

8. Рамський, Ю. С. Основи нечіткої логіки – важливий компонент фахової підготовки майбутніх вчителів інформатики / Ю. С. Рамський, І. А. Твердохліб // Науковий часопис НПУ ім. М. П. Драгоманова. Серія 2 : Комп'ютерно-орієнтовані системи навчання: зб. наук. пр. – Київ : Вид-во НПУ ім. М. П. Драгоманова, 2016. – Вип. 18 (25). – С. 6 – 12.

9. Семеріков С.О. Фундаменталізація інформатичної освіти // Науковий часопис НПУ імені М.П. Драгоманова. Серія № 2. Комп'ютерно-орієнтовані системи навчання: Зб. наук. праць / Редрада. – К.: НПУ імені М.П. Драгоманова, 2009. – № 7(14). – С. 40 – 49.

10. Твердохліб І.А. Методичні аспекти вивчення багатозначних логік в курсі логічних основ інформатики / І.А. Твердохліб // Вісник Чернігівського національного педагогічного університету імені Т.Г. Шевченка [Текст]. Вип. 113 / Чернігівський національний педагогічний університет імені Т.Г. Шевченка; гол. ред. Носко М.О. – Чернігів: ЧНПУ, 2013. (Серія: педагогічні науки) – С. 150 – 157.

11. Твердохліб І.А. Навчання логічних основ інформатики студентів фізико-математичних спеціальностей педагогічних університетів [Текст] : дис. ... канд. пед. наук : 13.00.02 / І. А. Твердохліб ; наук. керівник Ю. С. Рамський ; М-во освіти і науки України, Нац пед. ун-т ім. М. П. Драгоманова. – Київ, 2014. – 236 с.

12. Твердохліб І.А. Навчання мінімізації булевих функцій як невід'ємна складова вивчення логічних основ інформатики / І.А. Твердохліб // Науковий часопис НПУ імені М.П. Драгоманова. Серія № 2. Комп'ютерно-орієнтовані системи навчання: Зб. наук. праць / Редрада. – К.: НПУ імені М.П. Драгоманова, 2014. – № 14 (21). – С. 59 – 65.

13. Глумачний словник з інформатики / Г.Г. Півняк, Б.С. Бусигін, М.М. Дівізінюк та ін.. – Дніпропетровськ, Національний гірничий університет., 2010. – 600 с.

14. Хаптахаева Н.Б. Введение в теорию нечетких множеств: Учебное пособие. – Часть 1 / Н.Б. Хаптахаева, С.В. Дамбаева, Н.Н. Аюшеева. – Улан-Удэ: Изд-во ВСГТУ, 2004. – 68 с.: ил.

15. Fuzzy Logic – Algorithms, Techniques and Implementations, Edited by Elmer P. Dadios. – Rijeka: InTech, 2012. – p. 293.

Некоторые аспекты изучения математической логики студентов информационных специальностей

Рамский Ю.С., Твердохлеб И.А.

Аннотация: в статье рассматриваются проблемы обновления содержания дисциплины «Математическая логика» путем включения таких тем, как многозначные логики, нечеткие логики, минимизация булевых функций; описаны методические особенности изучения указанных тем.

Ключевые слова: логика, математическая, студент, информатика.

SOME ASPECTS OF TEACHING MATHEMATICAL LOGIC OF STUDENTS OF INFORMATICS SPECIALTIES

Yuri Ramsky, Igor Tverdokhlib

Annotation: The article describes the importance of content updating of the course "Mathematical Logic" by including topics such as multi-valued logic, fuzzy logic, minimization of Boolean functions; describes the methodological features of teaching these topics.

Keywords: logic, mathematics, student, informatics.

УДК 373:1.37

Вельбицький І. В.¹, Дем'яненко В. Б.²

¹президент Фонду В.М. Глушкова.

²Національний центр «Мала академія наук України», Київської Малої академії наук України.

Програмування без мов програмування.

Графічна поліглот-концепція програмування

Анотація. У статті описано графічну поліглот-концепцію програмування, її переваги і перспективи використання. Для запису програм в графічному середовищі програмування використовуються математично строго визначені логічні і абстрактні схеми, для побудови яких не вимагається синтаксис певних мов програмування.

Ключові слова: графічне середовище програмування, графи, логічні і абстрактні R-схеми програм, графічна поліглот-концепція програмування.

Нині досвід розбудови розвинених країн переконує, що найбільш перспективною парадигмою розвитку суспільства виступає глобальна модель «суми високих технологій», що базується на сучасній світовій інформаційно-комунікаційній структурі, яка розвивається швидкими темпами. Передові країни світу розбудовують новий технологічний уклад, за якого забезпечуються інтенсивні взаємозв'язки та взаємозбагачення різних технологічних напрямів (мікроелектроніка, нанотехнології, інформатика, біотехнологія та інші). Розвиток індустрії інформаційних технологій та програмного забезпечення визначально впливає на розвиток галузей економіки, вона є високотехнологічною й високорентабельною сферою. І особливість полягає в тому, що в разі їх використання не

потребується великої кількості природних ресурсів, вони є екологічно чистими, їх використання сприяє реалізації високого наукового, інтелектуального і технологічного потенціалу суспільства. Важливою складовою інформаційних технологій є програмні засоби. Розробка програмних засобів та послуг щодо їх впровадження в розвинених країнах – стала самодостатньою індустрією. В Україні ж одним із завдань є перехід від мануфактурного до індустріального виробництва програмного забезпечення [6]. Готувати фахових програмістів слід починати з шкільних років. На початку становлення інформатики як навчальної дисципліни панувала парадигма навчання програмування (процедурного), пов'язана з розумінням інформатики як науки про алгоритмізацію, програмування та розв'язування задач за допомогою комп'ютера. Для розв'язування поставлених задач на тому етапі в більшості випадків достатньо було мати поняття про алгоритм та елементарні знання з алгебри висловлень. З набуттям практичного досвіду навчання предмету, з розширенням комп'ютерного парку та урізноманітненням програмного забезпечення змінювалися акценти в навчанні інформатики. Головним завданням навчання стала підготовка користувача програмних засобів. Слід зазначити, що за реалізації переходу до парадигми користувачького ухилу в навчанні інформатики проявилися деякі негативні наслідки, зокрема пов'язані з послабленням уваги до базової, фундаментальної підготовки з інформатики (стосується як середньої так і вищої школи) та навчання програмування, за яких основною метою навчання є засвоєння наукових основ, загальних методів опрацювання різноманітних повідомлень і даних за допомогою засобів сучасних інформаційно-комунікаційних технологій, а не просто елементарне оволодіння конкретними способами і прийомами роботи з певними програмними засобами [5]. Не можна допустити, щоб роль науки інформатики почала зводитись до допоміжної, суто утилітарної науки, як джерела постачання засобів діяльності для формування знань учнів в процесі вивчення різних дисциплін. Суспільству потрібні як грамотні користувачі програмного забезпечення, так і кваліфіковані програмісти. І саме в шкільні роки закладаються основи такої грамотності.

Ефективність засвоєння знань учнями за умов широкого впровадження засобів нових інформаційних технологій навчання під час вивчення програмування в значній мірі залежить від педагогічних програмних засобів, використання яких дозволяє поєднати високі обчислювальні можливості в процесі дослідження різноманітних об'єктів з унаочненням результатів на всіх етапах розв'язування задач [4]. Головна проблема шкільного навчання програмування полягає у відсутності системного підходу, в школі вчать не логіки розв'язування задач та виконання завдань за допомогою програмування, не розроблення програм, а лише мови програмування як такої. Часто шкільні уроки зводяться до вивчення конструкцій мови і виконання будь-яких завдань на ці конструкції. На уроках програмування діти мають вчитися працювати з даними, структурувати їх, ці навички життєво необхідні в умовах все зростаючого «інформаційного буму» сучасного життя. «Вчити логічних основ програмування, а не тільки синтаксису певної мови програмування» – має бути головним принципом навчання.

Термін «технологія програмування» було введено у 1960-х роках минулого століття як інтуїтивне усвідомлення того факту, що головне не тільки мова (в той час це була мова Алгол-60), а й те, як її використовувати і грамотно описувати за її допомогою процеси розв'язування задач [1, 3]. На тепер цей термін включений в навчальні програми як середньої, так і вищої освіти.

Програмування в графах почало формуватися в 70-х роках минулого століття в процесі розроблення великих систем управління в реальному режимі часу [3]. З'явилось усвідомлення важливості формального документування процесу їх розроблення для полегшення швидкого внесення постійних виправлень і удосконалень програм. Е. Дейкстра [8] вперше показав математичну не строгість і надлишковість традиційного програмування.

Нині, узагальнюючи багаторічний досвід розробок різних автоматизованих технологічних комплексів, було встановлено, що проблеми технологій програмування необхідно розв'язувати шляхом зміни вихідних концепцій програмування. В результаті було запропоновано графічну концепцію програмування на основі логічних R-схем, цікавою своїми перевагами: простотою, наочністю і можливістю використання всіма бажаними, а не тільки програмістами [2, 9].

Метою написання статті є розкриття сутності графічної поліглот-концепції програмування та перспектив її використання.

У графічній концепції програмування пропонується не записувати (певною мовою), а креслити (рисувати) програми. Людина завжди прагнула знайти графічний образ будь-яких своїх дій. Відомо значна кількість графічних способів запису програм у вигляді графів: Flow-chart, UML, Workflow, SDL, ДРАКОН, Google BLOCKLY та ін. Запропоновані R-схеми (див. Рис. 1) – це графи, навантажені на дуги (ребра). Вони виконуються на всьому «життєвому» циклі виконання програм, більш прості, наочні і компактні. Разом з тим передбачено включення доказового стилю роботи. З огляду на незвичайну (на 1-2 порядки вищу) компактність запису програм, наочність і простоту швидкого введення даних, особливо через використання сенсорних екранів, нова концепція має значні перспективи ефективного застосування.

Графічне середовище описане мовою C ++, як плагін REditor до Qt Creator. Основну, центральну частину монітора займає робоче поле, на якому здійснюється розроблення R-схем. Зверху, в трьох рядках робочого поля розташовуються: меню формування архітектури середовища, панель інструментів (з 14 графічних іконок) і панель відкритих (необмежене число) імен робочих

полів R-схем. Зліва розташована область для зображення дерева R-схем певного проекту. Графічне середовище реалізоване за принципом А. Ейнштейна: «...будь-яку роботу роби так просто, як це можливо, але не простіше цього».

Графічна концепція програмування є поліглотом (зрозумілою для всіх незалежно від природних мов) і наслідує строгі математичні принципи подання та опрацювання даних, що наближається до принципів роботи мозку людини.

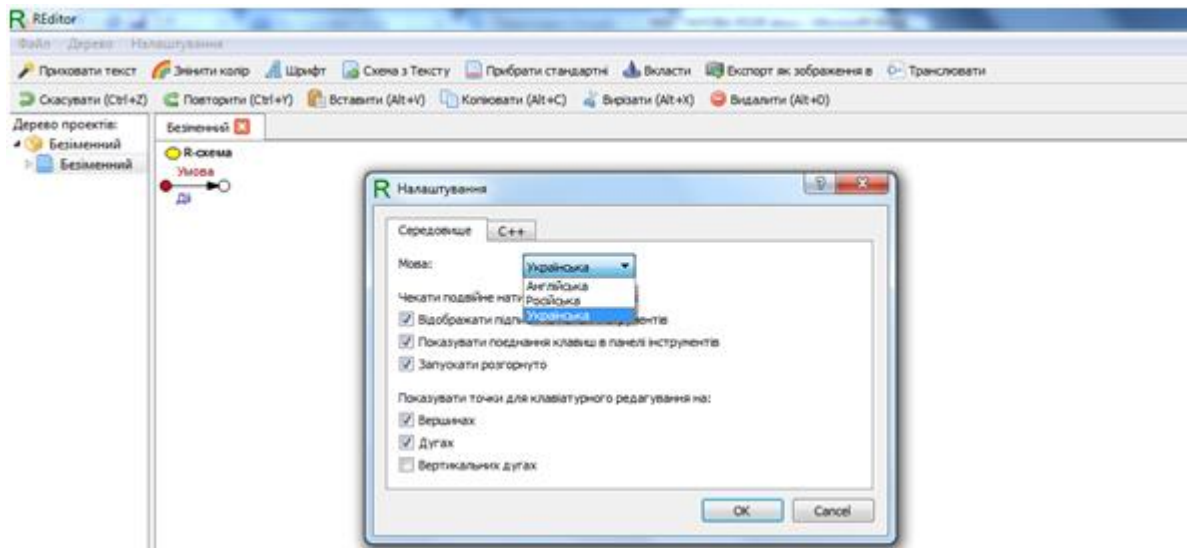


Рис. 1. Графічне середовище програмування логічними R-схемами.

В результаті – не потрібна спеціальна мова програмування, а графічний образ (креслення) проекту алгоритму збігається з кінцевим виробом – програмою, її документацією та мережевим графіком розробки. Це стає зрозумілим і для розробників-програмістів і для користувачів. Принципи такої концепції прості, природні для людини, програмування стає доступним для всіх, стає елементом загальної інформатичної грамотності й інформаційної культури людини в суспільстві. При цьому, всі традиційні машинно-орієнтовані оператори типу: if-then, else, for, while, goto ..., мітки, блокові дужки типу begin-end, {-} ..., відступи, переведення рядків, більшість знаків пунктуації тощо виключені, вони складні й емпіричні (строго не визначені). Для нейтралізації недоліків роботи з ними людина витрачає занадто багато зусиль, створюючи величезну кількість спеціальних мов, методів і середовищ програмування, що «нібито спрощує», описування програм, а насправді програмування стає ще більш складним. У графічній концепції програмування виключаються значна кількість символів: імен операторів, блокових дужок, розділових знаків, розриву рядків, відступів тощо. Це забезпечує компактність створення програм. Введення одного графічного оператора (дуги) здійснюється за допомогою натискання однієї клавіші мишки або клавіатури, що реалізується в кілька разів швидше, ніж введення існуючих операторів традиційних мов програмування.

У поліглот-концепції програмування для запису будь-яких алгоритмів, даних і програм використовується одна горизонтальна дуга графа, логічна R-схема (див. Рис.2.), на якій зверху записуються «умови» поставлених задач, а знизу – «дії», які виконуються, якщо умова є істинною. З вершини графа може виходити будь-яке число дуг вліво і/або вправо. Дуги розглядаються послідовно, виконуються всі дії і здійснюється перехід за стрілкою дуги в новий стан графа (алгоритму, програми). Якщо на всіх дугах, що виходять з вершини, вказані там умови, хибні, то здійснюється перехід за стрілкою останньої дуги у відповідний новий стан графа (програми) без виконання дій.



Рис. 2. Логічна R-схема.

Для запису «умов» та «дій» в один або кілька рядків можуть використовуватися будь-які мови як природні, так і спеціальні (без обмежень): англійська, російська, китайська, мова математики, будь-яка мова програмування та ін. (див. Рис. 3.). У графічному середовищі програмування всі ці мови перетворюються в мову математики і далі – в мову комп'ютера.

На Рис. 4 наведено приклад визначення нового оператора: «3-х (може бути будь-яке число) Умов та Циклу». Наведена його логічна R-схема (а), абстрактна R*-схема (б) або запис R-схеми (алгоритму, програми) без деталей реалізації, без записів на дугах, тобто максимально компактно із зображенням тільки математичної логіки (сутності) програми та еквівалентний запис мовою C ++, де червоним кольором позначені зайві символи для R-схем (с).

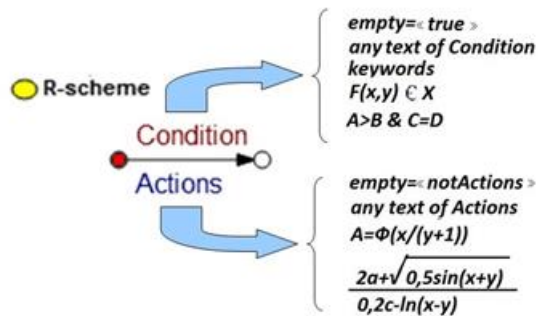


Рис. 3. Запис «умов» і «дій» на дузі графа.

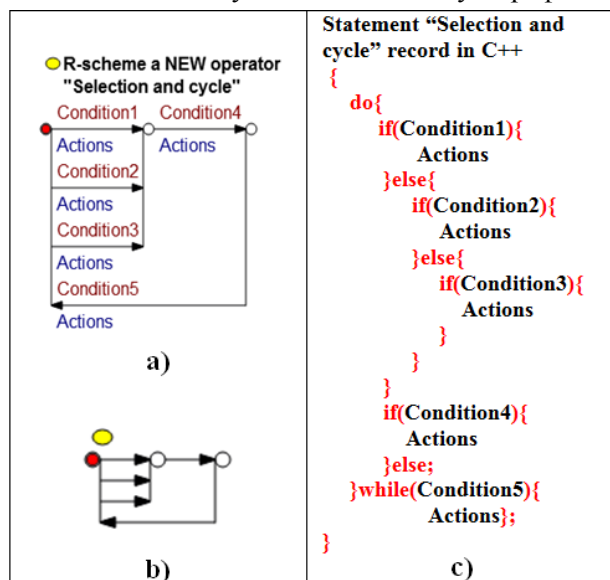


Рис. 4. Запис нового оператора «Добір та цикл» в графічному середовищі та мовою C++.

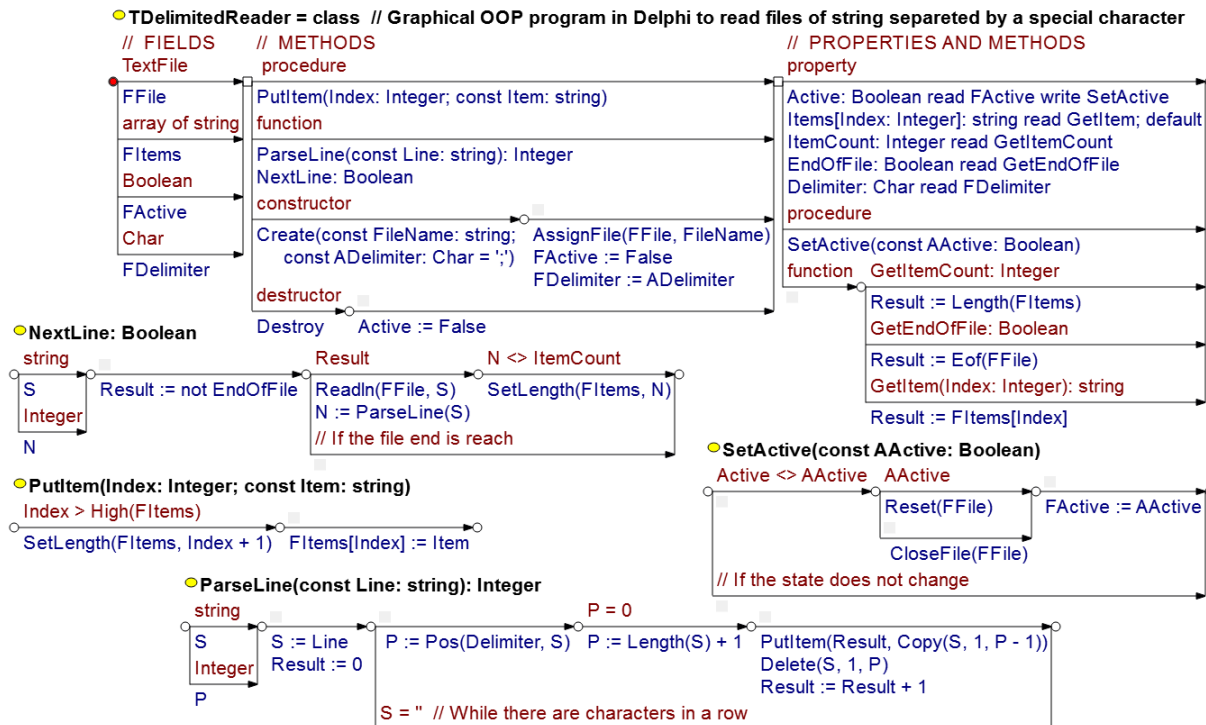


Рис. 5. Логічна R-схема об'єктно орієнтованої програми, поданої мовою Delphi.

Графові надається ім'я, яке записується зверху біля жовтого еліпса. У математиці це ім'я відповідає імені деякої функції, заданої графічно в запропонованому середовищі програмування. Ім'я може бути з параметрами або без них [7]. Програма в графічному середовищі програмування задається будь-яким числом взаємозалежних за іменем таких графів. Перший граф (R-схема) такої програми завжди є аксіомою, у якій перша вершина завжди відзначена червоним кружечком. На Рис. 5 наведено графічне подання об'єктно орієнтованої програми мовою Delphi [10], а на Рис.7 – мовою

C++ [7]. Це означає, що в цих програмах «умови» та «дії» записані мовами Delphi і C++ відповідно. За першою логічною R-схемою задається формула визначення об'єктно орієнтованої програми: задання даних (private для C++) і пов'язаних з ними методів (public для C++). Далі наводиться визначення чотирьох методів і (5 + 1) public-функцій для C++ відповідно.

На Рис. 6 наведено абстрактну R*-схему програми, поданої на Рис. 5. Логічна R- і абстрактна R*-схеми є єдиною графічною оболонкою (еталоном) для запису програм і бібліотек програм всіма мовами.

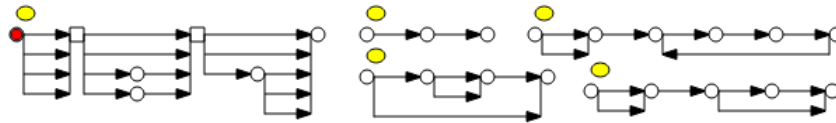


Рис. 6. Абстрактна R*-схема програми (без деталей реалізації).

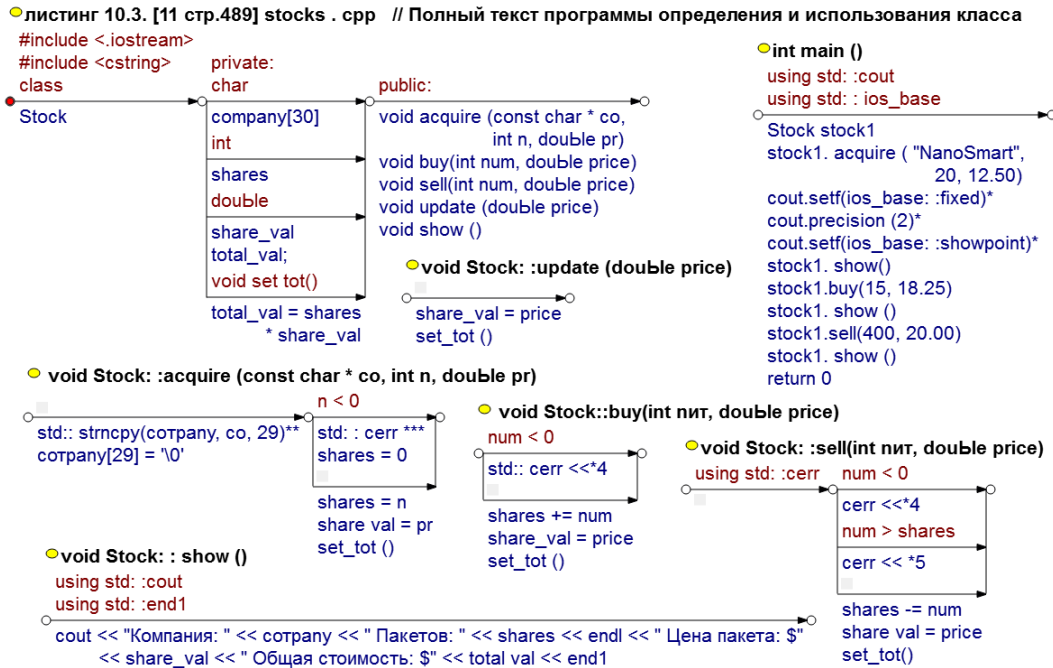


Рис. 7. Логічна R-схема об'єктно орієнтованої програми, подана мовою C++.

Над дугами можуть використовуватися ключові слова типу int, array, class, function, #include <iostream> та ін. (див. Рис. 5, 7), які завжди істинні й задають спеціальне використання всіх елементів на дугах R-схем. Такі дуги зазвичай безальтернативні, незалежні одна від одної і «умови» на цих дугах можуть виконуватися паралельно. Дані, що в них записані є допоміжними і служать для усунення недоліків концепції традиційного програмування та спрощення трансляції. Надалі такі записи можуть бути виключені з графічного середовища програмування частково або повністю.

Вершини графа можуть мати різну конфігурацію і колір для реалізації в програмі: визначення дуг, позначення маршрутів автоматичної генерації текстів програм та ін. Наприклад, червона вершина завжди визначає початок програми. Дуги можуть бути подвійними, хвилястими, пунктирними тощо.

На Рис. 5, 6 квадратною вершиною позначені дуги, умови яких можуть виконуватися паралельно, а на Рис. 8 – вершиною у вигляді паралелограма зображено багатовимірне виконання графічних програм. Це дозволяє значно спростити програмування тривимірних об'єктів, зробити їх більш наочними і зрозумілими, значно підвищити рівень автоматизації і компактності їх подання для опрацювання за допомогою комп'ютера.

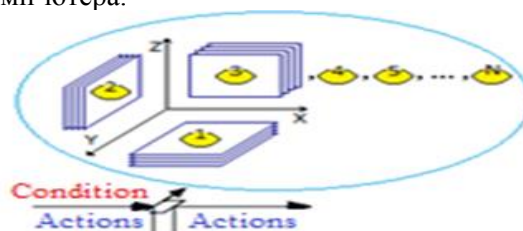


Рис. 8. Зображення багатовимірного виконання графічних програм.

На всьому проектному циклі програмування різноманітних задач забезпечується можливість програмувати різні частини проекту одночасно, генерувати функціонально-повний набір текстів і проводити налагодження програм. Разом з тим сам процес програмування стає простим, прозорим, зі збереженням і накопиченням досвіду для розробників і користувачів. В результаті програма з «речі в собі», не завжди доступної і зрозумілої тільки автору, стає прозорою для розуміння і розвитку іншими користувачами (див. Рис. 9, 10, 11).

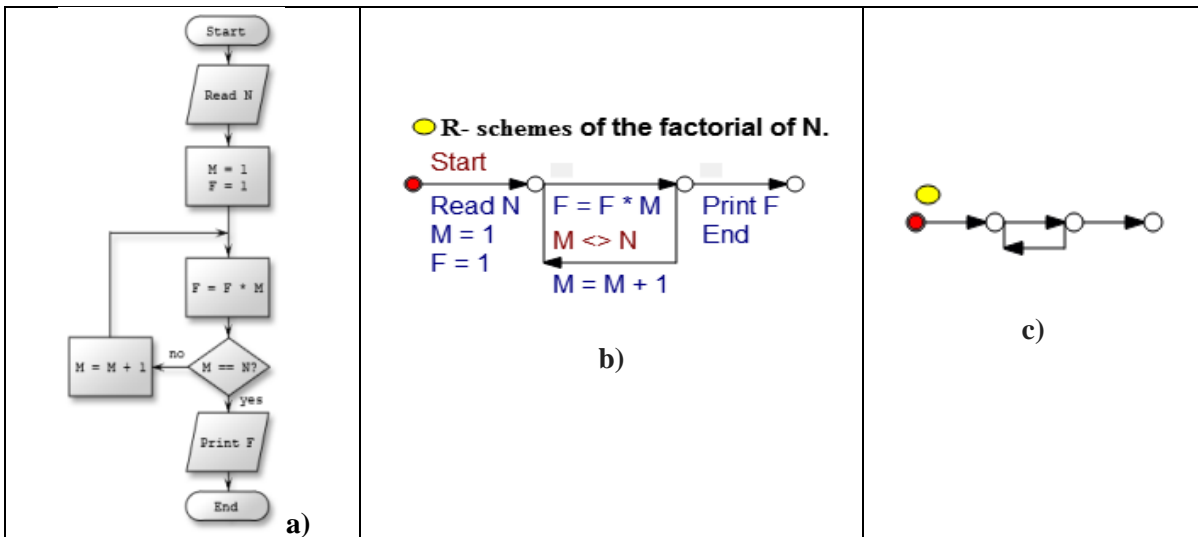


Рис. 9. Приклад запису алгоритму обчислення факторіала числа N в блок-схемах (а) та в логічній R-схемах (b) і абстрактній R*-схемах (c).

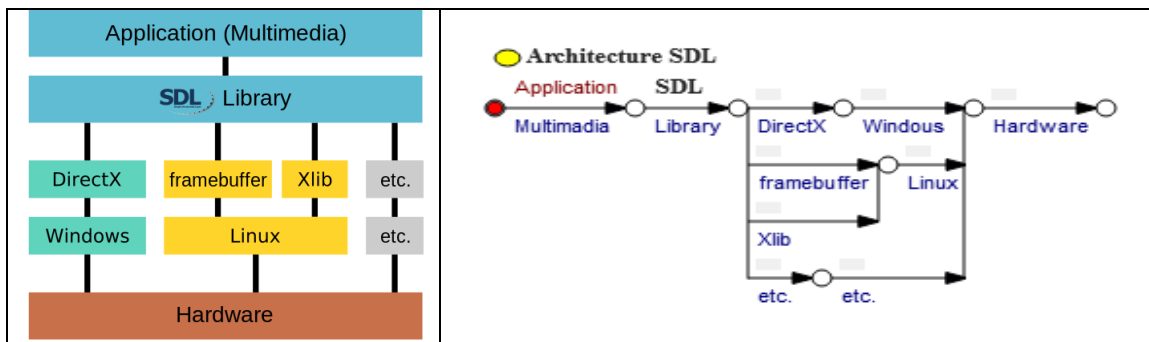


Рис. 10. Логічна R-схема Workflow архітектури SDL.

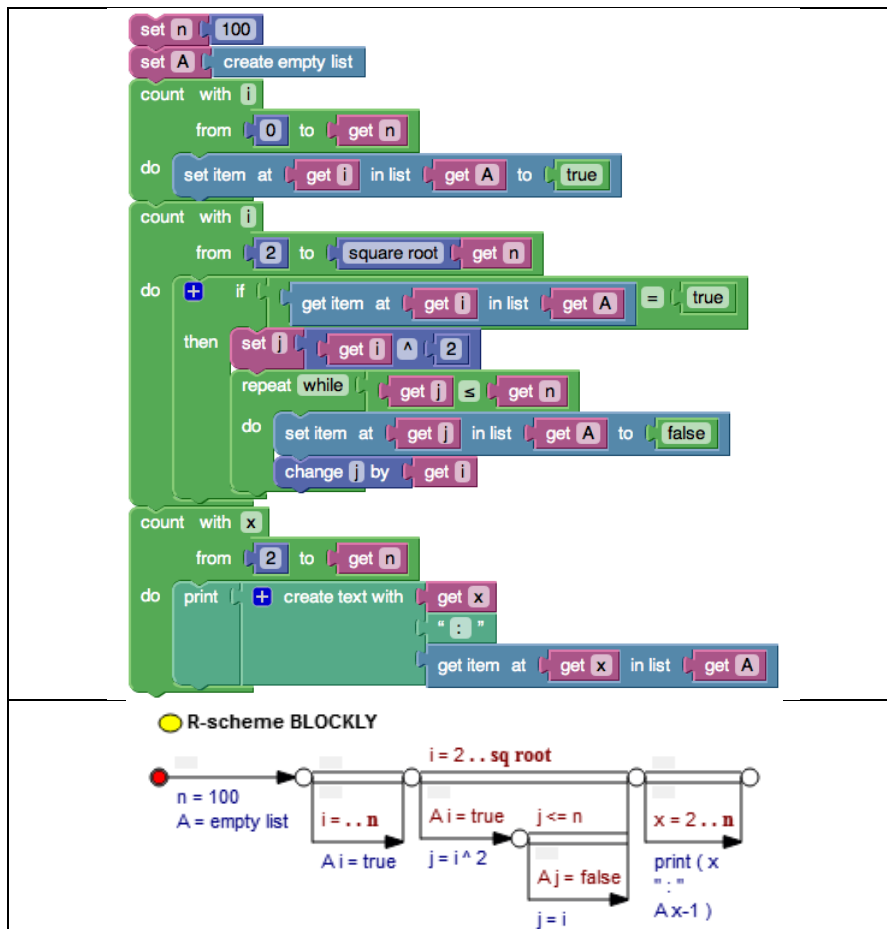


Рис. 11. Приклад опису програми візуальною мовою програмування Google Blockly та логічною R-схемою.

Висновки. Традиційна концепція проектування алгоритму (програми) заснована на побудові обчислювальної схеми розв'язування вихідного завдання за допомогою десятків фіксованих, текстових, машинно-орієнтованих операторів. Це складний, багатоступеневий процес. Для спрощення цього процесу застосовуються численні мови і методи програмування, що ще більше ускладнює процес програмування. Нова графічна поліглот-концепція програмування заснована на побудові логічної схеми вихідної задачі в термінах постановки задачі. Запропоновану концепцію може бути включено в систему освіти на будь-яких етапах підготовки будь-яких фахівців. Вчителям і викладачам не нав'язується ніяка методика подання навчального матеріалу, закріплення і контролю знань, конкретний зміст, методи, засоби й організаційні форми навчання, співвідношення між самостійною роботою учнів і роботою разом із вчителем, між індивідуальними і колективними формами роботи та ін. Все це педагог повинен визначити сам з врахуванням своїх власних позицій і уподобань, специфіки умов, в яких перебігає навчальний процес, індивідуальних особливостей окремих учнів і класного колективу. Програмувати зможуть всі, програмування стане елементом загальної інформатичної грамотності й інформаційної культури кожного громадянина суспільства.

Список використаних джерел

1. Вельбицкий И. В. Технология программирования. Техника. Украина, 1984. – 279 с.
2. Вельбицкий И. В., Дем'яненко В. Б. Графічне середовище програмування як засіб формування інформатичної грамотності // Інноваційні технології навчання обдарованої молоді: матеріали VII-ї Міжнародної науково-практичної конференції, 7-8 грудня 2016 року, м. Київ. К. : Інститут обдарованої дитини, 2016. С. 12-16.
3. Глушков В. М. Технология программирования и проблемы ее автоматизации // УСИМ. № 6, 1976. С. 75-93.
4. Дем'яненко В. Б. Информатика в системі фундаменталізації навчання учнів Малої академії наук України // Науковий часопис НПУ імені М. П. Драгоманова. Серія № 2. Комп'ютерно-орієнтовані системи навчання : Зб. наук. праць / Редрада. К. : НПУ імені М. П. Драгоманова, 2014. № 14 (21). С. 70-77.
5. Рамський Ю. С. Методична система формування інформаційної культури майбутніх вчителів математики: дис. ... докт. пед. наук : 13.00.02 – теорія та методика навчання (інформатика); Національний педагогічний університет ім. М. П. Драгоманова ; Київ : [б. в.], 2013. 560 с.
6. Рекомендації парламентських слухань на тему «Створення в Україні сприятливих умов для розвитку індустрії програмного забезпечення» [б. в.]. Київ, 14 грудня 2011. 11 с.
7. Стивен Прата, Язык программирования C++, SAMS 46240 USA. 1184 с. Пер. с англ. М. 2007.
8. Dijkstra E. Letters to the editor: go to statement considered harmful . Communications of the ACM. 1968. P. 147-148.
9. Velbitskiy I. Programming without Programming Languages. 6th International Conference on IT Convergence and Security (ICITCS September 26-29th 2016). Produced by IEEE Expres Conference Publishing, NY 12571 USA, 2016. p. 139-144.

Программирование без языков программирования. Графическая Полиглот-концепция Программирование

Вельбицкий И. В., Демьяненко В. Б.

Аннотация. В статье описано графическую полиглот-концепцию программирования, ее преимущества и перспективы использования. Для записи программ в графической среде программирования используются математически строго определенные логические и абстрактные схемы при построении которых не требуется синтаксис определенных языков программирования.

Ключевые слова: графическая среда программирования, графы, логические и абстрактные R-схемы программ, полиглот-концепция программирования.

Programming without Programming Languages. Graphic-concept Polyglot Programming *Velbitskiy I. V., Demianenko V.*

Resume. This article describes graphic-polyglot programming concept, its advantages and prospects. Using mathematically defined strictly logical and abstract schemes of the construction which does not require specific programming language syntax to record programs in graphical programming environment.

Keywords: graphical programming environment, graphs, logical and abstract programs R-schemes, polyglot concept of programming.

УДК 378.001.89

Спірін О. М., Носенко Ю. Г., Яцишин А. В.

Інститут інформаційних технологій і засобів навчання НАПН України

Підготовка наукових кадрів вищої кваліфікації з інформаційно-комунікаційних технологій в освіті

Анотація. У статті представлено досвід Інституту інформаційних технологій і засобів навчання НАПН України в контексті підготовки аспірантів – майбутніх докторів філософії. Наведено основні