

Тема: Стандартні графічні процедури.

Мета: Ознайомитися із стандартними графічними процедурами.

## 1. Клас TCanvas.

У Windows для позначення канви, на якій виконується малюнок, використовується термін контекст пристрою (DC). Взаємодія із контекстами пристроїв на рівні виклику функцій API може виявитися досить складним. Тому спочатку потрібно тримати в системі Windows дескриптор контексту пристрою (посилання на формальний опис можливостей пристрою), після чого вибрати в нього потрібні нам об'єкти (пера, кисті, шрифти тощо). Тільки після цього можна на ньому щось малювати. По завершенні малювання, перед тим, як вилучити контекст пристрою, потрібно вивільнити вибрані в нього об'єкти. У протилежному випадку пам'ять, яка використовується у додатку, ніколи не буде повернута операційній системі.

Існуючий клас TCanvas із бібліотеки візуальних компонентів значно полегшує роботу із контекстами пристроїв. В ролі властивостей (property) він входить до багатьох компонент.

Клас TCanvas має багато властивостей та методів.

### 1.1. Основні властивості класу TCanvas:

Властивість	Опис
Brush	Задає колір заливки та шаблон при зафарбуванні фігур
ClipRect	Являє собою поточний прямокутний регіон відсікання канви. Всі операції графічного виведення обмежуються заданим прямокутником. Властивість лише для читання
CopyMode	Визначає, яким чином буде відбуватися копіювання (метод CopyRect) на дану канву зображення із другого місця: один до одного, з інверсією зображення тощо
Font	Містить шрифт, який клас TCanvas використовує при виведенні тексту (метод TextOut)
Handle	Дескриптор (HDC) канви. Дана властивість дає можливість напряду викликати функції API Windows
Pen	Визначає стиль та колір ліній
PenPos	Задає координати x та y для чергової операції малювання
Pixels	Являє собою двовимірний масив пікселів об'єкту Canvas

### 1.2. Основні методи класу TCanvas:

Властивість	Опис
Arc	Малює на канві поточним пером сегмент еліпсу
BrushCopy	Малює бітову матрицю із прозорим фоном
CopyRect	Копіює на канву частину зображення
Draw	Копіює зображення із пам'яті на канву
Ellipse	Малює поточним пером еліпс та зафарбовує його поточною кистю
FloodFill	Виконує заливку області канви поточною кистю

Властивість	Опис
LineTo	Проводить поточним пером лінію із даної точки в точку із координатами x та y
MoveTo	Встановлює поточне положення пера
Pie	Малює на канві сектор еліпсу
Polygon	За допомогою масиву об'єктів типу TPoint будує багатокутник та зафарбовує його поточною кистю
Polyline	За допомогою масиву об'єктів типу TPoint будує поточним пером ламану лінію. Автоматичного замикання контуру при цьому не відбувається
Rectangle	Малює на канві поточним пером прямокутник та зафарбовує його поточною кистю
RoundRect	Малює зафарбований прямокутник із закругленими кутами
StretchDraw	Копіює бітову матрицю із пам'яті на канву. При цьому у відповідності із цільовою прямокутною областю відбувається масштабування (розтягування або стиснення) растрових зображень
TextExtend	Повертає висоту та ширину рядка Text у пікселях. При розрахунках враховується поточний шрифт канви
TextHeight	Повертає висоту рядка Text у пікселях. При розрахунках враховується поточний шрифт канви
TextOut	Виводить рядок Text за допомогою поточного шрифту в заданому місці канви
TextRect	Виводить текст із відсіканням. При цьому частина тексту, яка потрапила за межі прямокутного регіону відсікання, стає невидимою

Перераховані властивості та методи являють собою лише незначну частину можливостей контекстів пристроїв Windows. Але навіть ця частина може задовольнити 80% потреб, які виникають при роботі із графікою. Проте, перед тим, як перейти до розгляду класу TCanvas, зупинимося на графічних об'єктах, які застосовуються у програмуванні для Windows.

В інтерфейсі графічних пристроїв (GDI) Windows наявна велика кількість типів об'єктів, які мають певний вплив на роботу контекстів графічних пристроїв. Найбільш часто застосовуваними є об'єкти – пера, кисті, шрифти. Менш популярними є об'єкти – палітри, бітові матриці, регіони відсікання.

### 1.3. Клас TPen.

Розглянемо можливості вкладеного об'єкту перо (pen).

За допомогою пера можна малювати різні лінії – як окремі, які з'єднують одну точку з іншою, так і лінії, що обмежують різні геометричні фігури – прямокутник, еліпс, багатокутник.

Для доступу до пера використовують властивість Pen класу TCanvas, яка являє собою об'єкт типу TPen.

Властивості класу TPen:

Властивість	Опис
Color	Задає колір лінії

Властивість	Опис
Handle	Являє собою дескриптор пера (HPEN). Використовується у випадку прямих викликів відповідною функцією API
Mode	Визначає спосіб малювання лінії: нормальний, інверсний, із винятковим або (xor) тощо
Style	Визначає стиль лінії: суцільний, пунктирний, штриховий, штрих-пунктирний, прозорий тощо
Width	Задає товщину лінії у пікселях

У більшості випадків використання цих властивостей не викликає труднощів.

Зауваження. Якщо нам необхідно сховати вікно, на якому виведено малюнок та текст, а потім його знову активізувати, то виявиться, що малюнок на формі зник. Це відбулося тому, що малюнок поки що носить тимчасовий характер. Щоб зробити його постійним, потрібно помістити програмний код в обробник події OnPaint форми. Тоді при необхідності перемалювати вікно буде генеруватися подія OnPaint – малюнок відновиться.

Штриховий та пунктирний стилі можна використовувати тільки при роботі із пером товщиною 1 піксел. За допомогою стилю psClear можна вилучати лінії, які малюються на зовнішній границі таких об'єктів, як, наприклад, прямокутники, еліпси, та зафарбовані багатокутники.

#### 1.4. Клас TBrush.

Кисті визначають спосіб зафарбування фігур. За допомогою активної кисті заповнюється внутрішня область кожної із намальованих фігур – еліпсів, прямокутників, багатокутників тощо. Кисть може намалювати штриховку або бітову матрицю. Зовнішній вигляд кисті можна контролювати за допомогою властивості Brush класу TCanvas, який, у свою чергу, є об'єктом класу TBrush. Як і TPen, TBrush не містить методів та подій для програмування.

Властивості класу TBrush:

Властивість	Опис
Bitmap	Ідентифікує бітову матрицю, яка використовується у ролі фону кисті
Color	Задає колір кисті
Handle	Являє собою дескриптор кисті (HBRUSH). Використовується у випадку прямих викликів відповідною функцією API
Style	Визначає стиль кисті: суцільний, прозорий, одна із можливих штриховок

Властивість Style за замовчуванням має значення bsSolid (суцільна заливка). Якщо при зафарбовуванні геометричних фігур Ви маєте бажання використати штриховку, то властивості Style потрібна присвоїти один із наступних стилів: bsHorizontal, bsVertical, bsFDiagonal, bsBDiagonal, bsCross, bsDiagCross.

Коли Ви використовуєте кисть – штриховку, її властивість Color визначає колір лінії штриховки. При зафарбовуванні графічних об'єктів за допомогою такої кисті VCL автоматично встановлює режим фону в Transparent (наприклад, прозорий). Це значить, що колір фону кисті буде співпадати із кольором вікна, на якому намальований графічний об'єкт.