http://www.moderntechno.de/index.php/meit/article/view/ge4-020
DOI: 10.30890/2567-5273.2018-04-01-020

УДК 004.42

ANALYSIS OF PROGRAMMING LANGUAGES
АНАЛІЗ ЗАСОБІВ ПРОГРАМУВАННЯ

Andriiv.I.V. / Андріїв І.В.
Ivano-Frankivsk National Technical University of Oil and Gas,
Ukraine, 76019, Ivano-Frankivsk, st. Karpatska, 15,
Івано-Франківський Національний Технічний університет нафти і газу,
Україна, 76019, Івано-Франківськ, вул. Карпатська, 15,

Анотація. *Актуальність дослідження пояснюється важливістю отримання якісних прогнозів основних фінансових індикаторів в умовах сучасних тенденцій, а також успішним досвідом застосування нейронних мереж в задачах прогнозування, що може бути здійснено вірним вибором мови програмування. Це актуалізує мета даної статті - проведення порівняльного аналізу мов програмування.*

Ключові слова: *мова програмування, середовище програмування, інструменти.*

Існуючі мови програмування призначені для вирішення орієнтованого досить вузького кола завдань. Мова не може бути панацеєю, його хороші якості для одних завдань (або людей) можуть виявитися поганими для інших. Слід відрізняти мову програмування (Basic, Pascal) від його реалізації, яка зазвичай представлена в складі середовища програмування (Quick Basic, Virtual Pascal) - набору засобів для редагування вихідних текстів, генерації виконуваного коду, відладки, управління проектами і т.д. Синтаксис і семантика мови програмування фіксується в стандарті мови. Кожна середовище програмування надає свій інтерпретатор або компілятор з цієї мови, який часто допускає використання конструкцій, які не фіксованих в стандарті. Істотною є мета, для реалізації якої обирається мова - або для навчання програмуванню, або для вирішення конкретної прикладної задачі. У першому випадку мова повинна бути простою для розуміння, суворим і по можливості позбавленим "підводних каменів". У другому - нехай складним, але ефективним і виразним інструментом для професіонала. По набору понять мови поділяються на високо- і низькорівневі.

Перші надають високий рівень абстракції від устаткування, другі - низький, наближений до машинного. З точки зору того, чи внесені в набір понять особливі, специфічні для предметної області об'єкти, мови діляться на універсальні (процедурні) і спеціалізовані. До останніх можна віднести Prolog, Lisp. Універсальні мови дозволяють реалізувати будь-який алгоритм, користуючись стандартним набором конструкцій. Завдяки цьому, код такою мовою може бути досить легко перенесений з одного процедурного мови в інший за допомогою консервативних змін. Відмінності в мовах зводяться до способів визначення процедур і функцій, варіантів передачі параметрів, можливостям визначення рекурсивних процедур і наявності процедурного типу даних.

Наявність і широка класифікація типів пам'яті дає можливість ефективно управляти її розподілом, а й вносить складність, що вимагає від програміста більш уважного ставлення. Зазвичай виділяють (максимальний спектр):

регістри, глобальні, локальні і динамічні змінні. Наявність коштів логічного об'єднання групи процедур, функцій, змінних дозволяє працювати з великими проектами, спрощуючи їх структуру. Важлива властивість - можливість опису процедур ініціації і завершення модуля. Об'єднання структур і методів їх обробки (інкапсуляція) створює значні зручності при програмуванні. Можливість успадкування дозволяє привести в систему набір структур. Автоматично викликаються конструктори і деструктори полегшують процес взаємозв'язків. Все це становить зручний інструмент для опису понять і дій прикладної програми.

Незалежність від апаратури, що реалізується за допомогою семантики, що не залежить від конкретної машини і внесенням в мову ряду специфічних понять - таких, як базовий тип з нефіксованим розміром (int в C) [1]. З точки зору ефективності, важливо як виповнюється програма - послідовної інтерпретацією вихідного тексту (інтерпретатор) або безпосереднім виконанням готового коду (компілятор). Інтерпретатор доцільно використовувати лише в разі, коли швидкість інтерпретації не сильно позначається на ефективності програми. Крім інтерпретації і компіляції можливі проміжні варіанти з генерацією псевдокоду, який відрізняється від від початкового тексту високою швидкістю інтерпретації або іншими корисними властивостями (наприклад, можливістю виконання на машинах різної архітектури - як Java). Найяскравіший представник мов низького рівня Assembler, набір понять якого заснований на апаратній реалізації. Це засіб автоматизації для програмування безпосередньо в кодах процесора. Машинні команди описуються у вигляді мнемонічних операцій, що дозволяє домогтися досить високою модифікованості коду. Оскільки набір команд на різних процесорах різний, то і про сумісність говорити не доводиться. Використання асемблера доцільно у випадках, коли необхідно безпосередньо взаємодіяти з обладнанням, або отримати більшу ефективність для деякої частини програми за рахунок більш високого контролю над генерацією коду. [4] Після тривалої боротьби на фронті програмних середовищ для Windows, Borland пішла на ринок корпоративних систем. Delphi - це не продовжувач справи Borland Pascal / Borland C, його ніша - т.зв. швидке створення додатків (Rapid Application Developing, RAD). Подібні засоби дозволяють в найкоротші терміни створити робочу програму з готових компонентів, не витрачаючи масу зусиль на дрібниці. Особливе місце в таких системах займають можливості роботи з базами даних. [5] Як яскравий приклад спеціалізації, мова Java з'явився у відповідь на потребу в ідеально стерпному мовою, програми на якому ефективно виконуються на стороні клієнта WWW. В зважаючи на специфіку оточення, Java може бути хорошим вибором для системи, побудованої на Internet / Intranet технології. [6] В основі мови C - вимоги системного програміста: повний і ефективний доступ до всіх ресурсів комп'ютера, засоби програмування високого рівня, переносимість програм між різними платформами і операційними системами. C ++, зберігаючи сумісність з C, вносить можливості об'єктно-орієнтованого програмування, висловлюючи ідею класу (об'єкта) як визначається користувачем типу. Завдяки перерахованим

якостям, C / C ++ зайняв позицію універсальної мови для будь-яких завдань. Але його застосування може стати неефективним там, де потрібно отримати готовий до вживання результат в найкоротші терміни, або там, де невигідним стає сам процедурний підхід. [7] Для реалізації проекту побудови нейромережевої моделі для прогнозування часових рядів фінансових даних на базі багатошарового персептрона, навченого за алгоритмом зворотного поширення помилки (а також формалізації повної схеми застосування даної моделі для аналізу і прогнозування часових рядів на прикладі котирувань акцій російських емітентів на ММВБ) обрано середовище розробки C ++ Builder 2010, оскільки вона поєднує в собі функціональність і гарну швидкість роботи програм зроблених на C ++, а також дозволяють в найкоротші терміни з дати робочу програму з готових компонентів, не витрачаючи масу зусиль на дрібниці. Змінюється апаратура та операційні системи. Виникають нові завдання з найрізноманітніших предметних областей. Відходять у минуле і з'являються нові мови. Але залишаються люди - ті, хто пише і ті, для кого пишуть нові програми і чиї вимоги до якості залишаються тими ж незалежно від цих змін.

Література:

1. Джошуа Блох Java. Ефективне програмування Effective Java. Programming Language Guide - Серія: Java Видавництво: Лорі, 2002 г. - С. 224.

2. Лафоре Р. Объектно-ориентированное программирование в C++ Object-Oriented Programming in C++ − Издательство: Питер, 2011 г.

3. Джаррод Холінгворт, Боб Сворт, Марк Кешмен, Поль Густавсон Borland C ++ Builder 6. Керівництво розробника Borland C ++ Builder 6 Developer's Guide. - M .: «Вільямс», 2004. - С. 976.

4. 1. Joshua Bloch Java. Effective Java programming. Programming Language Guide - Series: Java Publisher: Lori, 2002 - P. 224.

5. 2. Lafore P. Object-Oriented Programming in C ++ Object-Oriented Programming in C ++ - Publisher: Piter, 2011

6. 3. Jarrow Holingworth, Bob Swart, Mark Keshman, Paul Gustavson Borland C ++ Builder 6. Developer's Guide Borland C ++ Builder 6 Developer's Guide. - M.: Williams, 2004. - P. 976.

Abstract. *The relevance of the study is explained by the importance of obtaining qualitative forecasts of the main financial indicators in the context of current trends, as well as the successful experience of using neural networks in prediction tasks, which can be done by the right choice of programming language. This updates the purpose of this article - a comparative analysis of programming languages.*

Key words: *programming language, programming environment, tools.*

Стаття відправлена: 09.06.2018р.