

Мови програмування в навчанні майбутніх програмістів

Вибір відповідної мови програмування для навчання студентів програмування представляє реальну проблему протягом багатьох років і є предметом обговорення викладацького співтовариства в усьому світі (див., наприклад, [1]–[13]). Деякі фахівці пропонують навчати програмування, використовуючи одну мову програмування, наприклад, Pascal або C++; інші – пропонують одночасне вивчення двох мов, що дозволяє порівнювати різні концепції й парадигми, закладені в ці мови. Запропоновані різні педагогічні підходи [1], які використовуються при навчанні програмування студентів першого, другого років навчання. Найбільш відомі підходи (їх шість) відображені у відомому освітньому стандарті [11], [12]: імперативний, об'єктний, функціональний, широкий, алгоритмічний і апаратний.

Як навчати програмування, які парадигми, середовища, мови програмування для цього використовувати – ці питання продовжують хвилювати й цікавити викладачів відповідних кафедр і відділень університетів.

Огляд сучасних досліджень проблем навчання програмування

У науковому дослідженні співробітників університету Карнегі Меллона (Carnegie Mellon University) К. Келлехер і Р. Пош на тему «Як усунути перешкоди для оволодіння програмуванням: таксономія середовищ і мов програмування для новачків» [2] наведена таксономія мов і середовищ, спроектованих таким чином, щоб зробити мову програмування легко освоюваною для новачків-програмістів незалежно від їх віку. При створенні середовища програмування в такій ситуації головне питання – для чого новачкові потрібні програми, де й навіщо він буде згодом використовувати надбані їм навички програмування. Існують різні можливі мотивації для вивчення програмування, наприклад, – програмування, необхідне для кар'єри; необхідне для розв'язування завдань певними структурними й логічними способами; необхідне при написанні прикладних програм для власних потреб або в якій-небудь предметній галузі, і т.п. Системи, представлені в таксономії [2], поділені на дві великі групи: системи, в яких передбачається навчання програмування для власної користі, і системи, де передбачається використання програмування для інших цілей, – наприклад, навчання студентів когнітивного моделювання в психології.

У статті [3] проведено порівняння мов програмування, використовуваних для навчання програмування студентів на молодших курсах. Виконаний авторами дослідження аналіз показав, що 34% опитаних викладачів при виборі мови програмування враховують попит з боку студентів, популярність мови; 20% опитаних вказали, що віддають перевагу певній мові програмування через її педагогічні переваги; 17% викладачів враховують специфіку спеціальності й напряму підготовки; 16% – вибирають об'єктно-орієнтовані мови; 6% – враховують привабливість інтерфейсу; 5% – доступність і ціну програмного забезпечення; 2% опитаних вибирають ту мову програмування, для якої легко знайти відповідні підручники, посібники. Крім перерахованих вище критеріїв, існують і інші. Сьогодні в університетах обирають ту або іншу мову програмування, опираючись на локальні педагогічні традиції й можливість їх реалізації за допомогою мови програмування. У статті [3] є список вимог для вибору мови програмування в школах: (1) простота у використанні; (2) структура дизайну; (3) потужні обчислювальні засоби; (4) простий синтаксис; (5) оголошення зінних; (6) простота введення/виведення й форматування виведення; (7) значимі назви ключових слів; (8) можливість добирати виразні імена змінних; (9) забезпечення структури one-entry/one-exit; (10) швидкий зворотний зв'язок; (11) наявність гарного діагностичного інструментарію для тестування й налагодження. Багато із критеріїв, перерахованих вище, обговорювалися у відомій статті Л. Макивера й Д. Конвея «Сім смертних гріхів проектування ввідних мов програмування» [4]. Автори роботи [4] дають список причин, що пояснюють, чому мови програмування, які звичайно використовуються на молодших курсах, роблять важким навчання програмування. Вони також пропонують певні принципи проектування мови програмування, призначення яких допомогти розробці мови програмування, прийнятної з педагогічної точки зору.

У роботах [5], [6], [7], [8] досліджуються проблеми й вигоди використання різних підходів до подання початкового курсу програмування, а також мов програмування. У статті [5] Атанасом Раденських запропонований підхід "Python First", заснований на використанні мови Python у вступному курсі програмування. Успішна реалізація цього підходу опирається на онлайн-пакет навчання «Study Pack». Брайан Байтс в [6] пропонував використовувати в якості вступної мови програмування C#. Б. Байтс зіставив мови програмування C++ і C#. Зауважуючи, що C++ доступний на багатьох платформах, а C# – поки тільки для Windows-машин, що функціонують під управлінням .NET Framework, Б. Байтс вважає, що студентів необхідно навчати на молодших курсах мови C#, оскільки ця мова програмування використовується часто в реальному світі.

У роботах [7] і [8] було проведено зіставлення мов програмування Java і C#. Зокрема, на базі цього зіставлення була запропонована для використання мова Java у багатьох навчальних дисциплінах, у яких необхідне використання мови програмування для виконання навчальних проектів.

У статті великої міжнародної групи авторів, викладачів університетів Швеції, Фінляндії, Великобританії, США й Данії [9], наведені результати опитування, яке проведене цією міжнародною групою серед університетських викладачів вступного курсу програмування, про літературу, у якій відображені проблеми навчання програмування в перший і другий рік навчання. Автори вважають, що протягом десятиліть дослідження з стосовно вступного курсу програмування не створювали достатнього впливу на аудиторну практику, оскільки ці дослідження й педагогічні,

когнітивні дослідження йшли незалежно одне від одного, й в них не враховувалися результати один одного. Автори огляду [9] опитали 100 відомих у світі експертів, а також більше тисячі викладачів, які знаходяться в електронній підписці групи ACM за спеціальними інтересами у питаннях освіти стосовно комп'ютерних наук SIGCSE (ACM Special Interest Group on Computer Science Education). У результаті такого опитування в список для розгляду було включено більше 180 статей. Статті були згруповані за наступною тематикою: навчальні програми й плани, педагогічні питання, мови програмування, інструментарій. Для відбору статей використовувалася модифікована версія Лісабонської таксономії [10]. У результаті обговорень було відібрано 45 статей, рекомендованих для вивчення й використання викладачами університетів світу. Повний список цих статей з анотаціями й класифікаціями наведений в [9, 216–223].

Індекс ТІОБЕ мов програмування

Вибір мови програмування для навчання програмування – важливе й відповідальне завдання викладачів. Критеріїв вибору мови програмування багато, деякі з них згадані вище. Одним з важливих критеріїв є частота використання мови світовим співтовариством розроблювачів.

Відома в професійному світі європейська фірма ТІОБЕ (www.tiobe.com) щомісяця на своєму сайті публікує індекс мов програмування, найбільше часто використовуваних у пошукових машинах у якості ключових слів при запитах.

Для знаходження індексу мови програмування використовуються результати пошуку в таких популярних пошукових службах як Google, Google Blogs, MSN, Yahoo, Wikipedia і Youtube.

Компанія ТІОБЕ заснована в жовтні 2000 року швейцарською компанією Synspace (www.synspace.com) і кількома приватними інвесторами. Назва ТІОБЕ – це скорочення від «The Importance Of Being Earnest», що в перекладі означає «Як важливо бути серйозним». Як відомо, «The Importance of Being Earnest» – це назва комедійної п'єси, написаної великим ірландським поетом і письменником Оскаром Уайльдом (Oscar Wilde) наприкінці дев'ятнадцятого століття. Ця назва була обрана засновниками ТІОБЕ для того, щоб підкреслити їх шире й професійне відношення до клієнтів, постачальників і колег. Більшість працівників ТІОБЕ починали свої кар'єри в дослідницькій лабораторії компанії Philips Electronics. Технологія, використовувана в ТІОБЕ, заснована на наборі інструментарію генератора компіляторів Elegant компанії Philips.

На сайті ТІОБЕ відзначено, що індекс ТІОБЕ не виявляє кращу мову програмування або мову програмування, якою описано найбільше рядків коду; адміністратори сайту вважають, що індекс Tiobe – індекс частоти пошуку в Інтернеті мови програмування – може відображати число кваліфікованих інженерів-програмістів, число навчальних дисциплін, у яких використовуються мови програмування, число робочих місць в усьому світі. Таким чином, індекс ТІОБЕ ранжує мови за кількістю людей, що користуються тією або іншою мовою програмування. Усього в цей рейтинг включено 50 мов програмування.

На рисунку 1 наведені індекси мов програмування, виміряні в грудні 2009 і 2008 років. Дані 2009 року подані стовпчиками сірого кольору, а 2008 року – смугастими стовпчиками. З малюнка видно, що в грудні 2008 і 2009 року найбільш популярним була мова програмування Java; рейтинг Java у грудні 2009 року становив 17,061%, а в грудні 2008 – 19,371%. Незважаючи на те, що мова програмування Java у рейтингу мов програмування залишається на першому місці за популярністю, частка запитів зменшилася на 2,31%. Друге місце в рейтингу мов програмування в грудні 2009 р. (16,285%) і 2008 р. (16,185%) займає мова програмування C, третє місце в 2009 році (в 2008 – четверте місце) – PHP (2009 – 9,77%; 2008 – 9,48%); четверте місце в 2009 році за мовою програмування C++ (9,175%), в 2008 році C++ посідав третє місце (10,895%).

Великий стрибок у попиті – з 35 місця в рейтингу грудня 2008 року (0,159%) до 13 місця в грудні 2009 року (1,159%) (див. рис. 1) – можна відзначити стосовно компільованої об'єктно-орієнтованої мови програмування корпорації Apple Objective-C.

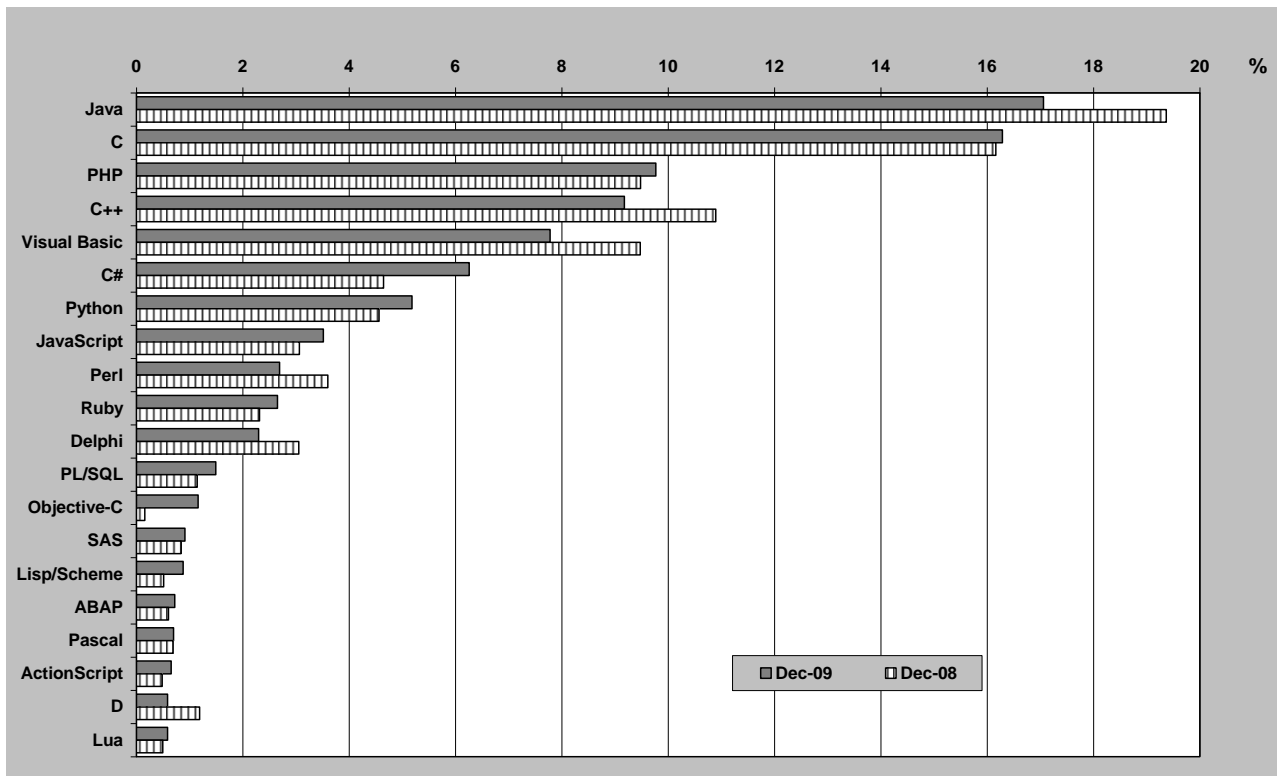


Рис. 1. Індеси мов програмування (дані із сайту <http://www.tiobe.com>: грудень 2009 vs грудень 2008)

На рисунках 2 і 3 представлені Парето-діаграми розподілу індексу мов програмування в грудні 2009 і 2008 років відповідно. Парето-діаграми (рис. 2, 3) показують, що мови програмування, які увійшли в двадцятку індексу ТІОБЕ, популярні майже у 90% користувачів. Крім того, індекси мов програмування Java (1-е місце) і C (2-е місце) у сукупності становили 33% у грудні 2009 року й 36% у грудні 2008 року; індекси мов програмування, що займають в ТІОБЕ з 3-го по 6-е місця (PHP, C++, Visual Basic, C#) у сукупності становили 33% у грудні 2009 року й 35% у грудні 2008 року.

Коротко ці закономірності можна виразити в такий спосіб:

– Дві найбільш популярні мови (приблизно рівні за популярністю) поглинають зараз приблизно третину світового користувацького попиту; ця частка має слабку тенденцію до зниження. Наступні чотири, менш популярні – ще одну третину; ця частка має слабку тенденцію до підвищення. Досить мати уявлення про ці шість мов (і навички їх застосування), щоб ефективно спілкуватися із двома з кожних трьох програмістів світу.

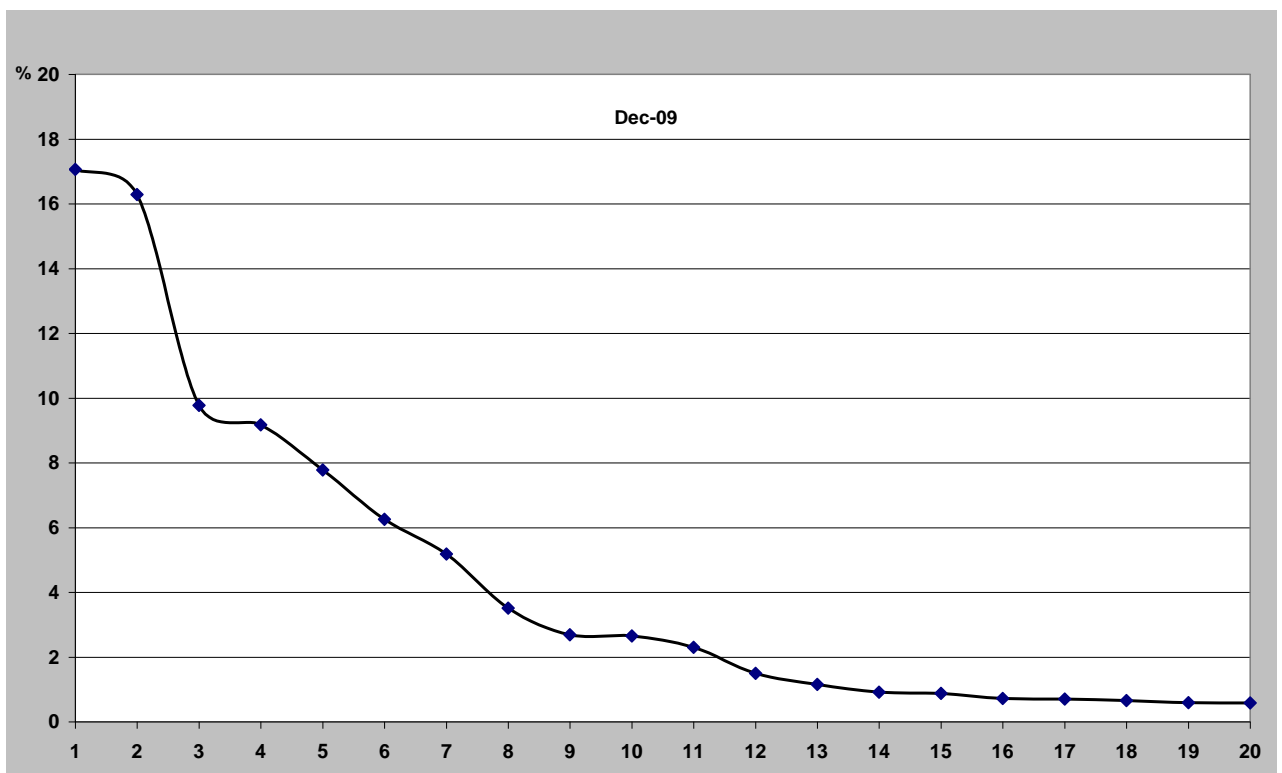


Рис. 2. Парето-діаграма розподілу індексу мов програмування в грудні 2009 року

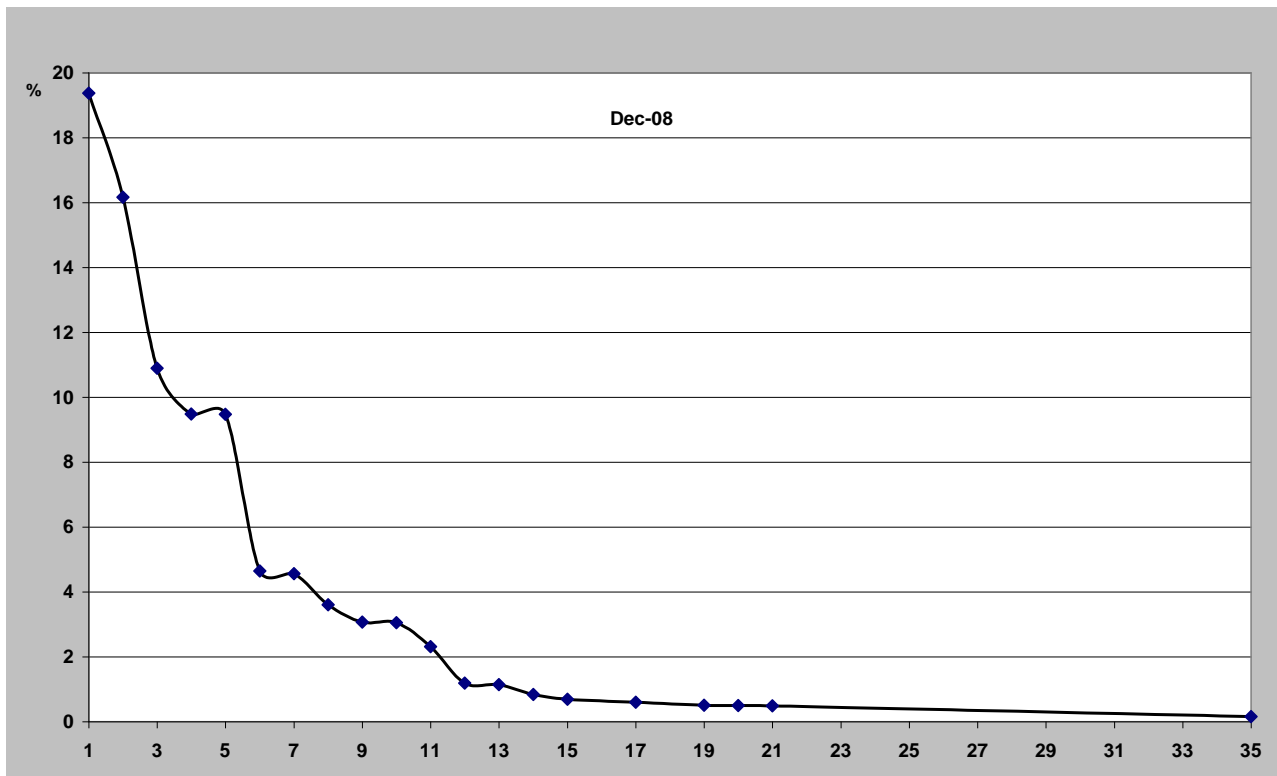


Рис. 3. Парето-діаграма розподілу індексу мов програмування в грудні 2008 року

На рис. 4 показана еволюція популярності мов програмування, що увійшли в першу десятку індексу ТЮВЕ у грудні 2009 року (Java, C, PHP, C++, Visual Basic, C#, Python, Javascript, Perl, Ruby). З графіка видно, що індекс мови програмування Java в 2002 році становив 24%, попит знизився в 2005 році (17,5%), після чого в 2006 році попит підвищився (22,5%) і почав знижуватися, в 2009 році він опустився до 17%. Слід зазначити, що за популярністю мова Java у рейтингу ТЮВЕ займала з 2002 року перше місце, за винятком 2005 року – у той рік перше місце зайняла мова C (20,5%), а мова Java була другою. Мова програмування C стійко з 2002 року посідала друге місце, за винятком 2005 року, коли вона виявилася найбільш затребуваною. Також, як і у випадку з мовою Java, спостерігається тенденція незначного зниження попиту на цю мову: з 20% в 2002 році до 16% в 2009 році. Така саме тенденція спостерігається щодо мови C++: з 2002 (15%) по 2008 (11%) роки C++ була третьою за популярністю мовою програмування, а в 2009 році виявилася на четвертій позиції з 9%, після мови PHP з 9,8% рейтингу. Зниження індексу популярності спостерігається також у мови Perl: з 7,4% в 2002 році до 2,7% в 2009 році. Найбільший індекс популярності (9%) в Perl спостерігався в 2004 році. Інші мови програмування – Visual Basic, C+, Python, Javascript, Ruby демонструють зростання популярності.

На рис. 5 показана еволюція кумулятивних індексів мов програмування, що займають у ранжирі ТЮВЕ перші два місця – Java і C (на рис. 5 цей кумулятивний індекс відзначений трикутниками), а також мов, що займають наступні чотири місця, із третього по шосте місце – PHP, C++, Visual Basic, C# (на рис. 5 цей кумулятивний індекс відзначений ромбиками). З малюнка видно, що кумулятивний індекс мов Java і C поступово знижується – з 44% в 2002 році до 33% в 2009 році; кумулятивний індекс мов PHP, C++, Visual Basic, C# слабо підвищується – з 31% в 2002 році до 33% в 2009 році (в основному зростання відбувається через популярну у розробці web-додатків мову PHP).

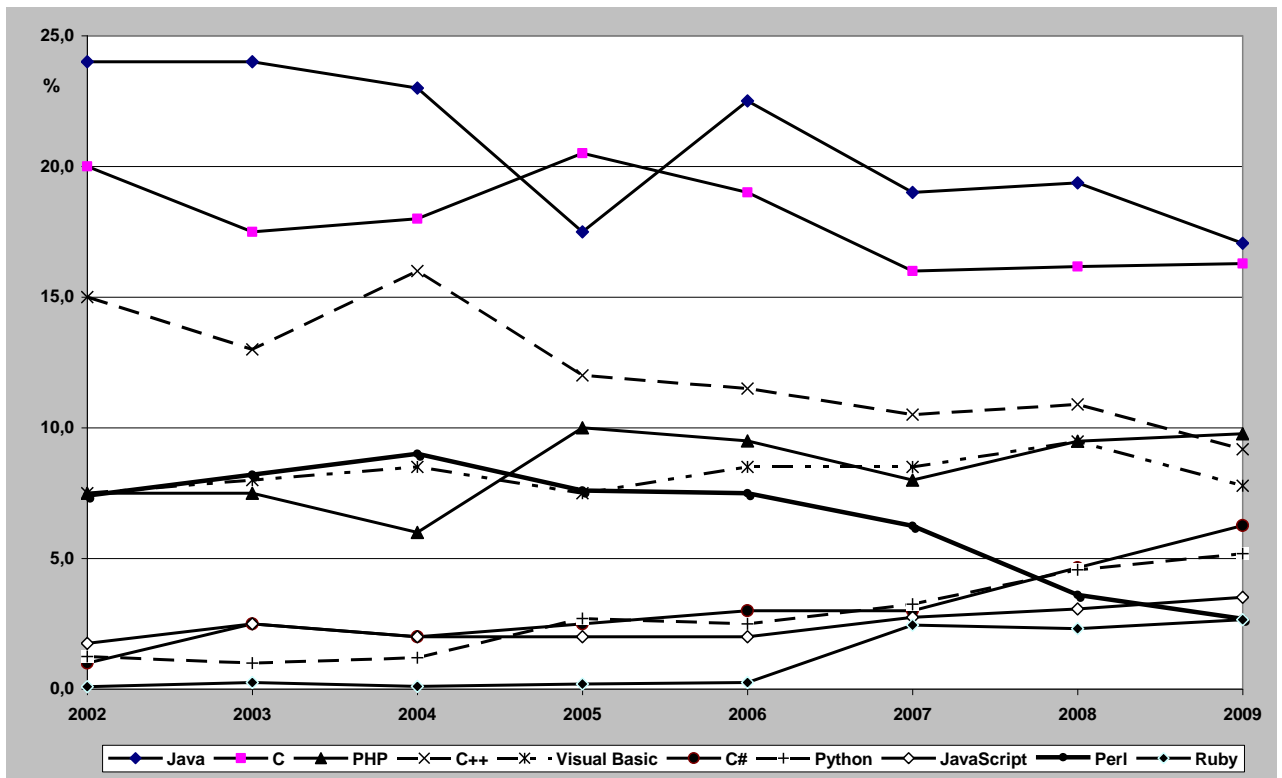


Рис. 4. Еволюція індексів мов програмування (дані із сайту <http://www.tiobe.com>)

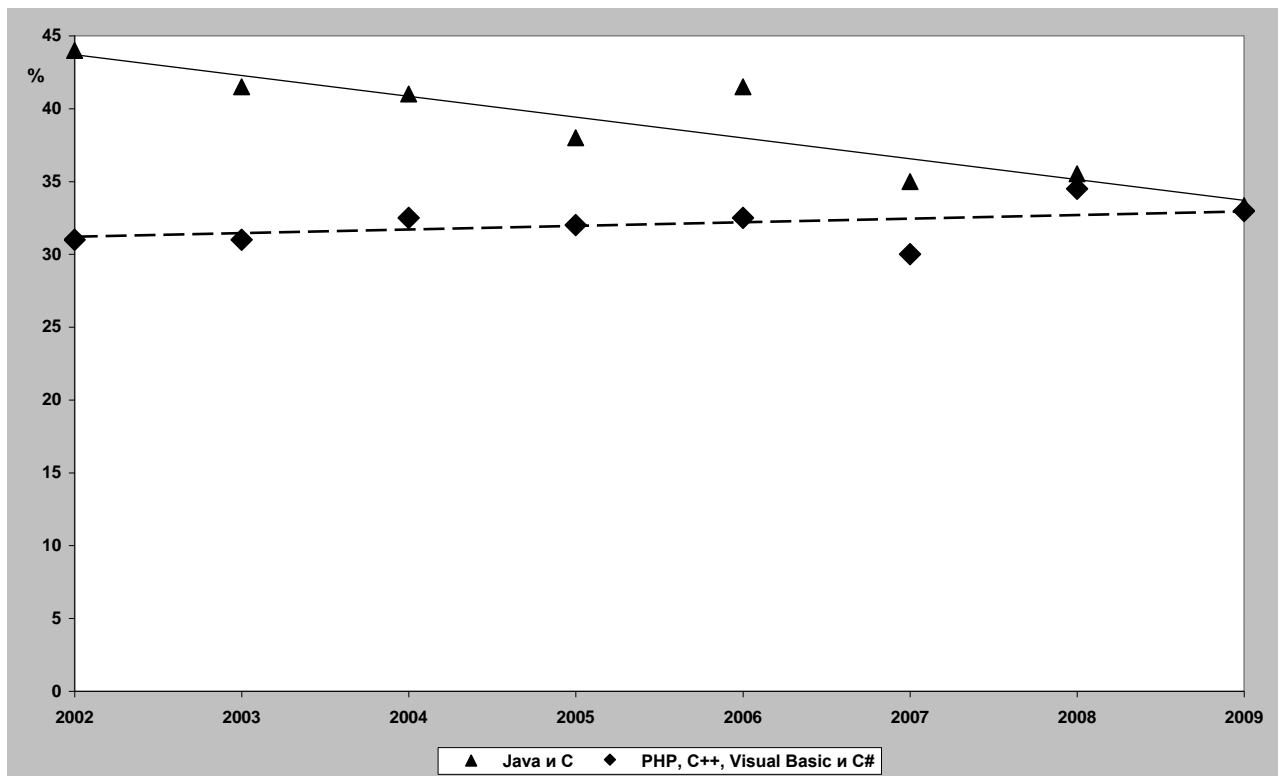


Рис. 5. Еволюція кумулятивних індексів мов програмування

Парадигми й типізація мов програмування

Кожна мова програмування являє собою штучну мову, спроектовану для виконання обчислень на машинах, зокрема, на комп'ютері; в мові програмування визначається набір лексичних, синтаксичних і семантичних правил і вона може бути використана для створення програм, виконуваних за допомогою машин. Мови програмування класифікуються за парадигмами, типізацією, структуризацією, семантикою і т.д. (визначення термінів «парадигми» і «типізація» для нашого контексту можна знайти, наприклад, в [13], [14]). Найбільш популярними парадигмами є:

–**Імперативна.** За цієї парадигмою описується процес обчислення у вигляді інструкцій, що змінюють стан програми;

–**Процедурна.** У процедурній парадигмі програма складається з послідовності операторів (інструкцій), що задають процедуру виконання завдання. У цій парадигмі пам'ять розглядається як сховище значень, зміст пам'яті може змінюватися за допомогою операторів надання значень. У цій парадигмі програмування завдання ділиться на алгоритмічні кроки, які послідовно виконуються у

відповідності до мовних конструкцій;

– **Об'єктно-орієнтована.** В якості основних концепцій використовуються поняття об'єкт (сутність в адресному просторі системи, що з'являється при створенні екземпляра класу), клас (описується устрій об'єкта), абстракція даних, спадкування, інкапсуляція й поліморфізм;

– **Узагальнена.** Описуються дані й алгоритми таким чином, що цей опис можна застосовувати до різних типів даних, не змінюючи його;

– **Подійно-орієнтована.** Являє собою спосіб побудови програми, при якому у кодї явно виділяється головний цикл додатка, тіло якого складається з вибірки події й опрацювання події;

– **Функціональна.** У функціональній парадигмі процес роботи програми трактується як обчислення значень функцій в математичному розумінні;

– **Аспектно-орієнтована.** Ця парадигма заснована на ідеї поділу функціональності для поліпшення поділу програми на модулі;

– **Компонентно-орієнтована.** Являє собою набір визначених правил і обмежень, що накладаються на механізм об'єктно-орієнтованого програмування. Програмний додаток у цій парадигмі являє собою набір компонентів з добре визначеними інтерфейсами; зміни в існуючий додаток вносяться створенням нових компонентів як доповнення або заміни вже наявних.

За результатами рейтингу ТЮВЕ грудня 2009 року є зрозумілим, що найбільш популярними є мови програмування з об'єктно-орієнтованою парадигмою; кумулятивний індекс популярності становить 55%. У грудні 2008 року індекс популярності таких мов становив 56%. В 2009 році 40% користувачів використовують процедурні мови (у грудні 2008 року – 40,4%), 3% – функціональні мови (у грудні 2008 року – 2,4%).

Мови програмування відрізняються також динамічними й статичними прийомами зв'язку змінної з типом. Динамічна типізація припускає зв'язок змінної з типом у момент надання значення, а не в момент оголошення змінної. У динамічній типізації в різних ділянках програми одна і та сама перемінна може набувати значень різних типів. При статичній типізації використовується прийом, при якому змінна, параметр підпрограми або значення функції, що повертається, зв'язуються з типом у момент оголошення типу; при цьому тип не може бути змінений пізніше.

Таблиця 1. Зіставлення мов програмування за парадигмами і типізацією

№ п/п	Мова програмування	Парадигми									Типізація		
		Імперативна	Процедурна	Функціональна	Об'єктно-орієнтована	Подійно-орієнтована	Компонентно-орієнтована	Аспектно-орієнтована	Узагальнена	Рефлексивна	Логічна	Статична	Динамічна
1.	Java	+			+				+	+		+	
2.	C	+	+									+	
3.	PHP	+	+		+					+			+
4.	C++	+	+		+				+			+	
5.	Visual Basic	+				+	+						+
6.	C#	+		+	+	+						+	
7.	Python	+		+	+			+		+			+
8.	JavaScript	+		+	+					+			+
9.	Perl	+	+	+	+				+	+			+
10.	Ruby	+		+	+			+		+			+

Продовження таблиці 1.

11.	Delphi (Object Pascal)	+			+	+			+			+	
12.	PL/SQL		+								+		+
13.	Objective-C	+			+					+		+	+
14.	SAS		+										+
15.	Lisp/Scheme			+									+
16.	ABAP		+										+
17.	Pascal	+	+									+	
18.	ActionScript	+			+	+						+	
19.	D	+			+				+			+	
20.	Lua	+		+	+			+		+			+

Індекс мов програмування, поданий на сайті ТЮВЕ, показує, що мовами зі статичною типізацією в грудні 2009 року користувалися 57,7%, а з динамічною типізацією – 42,3% користувачів; у грудні 2008

року ці цифри становили 59,4% і 40,6% відповідно.

У таблиці 1 подано зведення мов програмування, що увійшли в двадцятку в рейтингу ТЮВЕ у грудні 2009 року, за парадигмами і типізацією.

Кожна мова програмування може бути віднесена, як правило, більш, ніж до однієї парадигми (є, втім, і винятки із цього правила, наприклад, SAS, Lisp/Scheme, АВАР). Але всі мови (за винятком однієї, спеціально створеної для розширення можливостей типізації Objective-C) мають тільки один спосіб типізації.

Як видно з таблиці 1, найбільш часто використовуваною парадигмою є об'єктно-орієнтована парадигма програмування.

Привернення уваги педагогічного співтовариства України до наведених у цій статті даних може послужити стимулом для широкого обговорення національних педагогічних тенденцій, які склалися в цій сфері.

Література

1. Сейдаметова З.С., Меджитова Л.М. Моделі навчання основ програмування на молодших курсах комп'ютерних спеціальностей університетів // Науковий часопис НПУ ім. М.П.Драгоманова. Серія №2. Комп'ютерно-орієнтовані системи навчання. Зб.наукових праць / Редрада. – К.: НПУ ім.М.П.Драгоманова, 2009. № 7(14). – С. 103–107.
2. Kelleher C., R. Pausch. Lowering the barriers to programming: A taxonomy of programming environments and languages for novice programmers // ACM Computing Surveys, Vol. 37, Issue 2 (June 2005) – P. 83–137.
3. Manila L., Raadt M. An Objective Comparison of Languages for Teaching Introductory Programming // Baltic Sea (Proceedings of the 6th Baltic Sea conference on Computing education research: Koli Calling 2006). Uppsala, Sweden. Vol. 276. 2006. – P. 32–37.
4. McIver L., Conway D. Seven Deadly Sins of Introductory Programming Language Design // Proceedings of the 1996 international Conference on Software Engineering: Education and Practice (SE:EP'96). – IEEE Computer Society, 1996. – P. 309–316.
5. Radenski A. "Python First": A lab-based digital introduction to computer science // Proceedings of the 11th annual SIGCSE conference on Innovation and technology in computer science education. Bologna, Italy. 2006. – P. 197–201.
6. Bates B. C# as a first language: a comparison with C++ // Journal of Computing Sciences in Colleges. 2004. Volume 19, Issue 3. – P. 89–95.
7. Lunney T.F., McCullagh P.J., Lundy P.J. Java as the Lingua Franca for teaching graduate students // PPPJ; Vol. 42. Kilkenny City, Ireland. 2003. – P. 89–91.
8. Chandra S.S., Chandra K. A comparison of Java and C# // Journal of Computing Sciences in Colleges. 2005. Volume 20, Issue 3. – P. 238–254.
9. Pears A., Seidman S., Maimi L., Manila L., Adams E., Bennedsen J., Devlin M., Paterson J. A survey of literature on the teaching of introductory programming // Working group reports on ITiCSE on Innovation and technology in computer science education. Dundee, Scotland. 2007. – P. 204–223.
10. Pears A., Seidman S., Eney C., Kinnunen P., Malmi L. Constructing a core literature for computing education research // SIGCSE Bulletin, 2005. 37(4). – P. 152–161.
11. Chang C., Denning P.J. (chairs) et al. Computing Curricula 2001: Curriculum Guidelines for Undergraduate Degree Programs in Computer Science / A volume of the Computing Curricula Series – IEEE CS Press, ACM Press, 2001. – 240 p.
12. Computer Science Curriculum 2008 (CS 2008): An Interim Revision of CS 2001 – ACM, IEEE-CS, December 2008. – 108 p.
13. Snyder A. Encapsulation and inheritance in object-oriented programming languages // Proceedings OOPSLA '86, ACM Press, 1986. – P. 38–45
14. Floyd R.W. Turing Award Lecture, 1978: The Paradigms of Programming // Communications of the ACM, vol. 22(8), 1979. – P.455–460.