

# ІНФОРМАТИКА

# Налагодження програм

8

За новою програмою



**Урок 32**



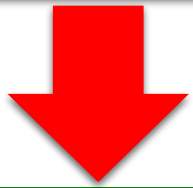
- 1. Яке призначення компілятора? Які особливості його роботи?**
- 2. Які помилки називають синтаксичними? Як вони виявляються на етапі компіляції проекту?**
- 3. Як відкрити раніше створений проект?**



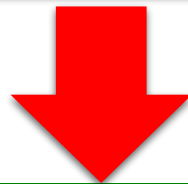


*Ви вже знаєте, що під час створення проекту можуть виникати синтаксичні помилки. Синтаксичні помилки, які не помітив автор проекту, знаходить компілятор на етапі компіляції під час запуску проекту на виконання. Потрібно виправити всі такі помилки і знову запустити на виконання нову версію проекту.*

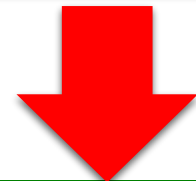
*Розрізняють три групи помилок:*



**Синтаксичні**



**Помилки під час виконання**



**Логічні**



Якщо виконання проекту неочікувано переривається або результат його виконання неправильний, то говорять, що проект містить **логічні помилки**.

Одним зі способів виявлення логічної помилки є створення **тестових наборів** вхідних даних. Для кожного із цих наборів даних визначають результат до виконання проекту і порівнюють його з результатом виконання проекту.

У разі їх розбіжностей потрібно знайти команди у проекті, виконання яких є причиною появи цих розбіжностей.



**Процес пошуку логічних помилок у проекті з використанням тестових наборів вхідних даних називають *тестуванням проекту*.**





Пояснимо цей спосіб на такому прикладі. Нехай потрібно скласти проект для обчислення значень функції  $y = \frac{2x-5}{x^2+1}$ . Процедура для обчислення значень цієї функції, у якій значення змінної  $x$  вводиться в поле, матиме такий вигляд:

```
procedure TForm1.Button1Click(Sender: TObject);  
var x, y: real;  
begin  
  x := StrToFloat(Edit1.Text);  
  y := (2*x-5)/(x*x+1);  
  Label1.Caption := FloatToStr(y);  
end;
```



Часто учні, записуючи команду присвоєння для обчислення значення відповідного виразу

$$y := (2*x-5)/(x*x+1),$$

забувають узяти знаменник (а то й чисельник) у дужки, тобто вводять, наприклад, команду або.

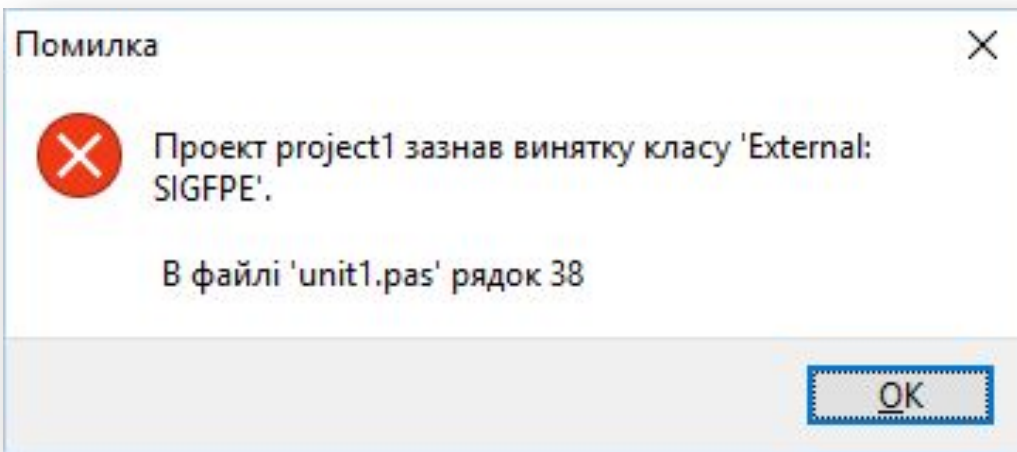
$$y := (2*x - 5)/x*x + 1$$

або

$$y := 2*x - 5/x*x + 1$$

Таку логічну помилку нескладно помітити, застосувавши вдалу добірку тестів.

**Так, під час тестування для  $x = 0$  стане очевидним, що проект не видає очікуваного правильного результату  $-5$ , а неочікувано перериває своє виконання, відкриваючи вікно з повідомленням про помилку.**



```
procedure TForm1.Button1Click(Sender: TObject);  
    var x, y: real;  
begin  
    x := StrToFloat(Edit1.Text);  
    y := (2*x-5)/x*x+1;  
    Label1.Caption := FloatToStr(y);  
end;
```

**Вікно з повідомленням про помилку**

**Текст процедури із зафарбованим рядком, що містить помилку**



**Щоб перервати виконання проекту в такому випадку, потрібно:**

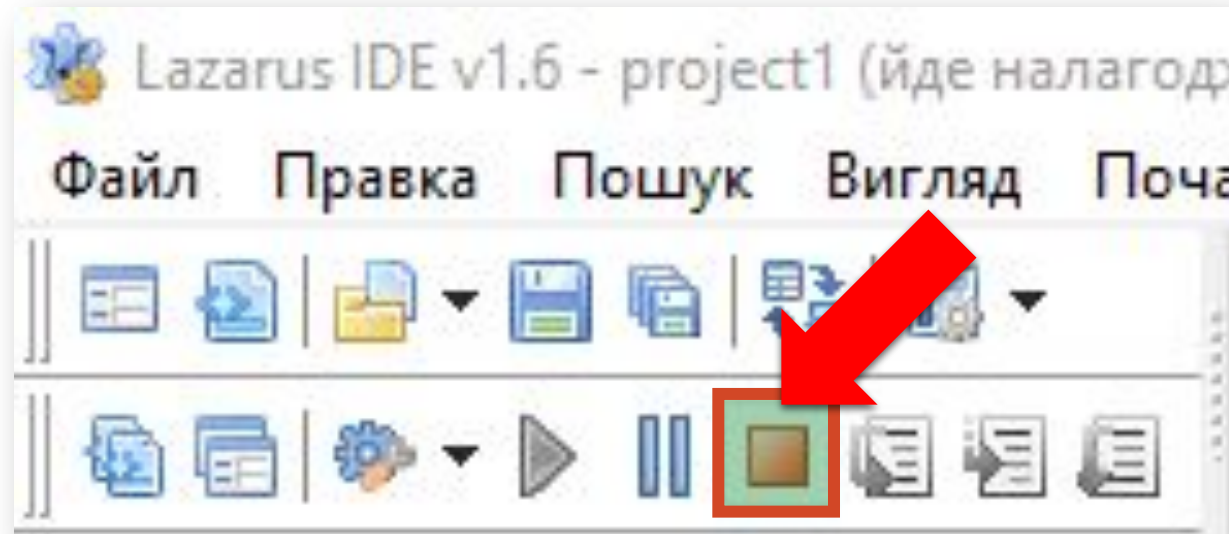
**Натиснути сполучення  
клавіш *Ctrl* + *F2***

**Вибрати кнопку  
*Завершити***

*Ctrl*

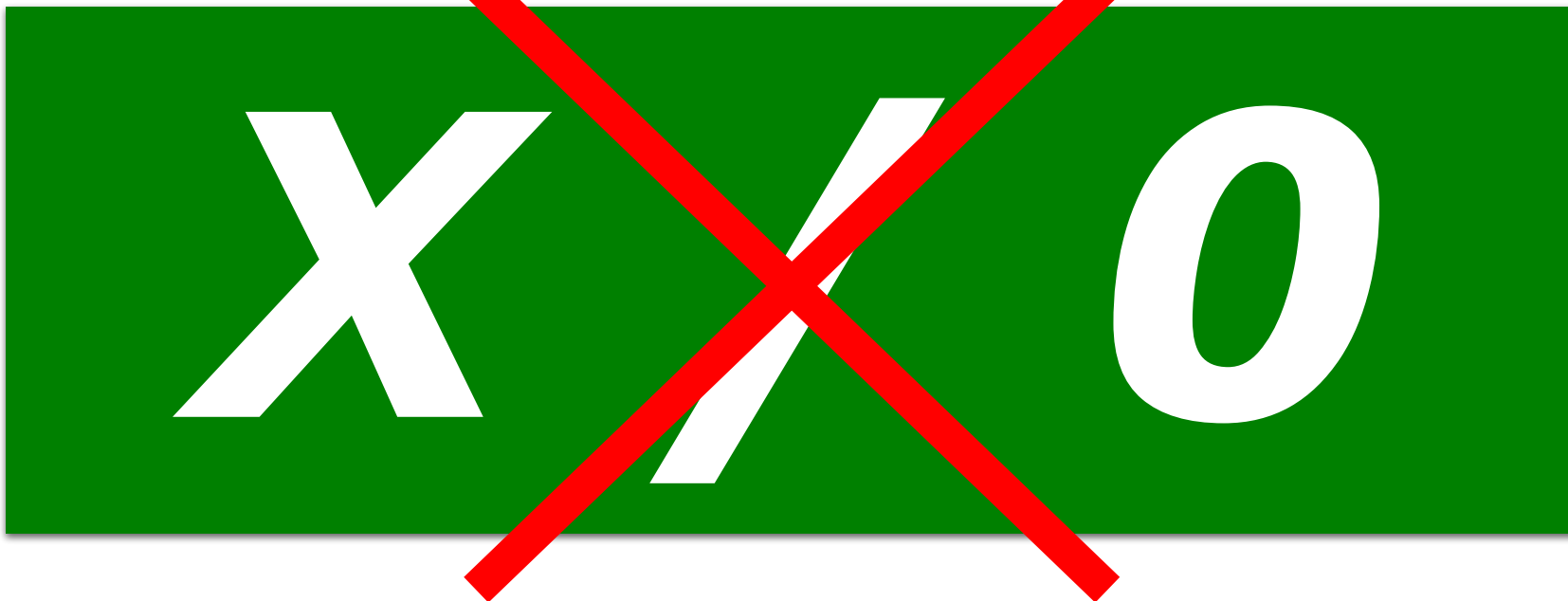


*F2*





*Далі потрібно зрозуміти, у чому полягає помилка, виправити її і знову запустити проект на виконання. У розглянутому випадку це відбувається тому, що в ході виконання проекту із цим значенням **x** має виконатися операція ділення на **0**, яку виконати **неможливо**.*

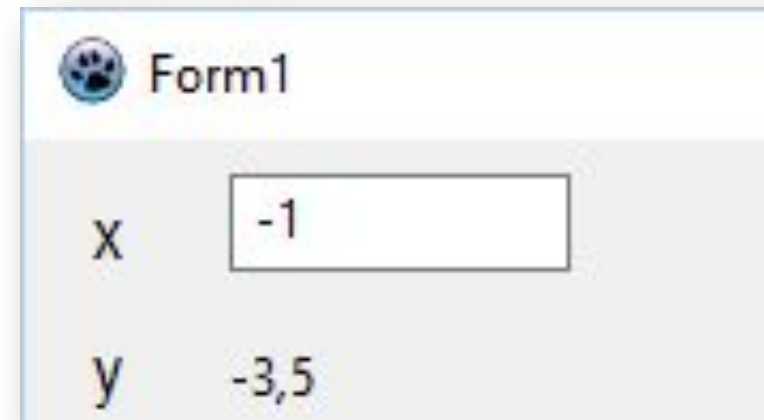


Якщо як тестовий приклад буде взято інше значення  $x$ , наприклад  $-1$ , то виконання проекту не переривається, вікно з повідомленням про помилку не відкривається, але як результат буде виведено число  $-6$ , хоча очікуваним правильним результатом є число  $-3,5$ . У такому випадку також потрібно уважно переглянути текст процедури, виявити помилку і виправити її.



Form1

x	<input type="text" value="-1"/>
y	-6



Form1

x	<input type="text" value="-1"/>
y	-3,5



Досить часто для полегшення пошуку логічних помилок у проект додають **коментарі**.



**Коментар** - це текст, який ігнорується компілятором і включається до тексту проекту з метою полегшити його розуміння та пошук логічних помилок.





Мовою програмування **Lazarus** — коментарі розпочинаються символами:

//

якщо коментар не займає  
більш ніж рядок

{ } або ( \* \* )

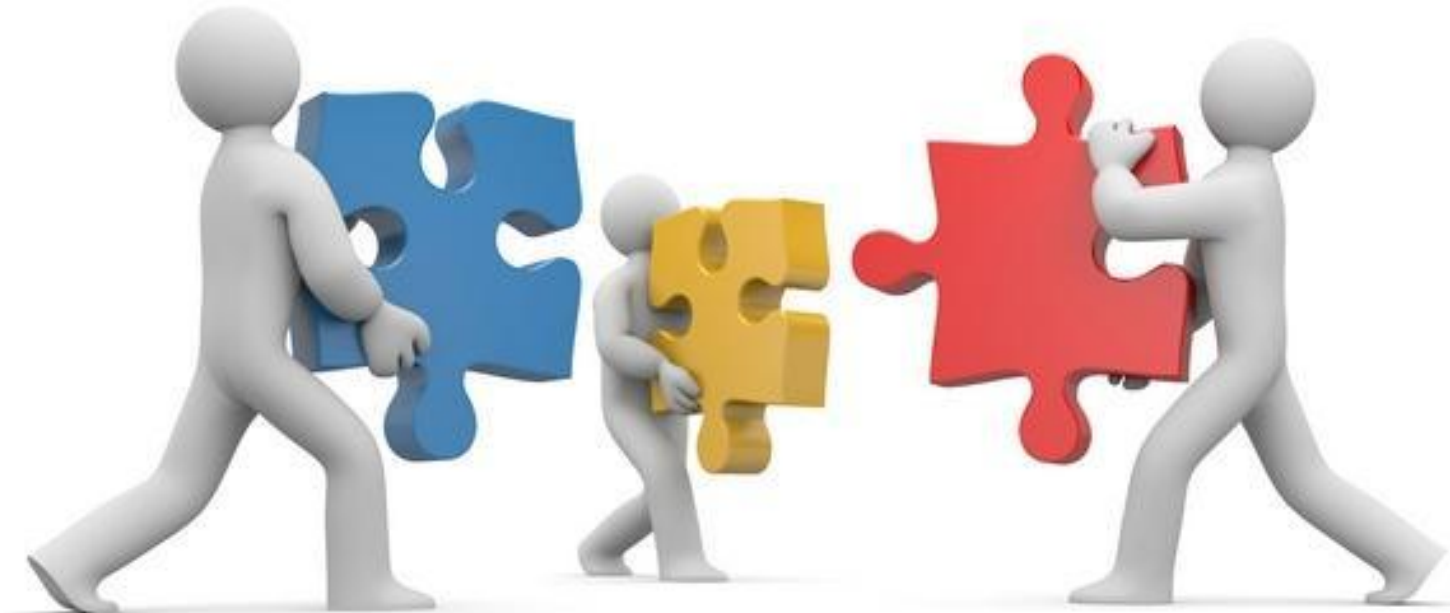
коментар, який може  
займати декілька рядків.

*S := Edit3.Text; // введення дати народження*

*S := Edit3.Text; { введення дати  
народження }*



Ще одним способом виявлення в проекті логічних помилок є його покрокове виконання, або **трасування**. **Крок виконання проекту** - це виконання команд, записаних в одному рядку проекту. У рядку може бути записано одну команду або кілька.





Щоб перейти до покрокового виконання процедури, потрібно встановити **точку зупинки** процедури і запустити проект на виконання. Якщо в деякому рядку проекту встановлено точку зупинки, то після запуску проекту на виконання в автоматичному режимі виконається частина проекту від самого початку і до цього рядка, не виконуючи команди саме цього рядка.

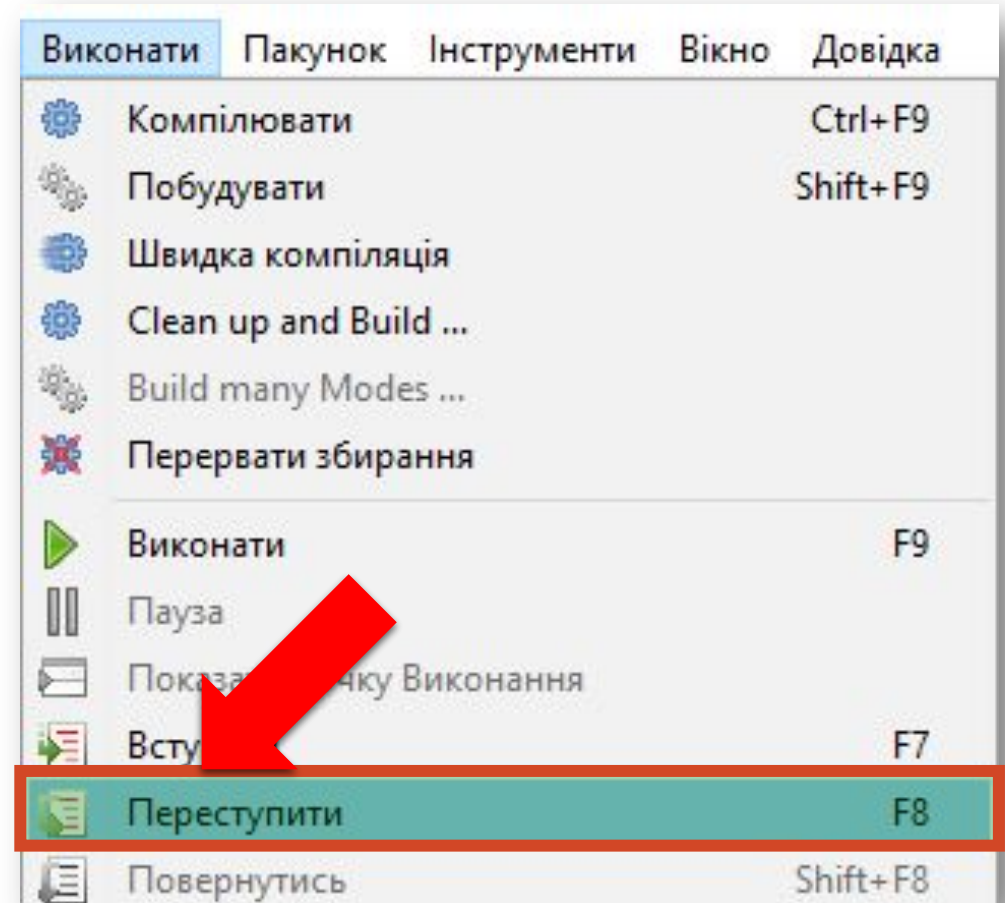
Для встановлення точки зупинки в потрібному рядку слід вибрати номер цього рядка в тексті проекту. Після цього фон вибраного рядка буде зафарбовано червоним кольором.

```
. procedure TForm1.Button1Click(Sender:
    var x, y: real;
begin
10  x := StrToFloat(Edit1.Text);
    y := (2*x-5)/(x*x+1);
    Label1.Caption := FloatToStr(y);
end;
```



Для виконання одного кроку проекту потрібно виконати **Виконати** ⇒ **Переступити**.

Або натиснути клавішу **F8**. Під час покрокового виконання фон рядка проекту, команди якого будуть виконані під час наступного кроку виконання, фарбується в сірий колір.







*У будь-який момент покрокове виконання проекту можна перервати:*

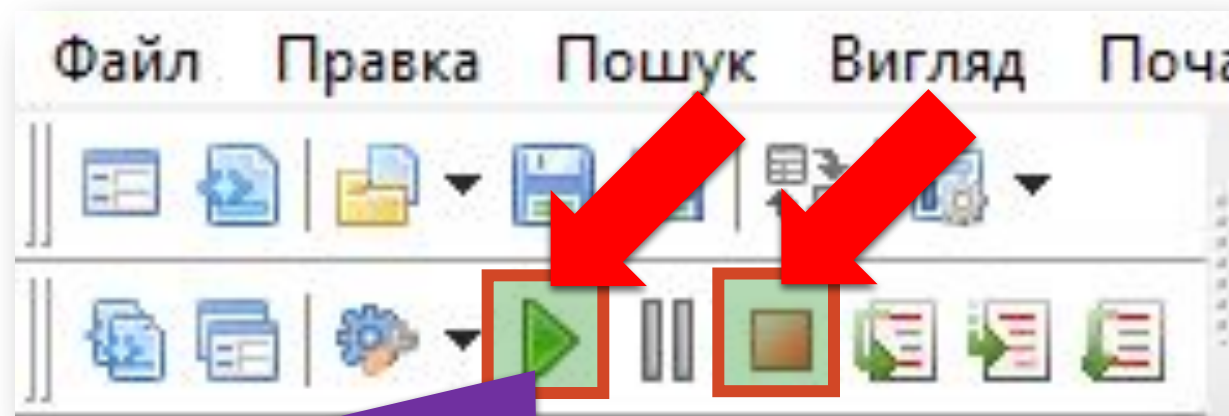
*Натиснути сполучення  
клавіш **Ctrl + F2***

*Вибрати кнопку  
**Завершити** або **Виконати***

*Ctrl*



*F2*



*Виконання проекту  
продовжується в  
автоматичному режимі.*



*Покрокове виконання проекту не має жодного сенсу, якщо в ході такого виконання не бачити і не спостерігати, як змінюються значення змінних. Саме таке спостереження може допомогти нам зрозуміти, які команди є причиною логічної помилки.*

*Щоб під час покрокового виконання проекту спостерігати за зміненням значень змінних, потрібно імена цих змінних помістити до вікна **Список Спостережень**. Це можна зробити, використавши вікно **Властивості Спостерігання**.*



Щоб помістити ім'я змінної до вікна **Властивості Спостереження**, потрібно натиснути сполучення клавіші **Ctrl + F5** або виконати **Виконати** ⇒ **Додати спостереження**. Ім'я змінної потрібно ввести в поле **Вираз** цього вікна і вибрати кнопку **Гаразд**.

Властивості Спостереження

Вираз:

Повторень:  Розряди:

Доступний  
 Дозволити Виклики Функцій  
 Викор. тип екземпляра класу

Стиль

Символ     Рядок     Десяткове  
 Шістнадцяткове     Непідписаний     Вказівник  
 Запис/Структура     Типовий     Ім'я

Список Спостережень

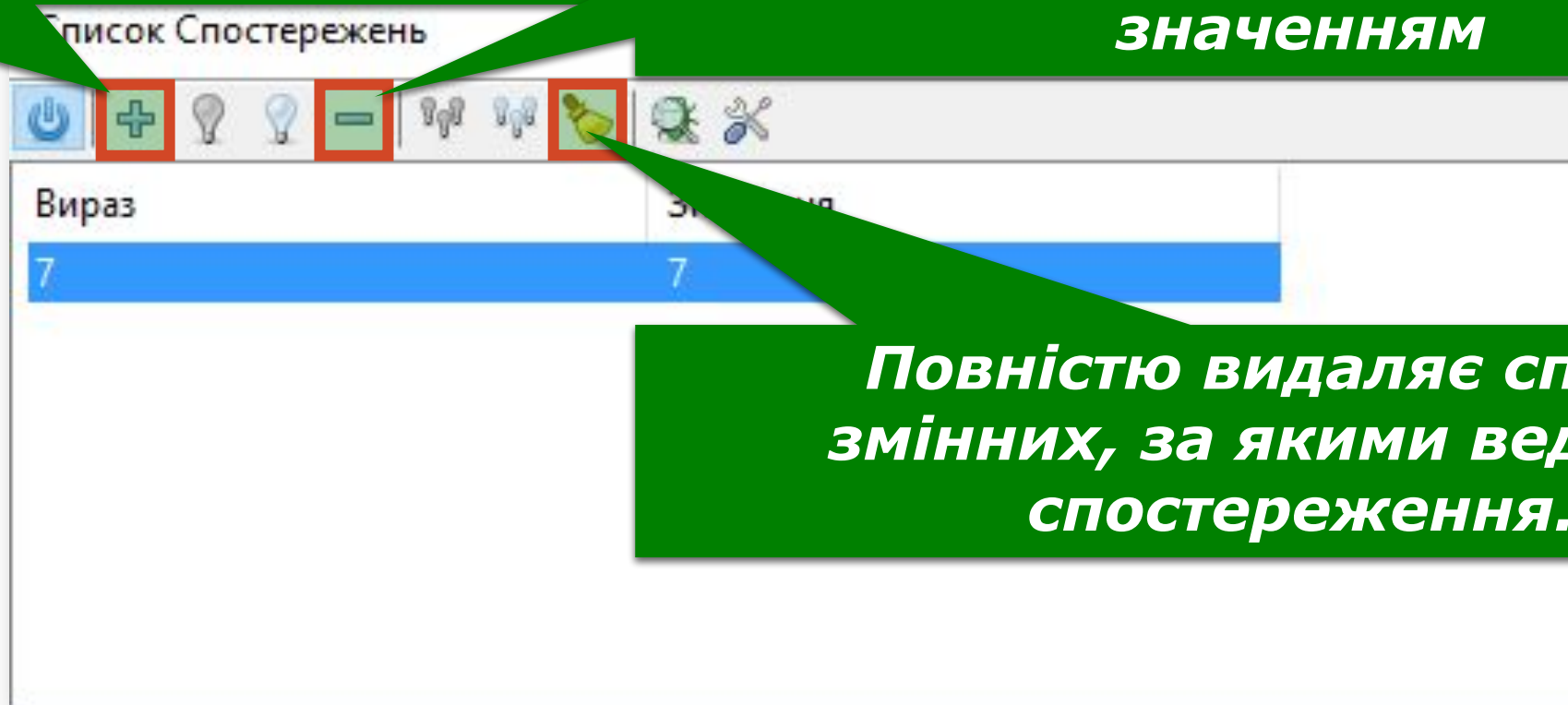
Вираз	Значення
7	7



## Вікно **Список спостережень**

**Відкриває вікно  
Властивості Спостерігання**

**Вилучає із цього вікна виділений  
рядок з іменем змінної та її  
значенням**



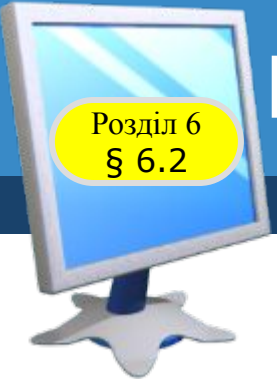
**Повністю видаляє список  
змінних, за якими ведеться  
спостереження.**



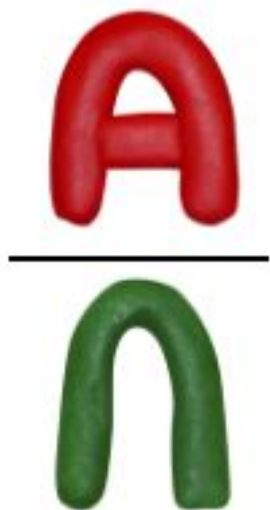
**Під час покрокового виконання проекту значення цих змінних змінюватимуться, що й можна спостерігати у вікні **Список Спостережень**.**

**Уважно стежачи за цими змінами і порівнюючи ці значення з очікуваними, можна помітити, у момент виконання якої команди проекту значення певної змінної стає неправильним. Саме ця команда й може бути причиною логічної помилки.**





# Розгадайте ребус



☞ 3 = E



**Налагодження**





- 1. У чому полягає процес тестування проекту?**
- 2. Поясніть, для чого виконується тестування проекту.**
- 3. Що таке коментар у проекті? Як можна коментар включити до тексту проекту?**
- 4. Поясніть, для чого в текст проекту включають коментарі.**
- 5. Для чого проект виконують покроково?**
- 6. Поясніть, у чому полягає процес покрокового виконання проекту.**





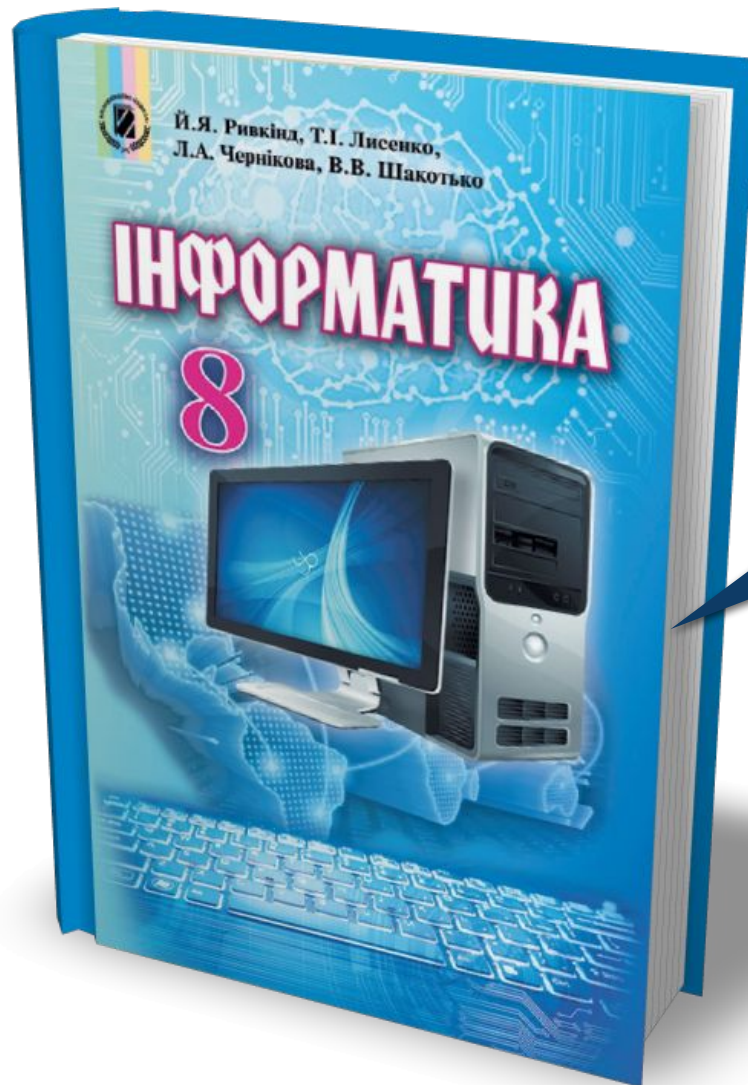
7. **Що таке крок виконання проекту? Як здійснити його виконання?**
8. **Що таке точка зупинки? Для чого вона використовується?**
9. **Як перервати режим покрокового виконання проекту?**
10. **Як переглянути значення змінних під час покрокового виконання програми?**





Розділ 6  
§ 6.2

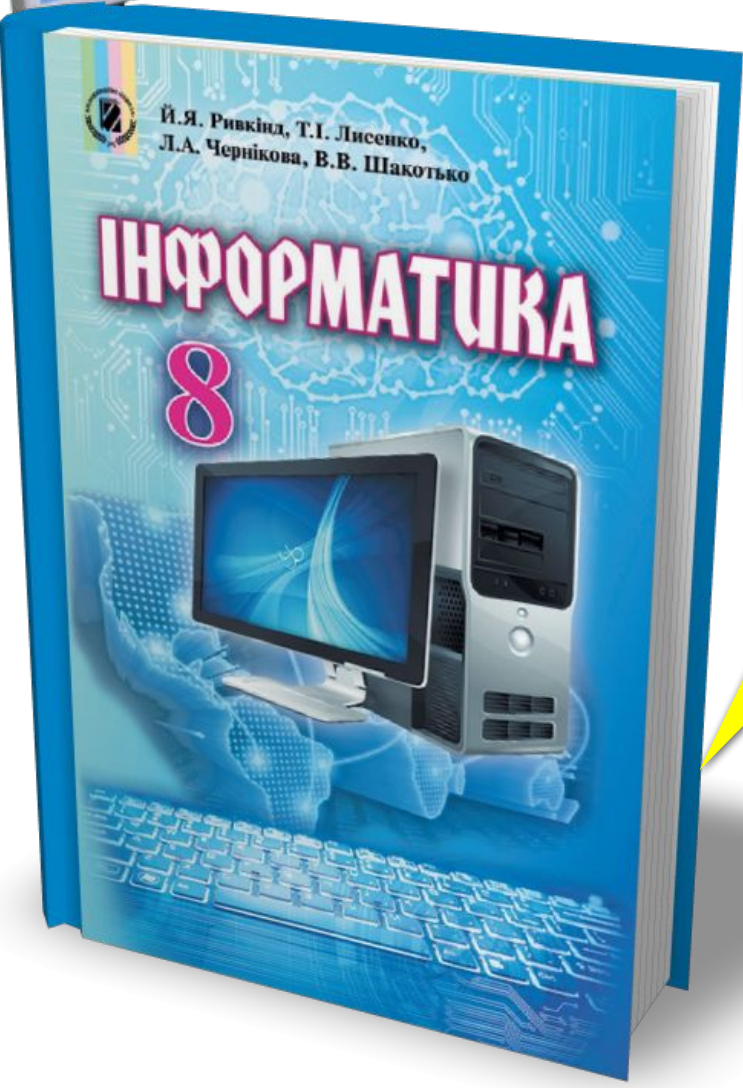
# Домашнє завдання



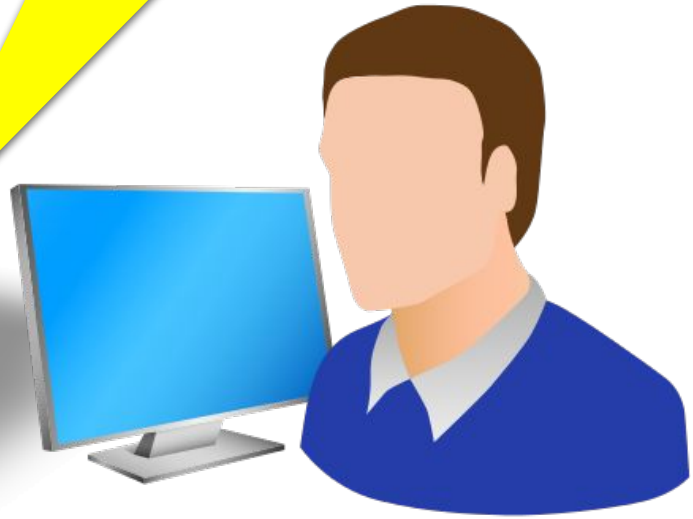
*Проаналізувати  
§ 6.2, ст. 188-195*

# Працюємо за комп'ютером

Розділ 6  
§ 6.2



**Сторінка  
193-195**



# ІНФОРМАТИКА

Дякую за увагу!

8

За новою програмою



**Урок 32**