaintDC, bool erase, TRect& rect) {
```

```
HPEN PenHandle, OldPenHandle;
```

```
HBRUSH BrushHandle, OldBrushHandle;
```

```
PenHandle = CreatePen(PS_SOLID, 1, RGB(0, 0, 255));
```

```
OldPenHandle = SelectObject(PaintDC, PenHandle);
```

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

```
OldBrushHandle = SelectObject(PaintDC, BrushHandle);
```

```
Ellipse(10, 10, 50, 50);
```

```
SelectObject(OldBrushHandle) ;
```

```
DeleteObject(BrushHandle) ;
```

```
SelectObject(OldPenHandle);
```

```
DeleteObject(PenHandle) ; )
```

```
void_fastcall TFormI::FormPaint(TObject *Sender) {
```

```
Canvas->Pen->Color = clBlue; // выбрать цвет контура Canvas-  
>Brush->Color = clYellow; // выбрать цвет заливки Canvas->Ellipse(10, 20, 50,  
50); // нарисовать эллипс }
```

1. Windows GDI и функцияларын пайдалану эллипсті салу мысалында Канва қасиеттері.

Канва пайдалану

"Фигуралар мен мәтіннің жеке сызықтарын салу үшін жоғары деңгейдегі функциялардан бастап, әр түрлі деңгейдегі Windows графикалық функцияларын инкапсуляциялайды. Содан кейін сурет салу үшін арнамен манипуляциялар үшін орта деңгейдің қасиеттері мен әдістері бар. Соңында, төменгі деңгейде Windows GDI функцияларына қол жеткізу қамтамасыз етіледі.