

## Лабораторна робота №4

**Тема:** Ієрархічна система побудови графічної сцени

**Мета:** Створити формат збереження векторного зображення. Додати рухомі елементи.

### Теоретичні відомості

Перший об'єкт нехай рухається поступово вздовж осі OX, а другий — рухається навколо першого по колу. Запишемо перетворення координат для руху вздовж відрізка:

$$\begin{cases} x := x_0 + (x_1 - x_0) \cdot t, \\ y := y_0 + (y_1 - y_0) \cdot t, \end{cases} \text{ де } 0 \leq t \leq 1.$$

Відповідно, для руху по колу теж можна використати параметричний запис:

$$\begin{cases} x := x_0 + r_x \cos(2\pi \cdot t), \\ y := y_0 - r_y \sin(2\pi \cdot t), \end{cases} \text{ де } 0 \leq t \leq 1.$$

Але для того щоб другий об'єкт пересувався навколо першого, потрібно сумістити пересування по прямій та по колу:

$$\begin{cases} x := x_0 + (x_1 - x_0) \cdot t + r_x \cos(2\pi n t), \\ y := y_0 - (y_1 - y_0) \cdot t - r_y \sin(2\pi n t), \end{cases},$$

де  $0 \leq t \leq 1$ ,  $n$  — кількість обертів на пересування вздовж відрізка.

$$\begin{cases} x_1 := dx_0 + x_0 \cos(\alpha) + y_0 \sin(\alpha) \\ y_1 := dy_0 - x_0 \sin(\alpha) + y_0 \cos(\alpha) \end{cases}, \quad \begin{cases} x_2 := dx_1 + x_1 \cos(\alpha) + y_1 \sin(\alpha) \\ y_2 := dy_1 - x_1 \sin(\alpha) + y_1 \cos(\alpha) \end{cases}.$$

Це можна досягти зміною класу векторних об'єктів додаванням списку залежних від нього об'єктів:

```
TMyImage = class
public
    Transform: TTransformer; //Відносне перетворення
    activeTransform: TTransformer; //Активне перетворення
    Child: TList; //Список залежних елементів
    Points: array of TPoint; //Список координат точок
    PolyLine: array of Integer; //Список індексів точок
    constructor Create;
    destructor Destroy; override;
    procedure LoadPicFromFile(FileName: String);
    procedure DrawMyImage(Canvas: TCanvas); //Малювати на вказаній канві
```

```
procedure AddChild(newImage:TMyImage); //Додати залежний об'єкт  
end;
```

Та зміною класу перетворення координат:

```
procedure Ttransformer.AddTransform(nextTransform:TTransformer);
```

### Хід роботи

1. Створіть та збережіть новий проект в середовищі Lazarus.
2. Додайте на головне вікно компонент TImage та TTimer.
3. Подвійним кліком на TTimer додайте процедуру Form1.onTimer.
4. Створіть довільний малюнок. Малюнок повинен бути унікальним для кожного виконавця роботи. Малюнок повинен складатися з кількох ієрархічно пов'язаних частин з власною формою руху відносно основної частини (колеса, що обертаються та підстрибують, качаються руки в рухомого чоловічка, тощо). Збережіть компоненти малюнка у окремі файли у вигляді списків точок та ламаних ліній.
5. Змініть текст програми з минулих робіт для підтримки послідовного перетворення координат з метою організації відносності положення елементів.
6. За допомогою додаткового лічильника кадрів організуйте рух вашого елементу малюнка.
7. Створіть композицію з намальованих елементів, щоб малюнок виглядав цілісним.
8. Результат роздрукуйте та додайте до звіту разом з текстом програми.
9. Дайте відповіді на контрольні питання.
10. В разі виконання роботи на поточній парі дозволяється використання електронного звіту з усними відповідями на контрольні питання.
11. Зробіть висновки що до досяжності мети поставленої в лабораторній роботі.

### Контрольні питання:

1. Яка послідовність перетворення координат потрібна якщо навколо об'єкту, що обертається, обертається інший об'єкт?
2. Чи можна за розробленою схемою використати один файл для збереження багатокomпонентного малюнка?
3. Як змінити програму для малювання фарбованих зображень?
4. Як знайти внутрішню точку для опуклого багатокутника?
5. Чи буде корисним використання зміни послідовності застосувань перетворень координат для малювання?