

# Тема: Цикл з післяумовою. Розв'язування задач.

## I. Повторення вивченого.

Завдання «Знайти пару»

Необхідно розмістити під об'єкти цифри з назвами так, щоб не порушувалась відповідність.

| <pre> <b>If</b> &lt;логічний вираз&gt; <b>Then begin</b>     &lt;послідовність команд 1&gt; <b>end</b> <b>Else begin</b>     &lt;послідовність команд 2&gt; <b>end;</b>                 </pre> |  | <pre> <b>If</b> &lt;логічний вираз&gt; <b>Then begin</b>     &lt;послідовність команд&gt; <b>end;</b>                 </pre> |              |      |       |       |      |   |          |          |                |      |      |      |      |       |       |       |      |       |       |       |       |
|--|--|--|--------------|------|-------|-------|------|---|----------|----------|----------------|------|------|------|------|-------|-------|-------|------|-------|-------|-------|-------|
|  |  |  |              |      |       |       |      |   |          |          |                |      |      |      |      |       |       |       |      |       |       |       |       |
|  | <table border="1"> <thead> <tr> <th><i>x</i></th> <th><i>not x</i></th> </tr> </thead> <tbody> <tr> <td>true</td> <td>false</td> </tr> <tr> <td>false</td> <td>true</td> </tr> </tbody> </table> | <i>x</i>   | <i>not x</i> | true | false | false | true | <table border="1"> <thead> <tr> <th><i>x</i></th> <th><i>y</i></th> <th><i>x and y</i></th> </tr> </thead> <tbody> <tr> <td>true</td> <td>true</td> <td>true</td> </tr> <tr> <td>true</td> <td>false</td> <td>false</td> </tr> <tr> <td>false</td> <td>true</td> <td>false</td> </tr> <tr> <td>false</td> <td>false</td> <td>false</td> </tr> </tbody> </table> | <i>x</i> | <i>y</i> | <i>x and y</i> | true | true | true | true | false | false | false | true | false | false | false | false |
| <i>x</i>   | <i>not x</i>   |  |              |      |       |       |      |   |          |          |                |      |      |      |      |       |       |       |      |       |       |       |       |
| true   | false  |  |              |      |       |       |      |   |          |          |                |      |      |      |      |       |       |       |      |       |       |       |       |
| false  | true   |  |              |      |       |       |      |   |          |          |                |      |      |      |      |       |       |       |      |       |       |       |       |
| <i>x</i>   | <i>y</i>   | <i>x and y</i>   |              |      |       |       |      |   |          |          |                |      |      |      |      |       |       |       |      |       |       |       |       |
| true   | true   | true   |              |      |       |       |      |   |          |          |                |      |      |      |      |       |       |       |      |       |       |       |       |
| true   | false  | false  |              |      |       |       |      |   |          |          |                |      |      |      |      |       |       |       |      |       |       |       |       |
| false  | true   | false  |              |      |       |       |      |   |          |          |                |      |      |      |      |       |       |       |      |       |       |       |       |
| false  | false  | false  |              |      |       |       |      |   |          |          |                |      |      |      |      |       |       |       |      |       |       |       |       |

1. CheckBox
2. ComboBox
3. Неповне розгалуження
4. Кон'юнкція
5. Повне розгалуження
6. Заперечення
7. Команда повного розгалуження
8. Команда неповного розгалуження
9. RadioButton

## II. Вивчення нового матеріалу.

### Цикл з післяумовою

Оператори **while** і **repeat** використовуються в тому випадку, коли деяку послідовність дій треба виконати кілька разів, причому необхідне число повторень під час розробки програми невідомо і може бути визначено тільки під час роботи програми.

#### Оператор циклу із післяумовою *repeat ... until*

Загальний вигляд оператора:

```
repeat  
  <Тіло циклу>  
until умова
```

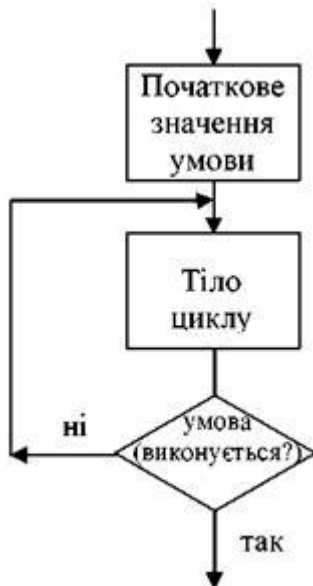
де **умова** - вираз логічного типу, визначальна умова завершення циклу.

Оператори тіла циклу виконуються доти, поки умова не стане істинним (False).

Зверніть увагу на відмінності:

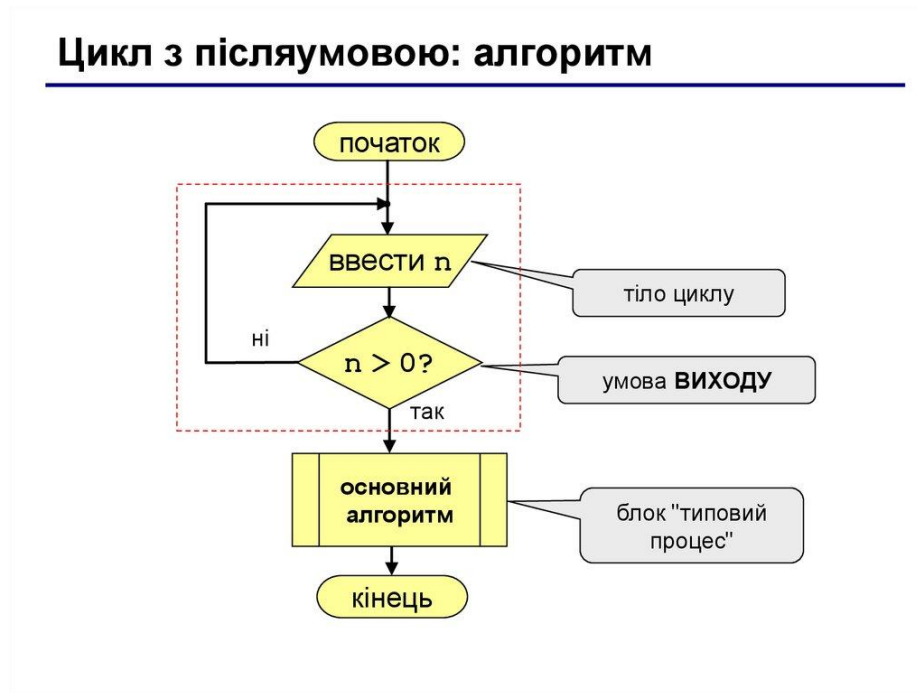
- в операторі **while ... do** вихід з циклу відбувається тоді, коли умова стане хибним,
- а в операторі **repeat ... until** - коли умова стане істинна.

Загальний приклад такого циклу у вигляді алгоритму:



Приклад знаходження числа  $n$  для виконання деяких дій над ним, якщо  $n > 0$ .

У даному прикладі буде відбуватись цикл допоки  $n$  не стане більшим за 0.

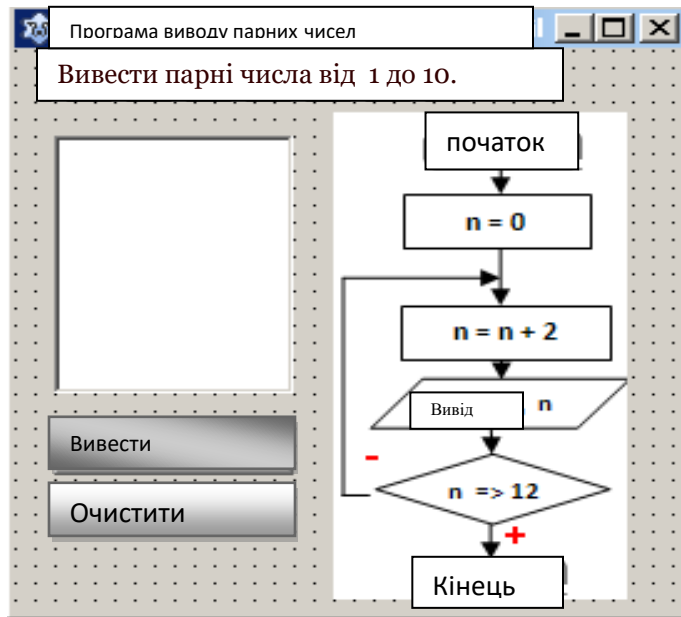


Виконайте програми в середовищі Лазарус. Якщо у вас немає на комп'ютері встановленого Лазарус, то перейдіть за посиланням: <https://lazarus-rus.ru/wp-content/uploads/2019/12/lazarus-2.0.6-fpc-3.0.4-win32.exe> - якщо у вас 32-бітна версія комп'ютера,

і <https://lazarus-rus.ru/wp-content/uploads/2019/12/lazarus-2.0.6-fpc-3.0.4-win64.exe> - якщо 64-біт.

**Вправа 1.** Вивести парні числа від 1 до 10.

В даному завданні вам необхідно створити в графічному редакторі( наприклад Paint) схожий алгоритм до поданого, вставити його у вигляді зображення на форму, розмістити інші необхідні компоненти які подані в прикладі, та прописати програмний код.



### Програмний код проекту

```

procedure TForm1.Button1Click(Sender: TObject);
var n: //опишіть якого типу може бути дане число;
begin
    n := 0;
    //Тут команда циклу з після умовою
    n := n + 1;
    listBox1.Items.add(' n = ' + inttostr(n));
    //Тут команда циклу з після умовою n => 10;
end;

//Опишіть процедуру очищення даних для другої кнопки!

```

**Вправа 2.** Виведіть на екран у вигляді таблиці номери і значення з двома цифрами після крапки перших і елементів числової послідовності, загальний елемент якої має вигляд  $12\cos i$ .

Домашнє завдання. Повторити з підручника ст.170-177. Виконайте завдання 1 та 2 та опишіть які компоненти ви використали у своїх формах.