

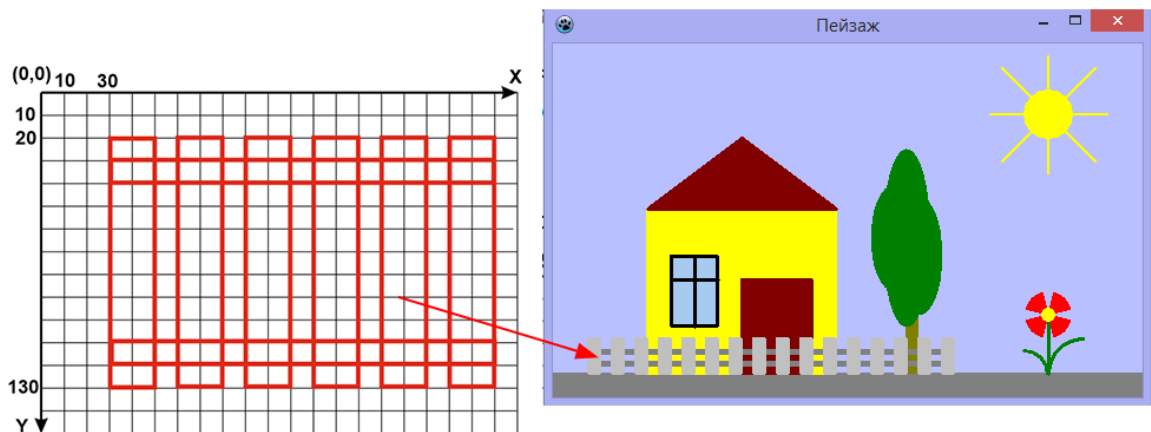
Урок 41. Цикли з лічильником.

Вивчення нового матеріалу.

Слайд №1

Сьогодні ми розглянемо оператори циклів на прикладі створення зображень.

Спочатку побудуємо паркан, який згодом можна буде додати до нашого пейзажу, вказавши необхідні координати.



Слайд №2

Код вийшов доволі громіздким. Однак зауважимо, що наше зображення містить 5 повторюваних елементів

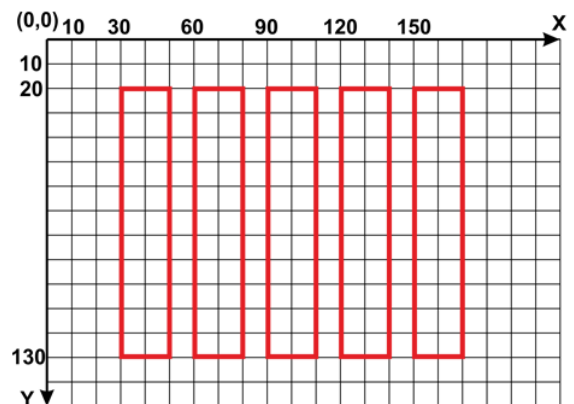
```
Canvas.Rectangle( 30 , 20 , 50 , 130 );
```

```
Canvas.Rectangle( 60 , 20 , 80 , 130 );
```

```
Canvas.Rectangle( 90 , 20 , 110 , 130 );
```

```
Canvas.Rectangle( 120 , 20 , 140 , 130 );
```

```
Canvas.Rectangle( 150 , 20 , 170 , 130 );
```



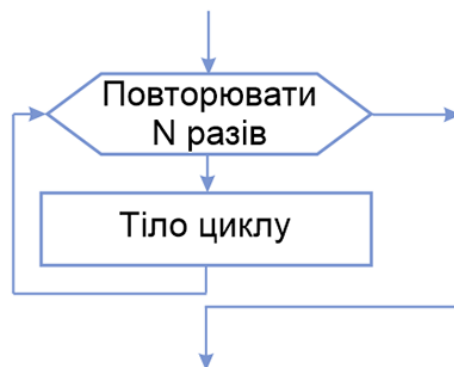
Тому код можна скоротити, скориставшись **циклом з лічильником**.

Програмний код **циклу з лічильником** має вигляд

For лічильник **:=** початкове значення **To** кінцеве значення **Do**
оператори тіла циклу

де **лічильник** – змінна цілого типу

Даному виду циклів відповідає така блок-схема



Приклад

For $i := 1$ **To** 3 **Do**
 $S := S * 2$;

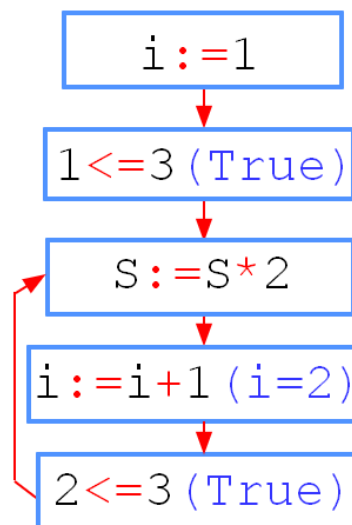
Алгоритм виконання програмного коду

1. Лічильнику присвоюємо початкове значення.

2. Якщо початкове значення менше або дорівнює кінцевому, переходимо до операторів тіла циклу, інакше завершуємо цикл

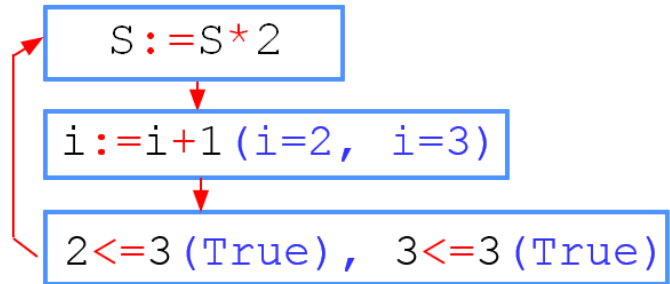
3. Виконуємо тіло циклу

4. Значення лічильника збільшуємо на 1



Слайд №5

Цикл буде виконуватися доти, доки значення лічильника міститься в діапазоні між початковим і кінцевим значенням.



Коли значення лічильника стає більшим за кінцеве значення, виконання тіла циклу закінчується і здійснюється перехід до наступних команд.

4 <= 3 (False)



Якщо початкове значення циклу є більшим за кінцеве, то тіло циклу жодного разу не виконається.

Слайд №6

Одне виконання тіла циклу називають **ітерацією**

Так, для наступного фрагменту програми буде виконано три ітерації.

```

S := 1;
For i := 1 To 3 Do
  S := S * 2;
  
```

№ ітерації	i	S
1	1	1 * 2 = 2
2	2	2 * 2 = 4
3	3	4 * 2 = 8

Слайд №7

Як видно з наведених кодів, саме на **30** відрізняються координати X

```

Canvas.Rectangle(30, 20, 50, 130);
Canvas.Rectangle(60, 20, 80, 130);
Canvas.Rectangle(90, 20, 110, 130);
Canvas.Rectangle(120, 20, 140, 130);
Canvas.Rectangle(150, 20, 170, 130);
  
```

Для визначення координати X було використано формулу

$$i * 30$$

Порядковий номер
зображення

Відстань між
зображеннями

Отже, для "розмноження" будь-яких зображень можна застосувати формулу

$$i * d$$

де i – порядковий номер зображення

d – відстань між зображеннями

Вправа 1.

Вправа №1

Вправа 1 у Lazarus

1. Відкрийте проект з попереднього уроку.
2. Уведіть код для побудови 5 дощечок паркану з використанням вказівки повторення.
3. Запустіть проект та переконайтеся у правильності побудови.

Підказка до вправи 1.

Підказка до вправи 1

Підказка до вправи 1

На початку коду оголосіть змінну i цілого типу.

Запрограмуйте цикл із заголовком

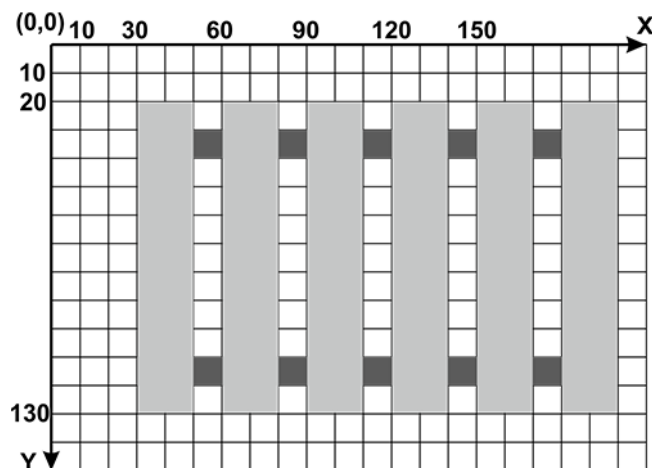
For $i:=1$ To 5 Do

Вправа 2.

Вправа №2

Вправа 2 у Lazarus

1. Змініть код для побудови на формі **15** дощечок.
2. Додайте коди для побудови горизонтальних дощечок.
3. Додайте команди для встановлення кольору паркану.
4. Запустіть проект та переконайтеся у правильності побудови.



Вправа 3.

Вправа №3

Вправа 3 у Lazarus

1. Відкрийте проект з пейзажем.
2. Змініть коди розташування дверей і вікна згідно наведених на малюнку.
3. Додайте коди створення горища та драбини.
4. На власний розсуд встановіть кольори горища та драбини.

