

УДК [378.091.2:004-057.21]:004.43

## ОГЛЯД ВІЗУАЛЬНИХ МОВ ПРОГРАМУВАННЯ ДЛЯ НАВЧАННЯ МАЙБУТНІХ ІНЖЕНЕРІВ-ПРОГРАМІСТІВ

Марчук М.<sup>1</sup>, Крашеніннік І.<sup>2</sup>

Мелітопольський державний педагогічний університет  
імені Богдана Хмельницького, м. Мелітополь,

<sup>1</sup>e-mail: [nickolay.marchuk@gmail.com](mailto:nickolay.marchuk@gmail.com)

<sup>2</sup>e-mail: [iryna.krasheninnik@gmail.com](mailto:iryna.krasheninnik@gmail.com)

**Постановка проблеми.** Сучасні технології програмування намагаються інтегрувати візуальний підхід до розробки програм. У результаті чого текстові мови програмування отримали візуальні засоби розробки і стали інтегрованими середовищами розробки (ICP, англ. Integrated development environment (IDE), як MS Visual Studio, що являє собою візуальне середовище програмування. Проте з 1970 років почали з'являтися візуальні мови програмування (ВМП), які дозволяють користувачам розробляти програми за допомогою маніпуляцій з графічними елементами (блоками, стрілками), що використовуються як елементи синтаксису мови, на відміну від написання тексту вихідного коду. На сьогодні виділяють більше 70 візуальних мов програмування [2]. Значний перелік наявних на сьогодні мов програмування викликає труднощі у процесі викладання дисципліни «Візуальне програмування», що пов'язані із вибором технологій для вивчення майбутніми інженерами-програмістами.

**Аналіз останніх досліджень і публікацій.** Як зазначає Толстова Н.С. візуальне програмування, що передбачає створення додатків за допомогою наочних засобів, при цьому програміст не створює текст програми, а показує, що повинно вийти в результаті, а текст програми генерується автоматично за допомогою візуального прототипу (оригінал, початковий зразок) [5]. Неузгодженість виникає на етапі класифікації. Гасько Р.Т. за рівнем складності розділяє візуальні мови програмування на мови найвищого рівня функціональності та складності (LabView, DRAKON); візуальні мови для програмування роботів, (Microsoft Robotics Studio, Lego Mindstorm); спеціалізовані навчальні мови для навчання основам програмування (LOGO, Scratch, Squeak, RoboMind та ряд інших) [3]. Величко В.Є. розрізняє два види візуального програмування: графічна мова програмування (мова зі своїм синтаксисом); візуальні засоби розробки (засоби проектування інтерфейсів, CASE-системи швидкої розробки додатків, SCADA-система для програмування мікроконтролерів тощо) [4, с. 52].

На сьогодні у дослідженнях не достатньо висвітлено питання доцільного вибору технологій для вивчення у рамках дисципліни «Візуальне програмування».

Метою статті є огляд візуальних мов програмування з метою вибору найбільш підходящої(их) для викладання у процесі професійної підготовки майбутніх інженерів-програмістів.

**Виклад основного матеріалу.** Однією з сфер, де ВМП досягли успіху є навчання, тому що вони допомагають користувачам зрозуміти як створити комп'ютерну програму. До таких мов відносять App Inventor, Blockly, Hopscotch, Microsoft TouchDevelop, Scratch, Snap! Та ін.. Ці ВМП покликані зробити програмування більш доступним, зокрема, щоб зменшити труднощі, з якими стикаються новачки, коли вони починають програмувати. У них основні елементи програмування, як правило, представлені у вигляді блоків, з візуальними орієнтирами для використання їх один з одним. Таким чином, для створення програми користувач збирає блоки, розташовуючи їх так, щоб вони правильно стикувалися один з одним відповідно до логіки чи алгоритму розроблюваної програми. Завдяки цьому недопустимі вирази може бути просто неможливо зібрати. Крім того, блоки, як правило, представлені у каталозі (розділені за типами), і користувачу не потрібно читати документацію, щоб зрозуміти, які функції доступні у цій мові програмування. Така концепція мови позбавляє проблем, які з'являються при вивченні синтаксису мов програмування, а це дозволяє їм зосередитися на логіці своєї програми, а не на тому, яку синтаксичну конструкцію обрати, як прописати властивості (атрибути), як правильно оформити запис відповідною мовою програмування. Слід зазначити, що середовище розробки для таких мов часто має спрощене середовище виконання програм, де користувач може швидко і легко запускати свою програму і бачити результати. У деяких візуальних мовах програмування такого типу код може інтерпретуватися на різні текстові мови програмування, наприклад, як у Blockly код інтерпретується на такі мови як JavaScript, Python, PHP, Lua, Dart та XML.

Використовувані у таких мовах форми, кольори, відступи дозволяють швидко отримати уявлення про те, що програма робить, тобто розроблювана програма є зручною для читання та рефакторингу. Адже структура блоків дозволяє легко змінити графічний код, на відміну від рефакторингу тексту на основі текстової мови.

Інший тип ВМП подібний до попереднього, але замість блоків використовуються графічні елементи, що використовуються у блок-схемах. Це, наприклад, такі мови як Bonita BPM, Discovery Machine, Flowgorithm, Flowhub, Grafcet, Raptor, WebML, Widget Workshop та ін.. Програми являють собою спрямовану послідовність виконання дій між блоками, де також використовується розгалуження з умовою, щоб вибрати, який блок виконується далі. Такий акцент на простій візуальній граматиці дозволяє швидко зрозуміти синтаксис мови, проте логічні конструкції, що створюються безпосередньо за допомогою ВМП, є обмеженими. Також багато що залежить у програмі від того, що знаходиться всередині блоків (які дані, команди, дії), а це не може бути змінено за допомогою графічного інтерфейсу.

Деякі формати, подібні до блок-схем, мають розширену граматику мови, яка надає можливість писати більш складні інструкції безпосередньо з графічного інтерфейсу. Це мови для програмування потоку даних, кінцевих автоматів та поведінки дерев. При програмуванні потоку даних блоки являють собою функції, що зв'язані між собою потоком, а також входом і виходом. До таких мов можна віднести AudioMutch, DRAGON, LabVIEW, NETLab Toolkit, OpenRTM, Simulink, Virtools та ін. Зокрема, LabVIEW – мова графічного програмування, в якій для створення додатків використовуються графічні образи (іконки) замість традиційного текстового коду. Від користувача пакета не потрібно знань мов програмування, але поняття про алгоритм, циклі, вихід за умовою і т.п. звичайно потрібно. Всі дії зводяться до простої побудови структурної схеми програми в інтерактивній графічній системі з набором всіх необхідних бібліотечних образів, з яких збираються об'єкти [6].

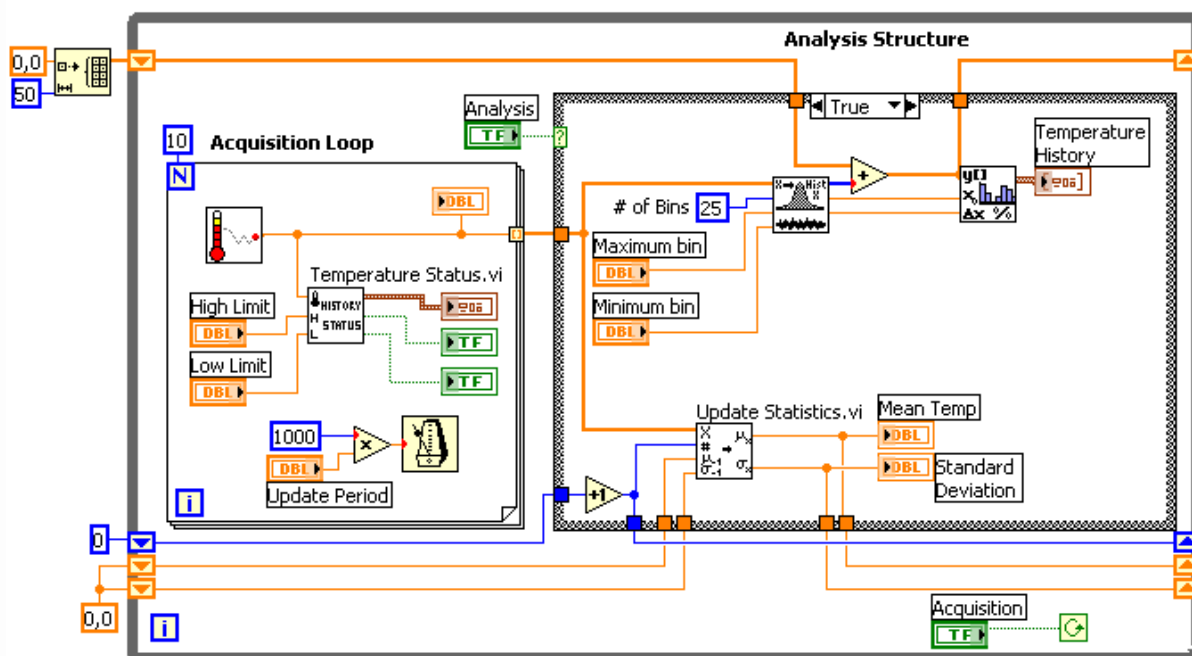


Рис. 1. Графічна програма мовою LabVIEW

При програмуванні кінцевих автоматів у графічному вигляді блоки являють собою стан, а зв'язки між ними представляють переходи. Програміст програмує стан і переходи між станами, які ініціюються умовами, а коли змінюється стан, спрацьовує інструкція, тобто розробляє потік виконання як за допомогою блок-схеми, що зв'язує стани, а також шляхом визначення тригерів зміну станів. Таким чином, логіка вибору контролюється безпосередньо через переходи і їх умови, замість того, щоб зосереджуватися всередині блоків, що дає можливість створювати більш складні інструкції, використовуючи інкапсульовану поведінку блоків. Перелік таких мов не значний: EKI One, NodeCanvas, Unity3D Mecanim Animator Controller, xaitControl.

До мови програмування поведінки дерев відносяться ті, візуальна граматику яких складається з блоків і зв'язків між блоками, які визначають

потік виконання, проте на відміну від формату блок-схем, блоки обробляють і повертають статус до попереднього блоку на зразок до генеалогічного дерева. Це дає можливість визначити логіку програми з графічного інтерфейсу без додавання додаткової візуальної граматики чи редагування тексту, використовуючи блоки для керування потоком. Таким чином, порівнюючи з кінцевими автоматами, розробка програми таким чином є більш зрозумілою і може бути зроблена шляхом перегрупування зв'язків між блоками [7]. Серед таких мов доцільно назвати такі як RAIN, AngryAnt Behave, Behavior3, NodeCanvas.

Іншим типом ВМП є мови, підґрунтям яких є правила на основі подій, де програміст визначає правила на зразок «якщо це відбудеться, робити таке». Правило спрацьовує кожного разу, коли умова виконується і запускає спеціальні інструкції. Як приклади таких мов можна навести Blender Game Engine, Construct 2, Kodu, Zapier та ін..

**Висновки.** Таким чином, ми розглянули типізацію візуальних мов програмування за форматом графічного представлення і виокремили дві групи: мови з блоками та блок-схемами. В останній групі виділили такі підтипи: що використовують потік даних, кінцеві автомати, поведінку дерев та правила на основі подій. Проаналізувавши основні типи візуальних мов програмування ми дійшли висновку, що ознайомлення майбутніх інженерів-програмістів із концепціями і технологіями візуального програмування потрібно починати з таких мов як Blokly і завершувати вивчення дисциплін ознайомленням із LabVIEW або подібними. У подальших дослідженнях планується більш детальний огляд обраних для навчання мов.

### *Література*

1. Dehouck R. The maturity of visual programming / R. Dehouck [Електронний ресурс]. – Режим доступу: <http://www.craft.ai/blog/the-maturity-of-visual-programming>.
2. Category:Visual programming languages [Електронний ресурс]. – Режим доступу: [https://en.wikipedia.org/wiki/Category:Visual\\_programming\\_languages](https://en.wikipedia.org/wiki/Category:Visual_programming_languages).
3. Гасько Р.Т. Візуальна навчальна веб-орієнтована мова програмування [Електронний ресурс]. – Режим доступу: <http://ena.lp.edu.ua:8080/bitstream/ntb/15885/1/152-Hasko-319-320.pdf>.
4. Величко В.Є. Використання технології візуального програмування в університетській освіті засобами вільного програмного забезпечення / В.Є. Величко // Вісник Житомирського державного університету: Педагогічні науки. – Випуск 4 (76). – С. 51-55.
5. Толстова Н.С. Системология языков и методологий программирования / Н.С. Толстова [Електронний ресурс]. – Режим доступу: [http://elar.rsvpu.ru/bitstream/123456789/5407/1/poisk\\_2003\\_3\\_2\\_09.pdf](http://elar.rsvpu.ru/bitstream/123456789/5407/1/poisk_2003_3_2_09.pdf).
6. Добро пожаловать в LabVIEW! // Промышленные измерения контроль автоматизация диагностика. – 1-2/2003. – С.50-58. [Електронний ресурс]. – Режим доступу: [http://www.picad.com.ua/1203/pdf/50\\_58.pdf](http://www.picad.com.ua/1203/pdf/50_58.pdf).
7. Dehouck R. The maturity of visual programming. [Електронний ресурс]. – Режим доступу: URL: <http://www.craft.ai/blog/the-maturity-of-visual-programming>.

**Анотація.** Авторамим здійснено огляд візуальних мов програмування. На основі аналізу їх властивостей візуальні мови програмування класифіковано на дві великі групи відповідно до формату графічного представлення синтаксису мов: мови з блоками та мови з блок-схемами. Візуальні мови програмування, що використовують блок-схеми класифіковано на чотири підтипи: що програмують потік даних, кінцеві автомати, поведінку дерев та правила на основі подій.

**Ключові слова:** візуальні мови програмування, Blokly, LabVIEW, професійна підготовка інженера-програміста