



Вілен КОЛІСЕЦЬКИЙ,
вчитель інформатики ліцею №1 ім.Героя України М.Дзювельського
м.Шепетівки, вчитель-методист

Візуалізація алгоритмічних процесів як засіб ефективного навчання Python у середовищі Thonny

У статті розглядається методика навчання основ програмування на мові Python з використанням середовища Thonny. Основний акцент зроблено на візуалізації внутрішніх процесів роботи комп'ютера: звернення до пам'яті, обчислення виразів та передача управління. Адаптація класичного підходу «ручного виконання програм» до сучасних інструментів автоматизованого зневадження/налагоджувача (Debugging).

Проблематика початкового етапу навчання

Вивчення мови програмування в 7 класі є складним етапом у викладанні інформатики. Саме тут закладається фундамент алгоритмічного мислення. Проте, як свідчить педагогічний досвід, більшість учнів втрачають інтерес до теми програмування через високий рівень абстракції процесів. Код для багатьох учнів – це набір незрозумілих символів, за якими вони не бачать фізичної логіки роботи машини, не уявляють, як дані «рухаються» всередині пам'яті комп'ютера.

Головна причина неуспішності полягає в тому, що учні не розуміють, що і як відбувається під час виконання програми. У класичних підручниках ми бачимо лише «вхід» і «вихід», але «чорна скринька» всередині залишається закритою.

Технологія «прозорого комп'ютера»

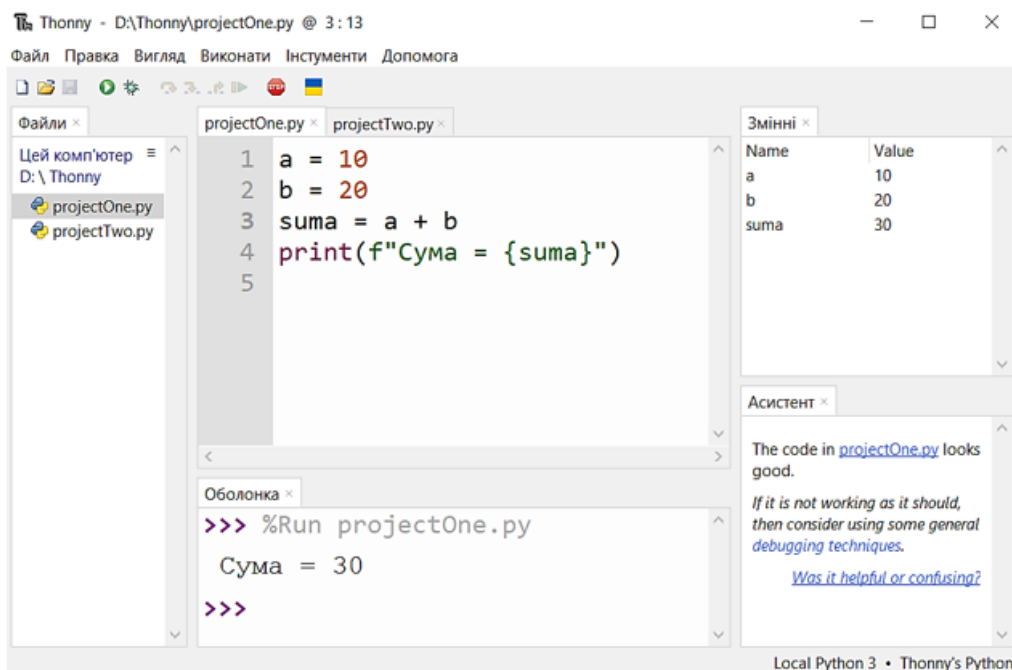
Л. Пул у книзі «Робота на персональному комп'ютері» стверджував: «Кращий спосіб навчитися – самостійно виконати операції». Для сучасного учня це означає не просто переписати код, а «пройти» його крок за кроком, спостерігаючи зміни в кожній комірці оперативної пам'яті.

Середовище Thonny ідеально підходить для реалізації цієї технології. На відміну від професійних IDE (PyCharm, VS Code), Thonny створено саме для освіти. Воно має функцію покрокової візуалізації процесів усередині виразів та команд.

Налагодження середовища Thonny

Thonny містить вбудований Python 3.10. Початковий інтерфейс користувача відкриває вікно редактора для введення тексту програми і позбавлений усіх функцій, які можуть відволікати початківців. Але для ефективного початкового навчання необхідно виконати такі налаштування:

- Tools/Options.../General – у списку Language вибрати український інтерфейс, для застосування перезавантажити середовище.



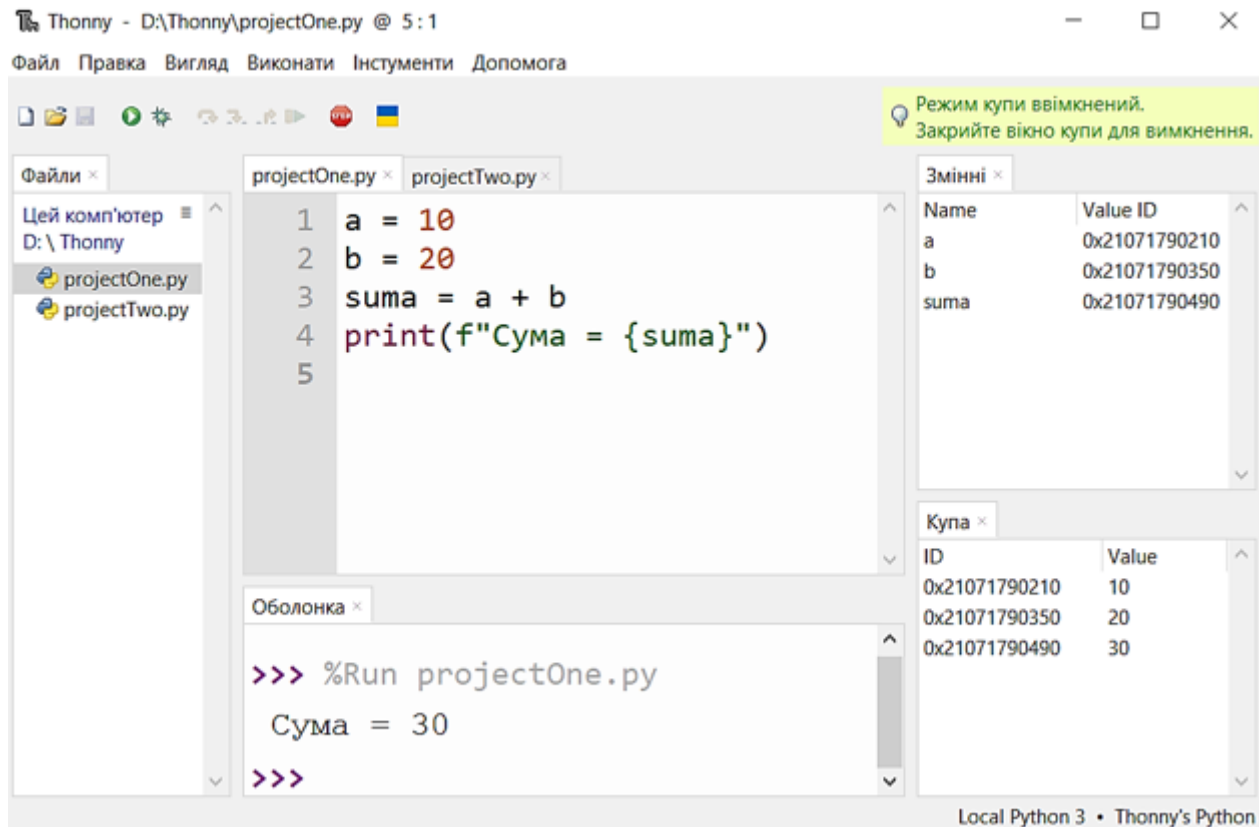
Вікно
середовища Thonny
з додатковими панелями

- **Вигляд/Файли** – панель для роботи з файлами;
- **Вигляд/Оболонка** – виконує багато функцій: показує результат виконання програми, інформацію про синтаксичні помилки, місце введення даних з клавіатури під час роботи програми, а також дослідження виконання команд (з їх миттєвим виконанням).
- **Вигляд/Змінні** – показує значення змінних під час

виконання програми (зручно використовувати в режимі налагоджувача (Debug...));

- **Вигляд/Купа** – демонструє дані в комітках пам'яті з фізичними адресами і які змінні на них вказують;

- **Вигляд/Асистент** – пояснення помилок, рекомендації як виправити.



Середовище Thonny з панеллю Купа (Heap)

Візуалізація пам'яті та змінних

Перше, що необхідно пояснити учню, що таке змінна, як комп'ютер «запам'ятовує» необхідні для роботи програми дані.

Методична вправа «Коробки з даними»

Мета: сформувати уявлення, що таке змінна, як зберігаються значення, різниця між змінною і даними.

Налаштування: увімкнути панель «Змінні» (Вигляд/Змінні), панель «Купа» вимкнути (Вигляд/Купа).

1. У вікні редактора учень пише: `x = 10`
2. Натискає кнопку "Запустити (F5)" ()
3. Спостерігає, як у вікні «Змінні» з'являється рядок:
`x = 10`
4. У вікні редактора в наступному рядку учень пише:
`x = 20`
5. "Запустити (F5)" () програму.
6. Спостерігає, як у вікні «Змінні» змінюється рядок на `x = 20`

Це дозволяє зрозуміти, що відбувається з даних. Учень бачить: якщо ми напишемо `x = 20`, старе значення 10 зникає, а 20 записується на його місце.

7. Увімкнути панель «Купа» (Вигляд/Купа).

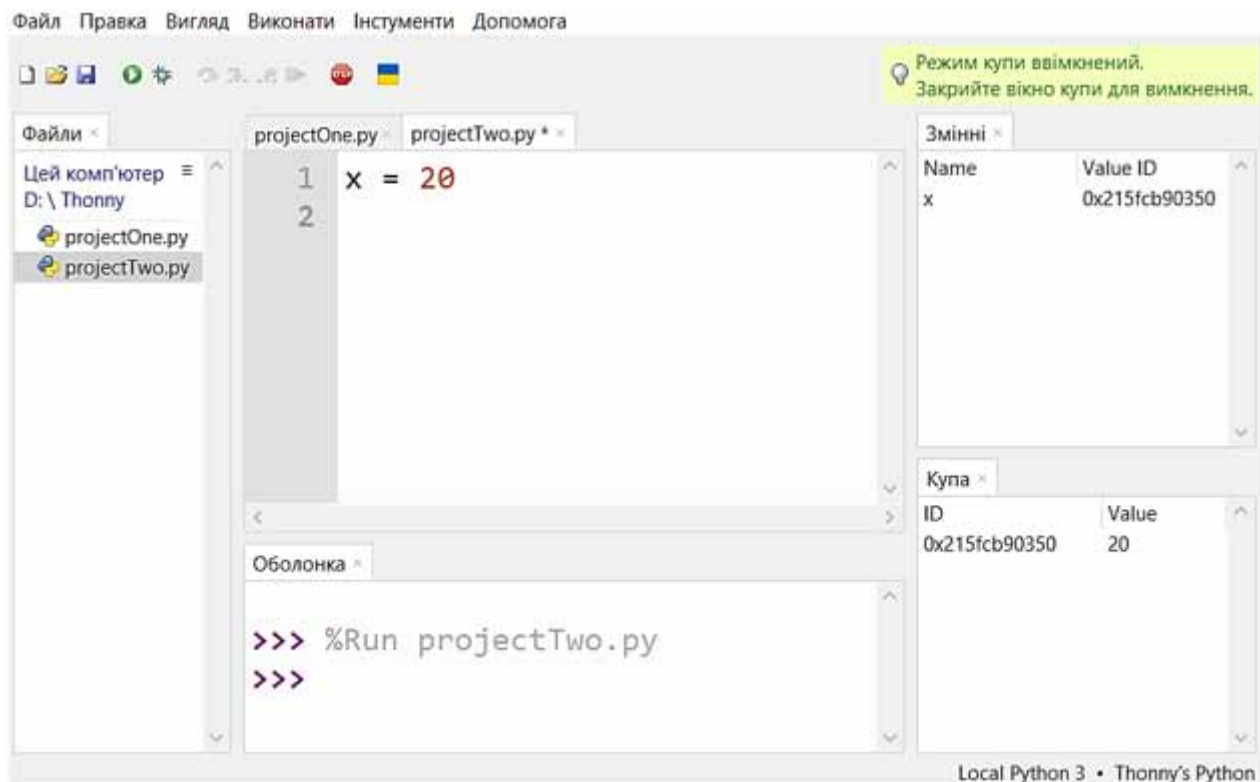
8. "Запустити (F5)" () програму.

Купа (heap) – це місце в пам'яті комп'ютера для збереження даних (об'єктів).

У вікні «Купа» учень спостерігає, що комп'ютер помістив значення 20 (Value) в комірку пам'яті з фізичним адресом 0x215fcd90350 (ID), яке є унікальним для кожного об'єкта. У вікні «Змінні» учень спостерігає, що змінна «x» вказує на туж саму адресу пам'яті.

Пояснення: ім'я змінної – це зручне посилання на комірку пам'яті, де зберігаються дані.

Запитання для дискусії: Що краще запам'ятати, ім'я змінної «x», яке ми самі обрали, чи фізичну адресу комірки пам'яті «0x215fcd90350»?



Демонстрація вікна середовища Thonny

Завдання для дослідження:Завдання 1. Написати текст програми і виконати:

```

1 num = 7
2 width = 10.5
3 myName = "Тарас"
4

```

1. Переглянути інформацію в панелях «Змінні» та «Купа»
2. Скільки даних зберігається в пам'яті?
3. Скільки комірок пам'яті вони займають?

Завдання 2*. Написати текст програми і виконати:

```

1 x = 10
2 y = x
3

```

1. Переглянути інформацію в панелях «Змінні» та «Купа»
 2. Скільки даних зберігається в пам'яті?
 3. Скільки комірок пам'яті вони займають?
- Команда `y = x` не створює новий об'єкт в пам'яті. Ми просто дали ще одне ім'я (створили ще одну змінну), яка вказує на той самий об'єкт в пам'яті.

Покрокове виконання як ключ до розуміння

Найпотужніший інструмент Thonny — клавіша  (F7) (Зайти/Step Into). Показує, що відбувається всередині кожного рядка програми.

Методична вправа «Візуалізації виразу»:

Мета: сформувати уявлення, що таке змінна, як обчислюються і зберігаються значення, як працює функція `print()` – виведення даних.

Налаштування: увімкнути панель «Оболонка» (Вигляд/Оболонка), увімкнути панель «Змінні» (Вигляд/Змінні), панель «Купа» вимкнути (Вигляд/Купа).

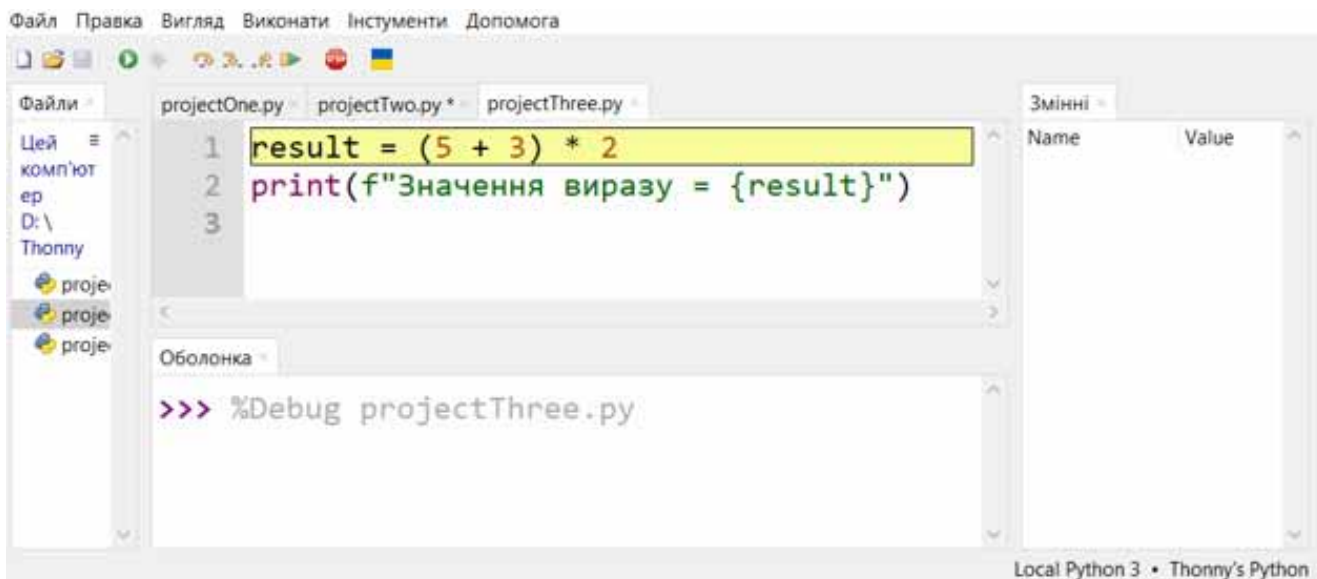
1. У вікні редактора учень набирає текст програми:

```

1 result = (5 + 3) * 2
2 print(f"Значення виразу = {result}")
3

```

2. **Обов'язково.** Зберегти текст програми у файлі з розширенням `.py`
3. Увімкнути режим налагоджувача  (Debug...)



Вікно середовища після увімкнення налагоджувача

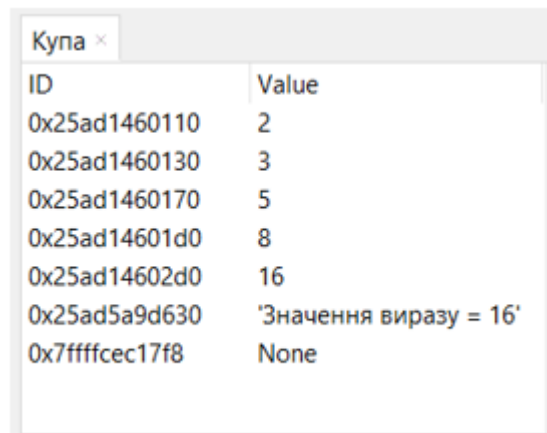
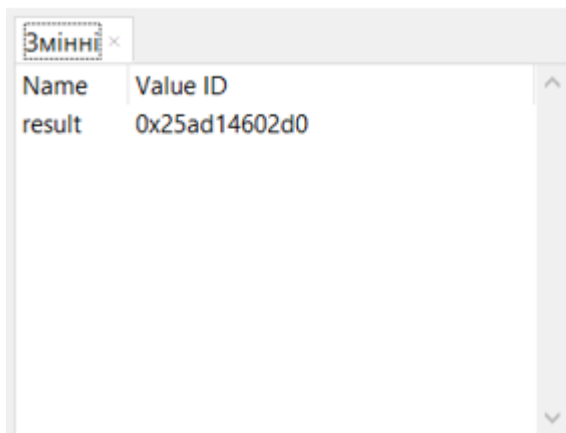
4. Учень послідовно натискає F7 (покрокове виконання) спостерігає за порядком виконання програми: порядком обчислення виразу, появою даних в панелі «Змінні», формуванням повідомлення та виведення результату роботи програми функцією print() у вікно «Обо-

лонка» та завершення процесу налагодження (кнопка стає неактивною).

5. Запитання до учнів:

- скільки змінних утворилось, яке в них значення?
- чи зберігає змінна result своє значення?

6. Увімкнути панель «Купа» (Вигляд/Купа).



Змінна result є вказівником на комірку пам'яті, в якій зберігається її значення 16. Інші дані та результати покрокового виконання програми також зберігаються в пам'яті комп'ютера. Якщо зупинити виконання програми або перезапустити програму, дані з пам'яті очищаються.

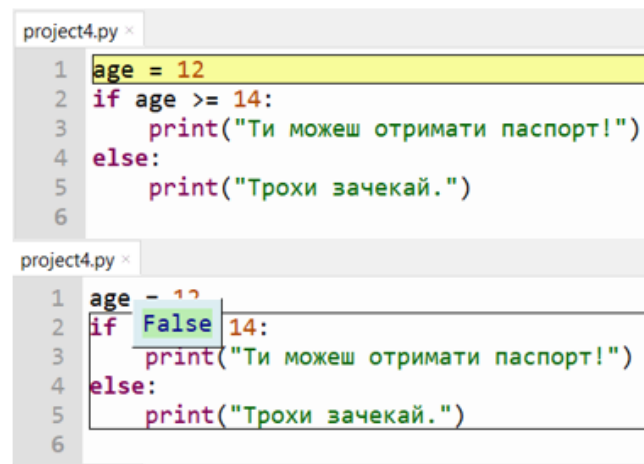
7. Повторити покрокове виконання з увімкненим вікном «Купа».

8. Проаналізувати зміни.

Керуючі конструкції: розгалуження та цикли

Розгалуження If-else

В режимі візуалізації Thonny підсвічує логічну умову (наприклад, `age >= 14`). Учень бачить, як в результаті перевірки вона перетворюється на True або False. Це важливо для розуміння того, чому програма обирає наступну дію.



```

project4.py ×
1 age = 12
2 if age >= 14:
3     print("Ти можеш отримати паспорт!")
4 else:
5     print("Трохи зачекай.")
6

```

Скріншоти покрокового виконання програми з розгалуженням

Цикли for та while

Це найскладніша частина для 7 класу. Використовуючи методику діаграм, ми просимо учнів вести «протокол циклу»:

1. Номер ітерації.
2. Значення лічильника (ітератора).
3. Дія в тілі циклу.

Thonny це автоматизує: учень бачить, як стрілка управління повертається до заголовка циклу for, а значення лічильника оновлюється у вікні «Змінні».

```

project5.py ×
1 print("Рахуємо до трьох:")
2 for i in [1, 2, 3]:
3     print(i)

project5.py ×
1 print("Рахуємо до трьох:")
2 for i in [1, 2, 3]:
3     print(i)

project5.py ×
1 print("Рахуємо до трьох:")
2 for i in [1, 2, 3]:
3     print(i)

```

```

project5.py ×
1 print("Рахуємо до трьох:")
2 for i in [2, 2, 3]:
3     print(i)

project5.py ×
1 print("Рахуємо до трьох:")
2 for i in [1, 2, 3]:
3     print(2)

```

Покрокове виконання програми з циклом

Висновки

Використання режиму зневадження (Debug) в Thonny робить програмування наочним, динамічним та цікавим дослідженням.

Наочність знімає когнітивне навантаження. Debug перетворює пошук помилок на цікаву гру-дослідження. Результативність вивчення основ програмування зростає на 30-40% за рахунок розуміння внутрішньої логіки процесів.

Поради

Візуалізація – ключ до успіху: використовуйте покрокове виконання для пояснення нових конструкцій та структур.

Не затагуйте з візуалізацією: Як тільки учень зрозумів принцип – дозволяйте писати код без покрокового розбору.

Помилка – це навчання: Використовуйте Debug для пошуку логічних помилок, коли програма працює, але видає не те, що треба.

