

## ОГЛЯД ВІЗУАЛЬНИХ МОВ ПРОГРАМУВАННЯ

ст. викл. Шафроненко Є.О., викладач Нестерук О.Г, викладач Бабич А.О.

Харківський національний університет радіоелектроніки,  
кафедра медіаінженерії та інформаційних радіоелектронних систем,  
м. Харків, Україна;

Відокремлений структурний підрозділ «Полтавський політехнічний  
фаховий коледж Національного технічного університету «Харківський  
політехнічний інститут», м. Полтава, Україна  
e-mail: yevhenii.shafronenko@nure.ua

**Abstract.** This paper reviews the programs of visual programming. It classifies visual programming languages into three categories based on how they use graphics to represent the syntax of the language. It also divides visual programming languages that rely on flowcharts into four subcategories: data flow programming, state machines, behavior trees, and event-based rules. We suggests that the best way to learn the main types of visual programming languages is to start with languages like Blockly and end with the LabVIEW language.

**Ключові слова:** візуальні мови програмування, Blockly, LabVIEW, CASE-системи, SCADA-системи.

**Вступ.** Сучасне програмування використовує візуальні методи для спрощення та пришвидшення створення програм. Візуальні мови програмування (ВМП) – це підхід, за яким програми будуються з візуальних компонентів, наприклад, блоків та стрілок, а не з текстового коду. Цей етап в історії програмування дав розробникам більше можливостей та зробив програмування доступнішим. Зараз існує понад 70 різноманітних візуальних мов програмування, які мають різне призначення, функції та сфери застосування.

**Основна частина.** Візуальне програмування, яке включає створення програм за допомогою графічних засобів та автоматичну генерацію програмного коду, може бути реалізоване через різноманітні інструменти та підходи. Цей метод відділяє процес програмування від прямого написання тексту програм та дозволяє використовувати візуальні елементи для визначення структури програми та її функціональності.

Існують різні системи та середовища, які впроваджують концепції візуального програмування. Деякі з них спрямовані на спрощення процесу створення програм для початківців, тоді як інші ставлять за мету полегшити взаємодію між розробниками та замовниками.

Однак на етапі класифікації можуть виникати питання через неоднозначність та величезний асортимент інструментів візуального програмування. Вони можуть різнитися за концепціями та рівнем складності. Класифікація ускладнюється через різні підходи та філософії,

які вони втілюють. Важливо враховувати, що хоча візуальне програмування може полегшити розробку (для певних завдань), ускладнення може виникнути при роботі над складними або масштабними проектами.

Візуальні мови програмування поділяються за рівнем складності та функціональністю. Залежно від характеристик, візуальні мови можна розділити на кілька категорій, як показано на рисунку 1.

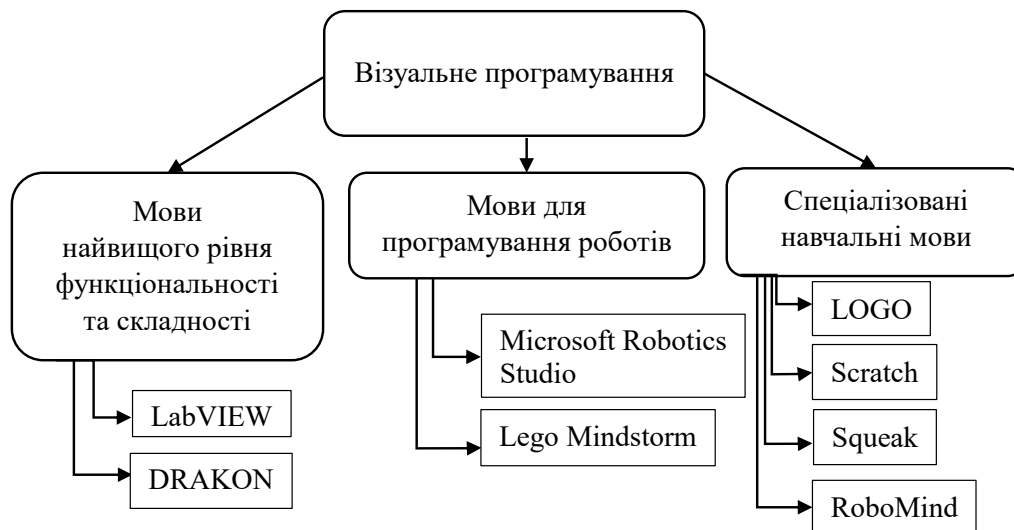


Рисунок 1 – Класифікація візуальних мов програмування

Ці категорії відображають відмінності, спрямованість та особливості візуальних мов програмування, а також їхню придатність для конкретних завдань та цільових аудиторій. Візуальне програмування може розглядатись в двох аспектах:

- графічна мова програмування, яка використовує графічні елементи для конструювання програми. Кожен графічний елемент представляє конкретну операцію або дію, а їх поєднання визначає послідовність виконання програми. Приклади таких мов – LabVIEW, Blockly тощо;

- візуальні засоби розробки, які включають інструменти та середовища для створення програм або розробки додатків, такі як CASE-системи, системи швидкої розробки додатків, SCADA-системи для програмування мікроконтролерів та інші.

Такі інструменти дозволяють інженерам і програмістам розробляти програми, не працюючи безпосередньо з кодом. Незалежно від використаного інструменту та обраної мови, використовуються графічні елементи, але їх функції та застосування можуть значно відрізнятись.

Візуальні мови програмування знаходять широке застосування у сфері навчання програмуванню, особливо серед початківців. Прикладами таких мов є App Inventor, Blockly, Scratch, Snap!, які спрощують процес навчання

та роблять його доступнішим.

У візуальних мовах програмування, основні елементи програми представлені у вигляді блоків, які користувач складає разом, визначаючи потік виконання програми. Блоки зазвичай групуються за типами у каталозі, що забезпечує зручний доступ до функцій без необхідності вивчення документації. Такий підхід розв'язує проблеми, пов'язані з вивченням синтаксису текстових мов програмування, дозволяючи зосередитися на логіці програми. Середовище розробки для таких мов зазвичай має спрощений інтерфейс, за допомогою якого користувач може легко запускати свій код на виконання та переглядати отримані результати. Для деяких візуальних мов є можливим «переклад» коду в текстові мови програмування, такі як JavaScript, Python, PHP, Lua, Dart, або збереження створеної логіки у вигляді XML-файлу.

Існують також візуальні мови, які використовують графічні елементи аналогічні тим, що використовуються у блок-схемах. Прикладами таких мов є Bonita BPM, Discovery Machine, Flowgorithm, Flowhub, Grafcet, Raptor, WebML, Widget Workshop тощо. Програми, створені за допомогою цих мов, являють собою послідовність дій заданих стандартними блоками, включаючи розгалуження з умовами для вибору подальших блоків виконання.

Такий метод використання візуальної граматики спрощує розуміння синтаксису мови, однак логічні конструкції, створені цим способом, можуть бути обмеженими в порівнянні з традиційними мовами програмування. Також важливо мати усвідомлення того, що саме міститься всередині блоків програми (дані, команди, дії), і цей вміст не завжди можна змінити за допомогою графічного інтерфейсу.

При програмуванні кінцевих автоматів у графічному вигляді, блоки являють собою стани, а зв'язки між ними відображають переходи. Програміст визначає стани і переходи між ними, ініційовані умовами. При зміні стану, спрацьовує відповідна інструкція, тобто створюється потік виконання у відповідності до блок-схеми, яка з'єднує стани, і визначає тригери для їх зміни. Таким чином, логіка керування контролюється безпосередньо через переходи і їх умови, замість фокусування на внутрішніх деталях блоків, що дозволяє створювати складніші інструкції, використовуючи інкапсульовану поведінку блоків. До таких мов можна віднести EKI One, NodeCanvas, Unity3D Mecanim Animator Controller, xaitControl тощо.

Мови програмування поведінки на основі дерев використовують візуальну граматику, яка складається з блоків і зв'язків між ними. Ці зв'язки визначають потік виконання і обробляють та повертають статус до попереднього блоку, що схоже на генеалогічне дерево. Такий підхід дозволяє визначати логіку програми через графічний інтерфейс, не вимагаючи додаткової візуальної граматики або редагування тексту.

Приклади таких мов – RAIN, AngryAnt Behave, Behavior3, NodeCanvas.

Ще одним типом візуальних мов програмування є мови, що базуються на правилах, побудованих на подіях. У таких мовах програміст визначає правила типу «якщо відбудеться ця подія, виконати такі дії». Правила спрацьовують кожен раз, коли умова виконується, і виконують вказані інструкції. Зазвичай такий підхід використовується під час розробки комп'ютерних ігор та симуляторів. Прикладами таких мов є Blender Game Engine, Construct 2, Kodu, Zapier тощо.

**Висновки.** В роботі виконано аналіз візуальних мов програмування, класифікація їх за форматом графічного представлення, та виділено дві основні категорії: мови, що використовують блоки, та мови, що базуються на блок-схемах. Висвітлено різні підтипи, такі як ті, що використовують потік даних, кінцеві автомати, дерева поведінки та правила на основі подій.

При розгляді основних типів візуальних мов програмування рекомендується починати з мов, таких як Blockly, і поетапно продовжувати вивчення, переходячи до складніших мов, наприклад, LabVIEW.

#### **Список використаних джерел.**

1. Dehouck, R. E. M. I. (2015). The maturity of visual programming. Режим доступу: <http://www.craft.ai/blog/the-maturity-of-visualprogramming>.
2. Завадський, І. О., & Заболотний, Р. І. (2009). Основи візуального програмування. К.: Видавнича група ВНУ, 272.
3. Семеренко, В. П. (2010). Візуальне програмування.
4. Вакалюк, Т. А. (2013). Візуальне програмування. Навчально-методичний посібник для студентів фізико-математичного факультету.
5. Лозовська, Л. І., Бандоріна, Л. М., Савчук, Р. В., & Климкович, Т. О. (2022). Візуальне програмування. Український державний університет науки і технологій, Дніпро.
6. Круглик, В. С., & Марчук, М. С. (2022). Сучасні тенденції у вивченні візуального програмування майбутніх інженерів-програмістів. In Мехатронні системи: інновації та інжиніринг. Київський національний університет технологій та дизайну.
7. Tsai, C. Y. (2019). Improving students' understanding of basic programming concepts through visual programming language: The role of self-efficacy. *Computers in Human Behavior*, 95, 224-232.