коммуникационных технологий во время обучения естественно-математических дисциплин для интеллектуального развития учеников. Подано примеры учебных заданий во время выполнения каких целесообразно использовать информационно-коммуникационные технологии.

Ключевые слова: интеллектуальное развитие, информационно-коммуникационные технологии обучения, педагогически целесообразное использование информационно-коммуникационных технологий в учебном процессе.

Some aspects of teaching weighted using of information and communication technologies during the learning of natural and mathematical sciences

Pidhorna T.

Abstract. One of the main tasks of learning is the intellectual development of students. In article examined conditions of pedagogical prudent use of ICT in the teaching of natural and mathematical disciplines for the intellectual development of students. Examples of learning tasks during the implementation of which should be used information and communication technologies.

Keywords: intellectual development, ICT training, teaching balanced use of ICT in the classroom.

УДК 378.016:004.438(045)

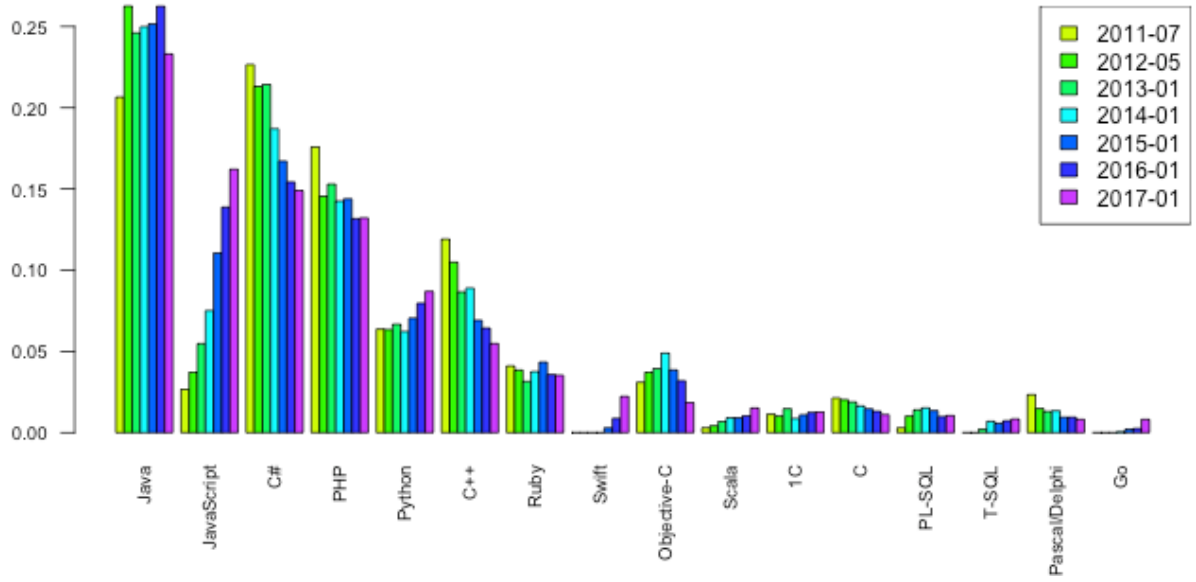
Горошко Ю. В., Цибко Г. Ю.
Чернігівський національний педагогічний університет імені Т.Г.Шевченка

Про перспективи використання мови Swift у навчанні програмування

Анотація: У статті аналізується добір мов програмування для навчання основ алгоритмізації і програмування у школі і педагогічному вищому навчальному закладі. Обґрунтовується можливість застосування у навчальному процесі мови програмування Swift. Swift – потужна відкрита мова програмування, що широко використовується для написання програм для операційних систем від фірми Apple і доступна в операційних системах на основі GNU/Linux. Описуються основні види величин, правила їх опису у Swift, детально розглядаються особливості типізації даних. Зазначається, що поширенню мови Swift у навчальних закладах сприятиме активне створення відповідних вільно поширюваних інтегрованих середовищ розробки для Windows і GNU/Linux.

Ключові слова: навчання програмування, мови програмування, мова Swift.

На даний час в загальноосвітніх середніх і педагогічних вищих навчальних закладах навчання основ алгоритмізації і програмування відбувається з використанням мови Pascal в середовищах програмування FreePascal та Lazarus. Але у все більшій кількості вчителів і викладачів, учнів і студентів виникає питання стосовно доцільності використання саме цієї мови. Програмування, називають її “мертвою” мовою, і це не сприяє зацікавленню у її вивченні. Чому так? Мова Pascal потрапила у вузьку нішу створення реальних програмних продуктів, не дивлячись на те, що вона залишається прекрасною мовою програмування. Трапилось це завдяки, насамперед, поганому менеджменту. Все менше проектів реалізують з використанням цієї мови. Вільно поширювані інтегровані середовища розробки не призначені для написання програм для смартфонів, вони не кращим чином інтегровані в Unix та Linux. До цього часу погано вирішеними є і проблеми з кодуванням текстових рядків, хоча це стосується і багатьох інших мов. Враховуючи сучасні реалії програмісти просто перейшли до використання інших технологій [5].



На наведеній вище діаграмі [4] можна побачити, як з кожним роком використання мови Pascal скорочується в реальному програмуванні.

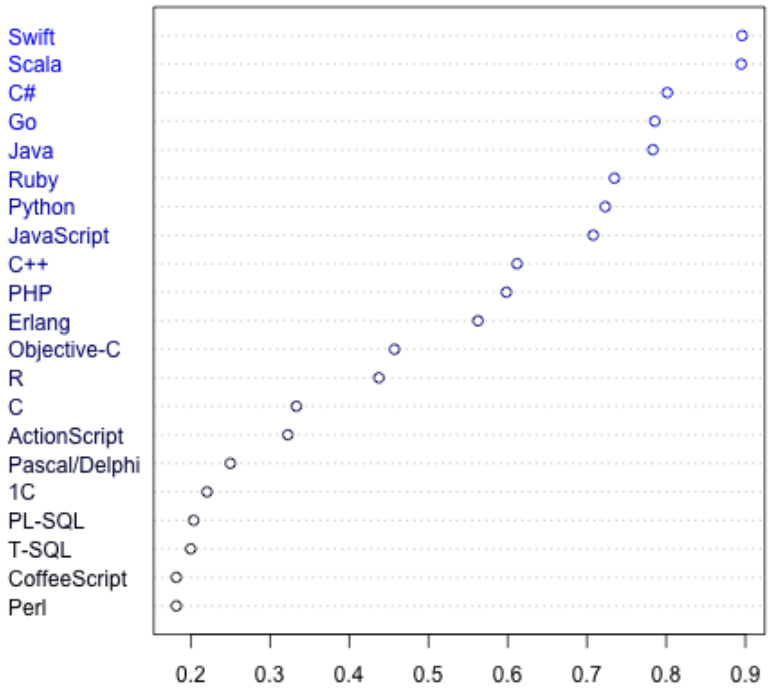
Одним з варіантів розв'язання посталої проблеми вбачається розгляд питання про перехід до навчання іншої основної мови в шкільному курсі інформатики. Учні, які активно займаються олімпіадним програмуванням, практично всі перейшли на використання мови C++. В [2] Жуковський В.С., Коротун О.В. пропонують вивчати саме цю мову у шкільному курсі інформатики. Але для пересічного учня ця мова залишається надто складною. Складною вбачається і мова Java з її чистим об'єктно-орієнтованим підходом. Одним з варіантів може бути вибір мови Python, яка є досить популярною і в той же час її основи доступні для учнів. Але і тут є свої проблеми. Програми,описані цією мовою, виконуються через інтерпретатор, а, отже,задачі за їх допомогою розв'язуються досить повільні. В реальності Python теж вже здає свої позиції. Свого часу ця мова була достатньо популярною в галузі комп'ютерних ігор, але ця популярність вже давно втрачена. У веб-програмуванні JavaScript все частіше використовується замість мови Python, як скриптової мови програмування. З методичної точки зору недоліком є відсутність у мові Python строгої типізації, що може негативно вплинути на якість вивчення основ алгоритмізації і програмування. П.Г. Шевчук пропонує вивчати мову C# у класах технологічного профілю [1].

Іншим варіантом може бути навчання достатньо нової мови Swift.

Знання мови програмування Swift є на даний момент одним з найбільш затребуваних, про це свідчать результати опитування, проведеного однією з найбільших фріланс-бірж Upwork [3]. Ця компанія оприлюднила рейтинг навичок, які необхідні для успішної роботи в Інтернеті (тобто для так званих фрілансерів). Найбільш перспективними напрямами стали машинне навчання і розпізнавання мови, а на другому місці виявилася мова програмування Swift. Частіше за все від співшукачів потребують створювати ефективні мобільні додатки для iPhone і iPad. Upwork є найбільшим ринком фрілансерів у США і може досить коректно визначати тренди на цьому ринку. В її дослідженні порівнюються найбільш швидко зростаючі щодо затребуваності навички в заключному кварталі 2016 року.

Фірма Apple позиціонує Swift як потужну мову програмування для її платформ. Починаючи з версії 3.0, компіляція додатків, описаних з використанням цієї мови, можлива для iOS, macOS, watchOS і Linux, а сама мова була переведена на відкриту ліцензію Apache. Навесні 2016 року фірма Google оголосила, що розглядає мову Swift як заміну мови Java для операційної системи Android.

Наведемо індекс задоволеності, що стосується тієї самої мови програмування, вибраної для наступного проекту після завершення поточного [4]:



На першому місці виявилася якраз мова Swift, якій з розміщеною другою мовою Scala надають звання "ідеальної мови".

На сайті фірми Apple [8] можна ознайомитись зі списком університетів США, які включили мову Swift до навчальних програм це:

1. Університет Аберіствайт;
2. Політехнічний університет штату Каліфорнія;
3. Коледж Куеста;
4. Дрексельский університет;

5. Університет Фулл Сейл;
6. Школа Ingésup;
7. Технічний коледж північно-західного Канзаса;
8. Плімутський університет;
9. Королівський мельбурнський технологічний інститут;
10. Південний методистський університет;
11. Стенфордський університет;
12. Мюнхенський технічний університет;
13. Технологічний інститут Монтеррея;
14. Каліфорнійський університет в Санта-Круз.

Програми, описані за правилами мови Swift, компілюються за допомогою LLVM. LLVM – низькорівнева віртуальна машина, це універсальна система трансформації, оптимізації та аналізу програм. За допомогою цієї машини байт-код перетворюється в машинний код, крім того вона використовується, як оптимізуючий компілятор [6]. Використовується для опрацювання описів програм з багатьох мов, в тому числі і Swift. Розроблена мова під керівництвом відомого спеціаліста Кріса Латтнера.

Розглянемо цю мову детальніше. Компільована об'єктно-орієнтована мова програмування Swift базується на мовах Objective-C і C, увібравши в себе все найкраще з них. Розробникам мови вдалося знайти баланс між легкістю читання коду програми і компактністю цього коду.

Можна віднайти багато спільних характеристик у мов Swift і Pascal, що полегшить оволодіння мовою Swift тими вчителями, які вивчали Pascal у педагогічному вищому навчальному закладі і навчають цієї мови в школі.

Мова Swift є строго типізованою. До основних типів відносять Int для цілих типів, Double і Float для дійсних, Bool для значень істинно/хибна, String для тексту. У цій мові реалізовано потужні версії трьох основних типів колекцій, Array (масив), Set (множина) і Dictionary (словник). Поряд з використанням змінних розробниками пропонується широке використання констант для того, щоб зробити код безпечнішим. У мові також визначено такий тип даних, як кортеж, тобто набір значень, можливо різнотипних. В разі звернення до функцій можна повертати одночасно кілька значень, поєднавши їх у кортеж.

Константи і змінні повинні бути оголошеними перед їх використанням. Константи оголошуються за допомогою використання службового слова let, а змінні – var:

```
let maxValue = 10
var Value = 0
```

У наведеному прикладі тип змінної не вказано, тому він буде визначений за допомогою компілятора на основі наданого змінній значення (в даному випадку Int). За необхідності тип змінної можна вказати:

```
var Value: Double = 0
```

Вказати тип можна і для констант:

```
let minByteValue: UInt8 = 0
let maxByteValue = UInt8.max
```

У останньому прикладі була оголошена константа типу беззнакове однобайтове ціле число із значенням 255.

Якщо у рядку коду використовується один оператор, то в кінці можна не ставити крапку з комою, але кілька операторів в одному рядку потрібно розділяти цим знаком:

```
var a = 5; let b = 6
```

Для полегшення читання коду в числових значеннях можна використовувати додаткові нулі і знаки підкреслювання:

```
let f = 00012.30
let g = 1_000_000.000_000_1
```

Назви типів можна змінювати для кращого розуміння контексту

```
typealias Byte = UInt8
```

У цій мові реалізована робота з опціональними типами даних, тобто такими, що для змінних таких типів можна визначити, чи надано змінній якесь значення, або не надано ніякого значення (nil). Для забезпечення безпеки не можна передати значення опціонального типу туди, де очікується такий самий, але неопціональний тип. Для порівняння всі стандартні типи мови Pascal є неопціональними.

Для позначення опціонального типу після назви типу пишуть знак питання:

```
var someValue: Int? = 123
```

Після виконання такого оператора змінній SomeValue надається значення 123.

```
someValue = nil А після виконання оператора
змінній SomeValue не надається будь яке значення взагалі.
```

Розглянемо ще один приклад:

```
let intFromText = Int("123")
```

В цьому прикладі оголошується константа, якій надається цілочисельне значення, як результат перетворення в ціле число рядка "123". Тип цієї константи буде автоматично визначений як Int?, тобто "опціональний цілочисельний тип". Це відбувається тому, що в даному коді має місце спроба

перетворити рядок у ціле число за допомогою використання методу Int (рядок). Зрозуміло, що таке перетворення не завжди може бути виконане успішно, наприклад спроба перетворення Int (“four”) буде неуспішною. Тому за методом Int(рядок) повертається опціональний Int замість неопціонального і після виконання

```
let anotherIntFromText = Int(“four”)
```

константа anotherIntFromText буде надано тип Int? і значення nil.

В мові визначено оператори для отримання діапазонів значень. Закритий діапазон можно визначити, наприклад, так:

```
let closeRange = 0...9 // всі цілі значення від 0 до 9 включно
```

Напіввідкритий – так:

```
let range = 1..<5 // цілі числа від 1 до 4 включно
```

Розглянемо тепер використання символів і рядків. Важливою перевагою мови Swift є те, що в ній реалізовано підтримку Юнікоду.

Для окремих символів використовується тип Character, а для їх наборів — тип String.

```
let someChar: Character = “f”
var someStr = “ У рядку можуть бути English та українські символи, а також ? ”
```

Створити порожній рядок можна так:

```
var s1 = “”
```

або

```
var s2 = String()
```

Властивість рядка isEmpty наюуває значення true, якщо рядок порожній:

```
let b = s1.isEmpty    //  b=true
```

В рядках можуть міститися наступні спеціальні символи:

\0 — нульовий символ;
\\ - backslash
\t — табуляція
\n — перехід на новий рядок
\” — лапки
\’ — апостроф
\u{код} — символ, що відповідає вказаному коду Юнікод, — шістнадцяткове число.

Наприклад, щоб розмістити у константі текст із лапками, потрібно виконати:

```
let someText = “\”Розум — сила!\”’
let anotherText = “Символ \u{2665} — це сердечко”
```

Для об’єднання рядків можна використовувати операцію “+”:

```
let s1 = “ab”
let s2 = “юя”
var s = s1 + s2   //  s=”abюя”
s+=s1    //   s=”abюяab”
let c: Character = “k”
s.append(c)   //   s=”abюяabk”
```

Символи рядка зберігаються у колекції characters, а через властивість count цієї колекції визначається кількість символів у ній:

```
let n = s.characters.count  //  n=7
```

Оскільки для зберігання різних символів рядка можуть знадобитися різні об’єми пам’яті, для отримання і-го символа потрібно переглянути всі попередні і-1. Тому символи рядка не можуть бути проіндексовані за допомогою цілочисельних значень. Для звернення до окремих символів використовують метод index рядка, а також властивості startIndex (позиція першого символа) та endIndex (позиція після останнього символа). Розглянемо це на прикладах:

```
let s = “abcdefg”
let c1 = s[s.startIndex]    //  c1=”a”
let c2 = s[s.index(after: s.startIndex)]   // c2=”b”
let c3 = s[s.index(s.startIndex, offsetBy: 4)]   // c3=”e”
let c4 = s[s.index(before: s.endIndex)]    // c4=”g”
```

Спроба доступу до символа за неіснуючим індексом призведе до помилки:

```
s[s.endIndex]    //   Error!
s[s.index(after: endIndex)]   //  Error!
```

Перебирання всіх символів рядка можна зробити з використанням циклу for, про який йтиме мова пізніше

Щоб вставити символ у рядок за вказаним індексом, використовують метод insert(_:at:), наприклад:

```
var s = “abc”
s.insert(“ю”, at: s.endIndex) // s=”abcю”
```

Для вставляння рядка у рядок за вказаним індексом використовують метод insert(contentsOf:at:), наприклад:

```
s.insert(contentsOf: “ef”.characters, at: s.index(before:
s.endIndex)) // s=”abcюef”
```

Вилучити один символ можна так:

```
s.remove(at: s.index(before: s.endIndex)) // s="abcюe"
```

Для вилучення кількох символів виконують дії, як у наступному прикладі:

```
let range = s.index(s.endIndex, offsetBy: -2)..<s.endIndex
s.removeSubrange(range) // s="abc"
```

Зрозуміло, що маніпуляції з рядками виглядають дещо громіздко, у порівнянні з мовою Pascal, але повна підтримка Юнікоду цього варта.

Для виведення у консоль використовують функцію print(), у масив виведення можна включати значення констант і змінних:

```
let someValue = 12
print(someValue)  //  12
print("значення someValue \(someValue)") // значення someValue 12
```

Зрозуміло, що в межах статті не можна познайомитись навіть з основами Swift, проте в мережі Інтернет можна знайти багато даних стосовно цієї мови. Завантажити програмне забезпечення для ознайомлення з мовою можна на офіційному сайті мови [7].

Отже Swift – потужна відкрита мова програмування, яка вже широко використовується для написання програм в межах операційних систем від фірми Apple і з якою можна познайомитись в операційних системах на основі GNU/Linux. Перспективи її використання в процесі навчання курсів інформатики і в школі педагогічних вищих навчальних закладах залежатимуть від розробки вільно поширюваних IDE для цієї мови в операційних системах Windows і GNU/Linux.

Список використаних джерел

1. Шевчук П.Г. Методика навчання програмування учнів класів технологічного профілю на основі використання мови C# : автореф. дис... канд. пед. наук:13.00.02 / П. Г. Шевчук. – К., 2013. – 20 с.
2. Жуковський В.С. Про перспективу введення мови програмування С++ в навчальний процес загальноосвітніх навчальних закладів / В.С. Жуковський, О.В. Коротун // Комп'ютер у школі та сім'ї №1(113), 2014. – С. 23-25.
3. Язык программирования Apple Swift стал вторым по востребованности среди фрилансеров [Електронний ресурс]. – Режим доступу: http://it-news.club/programming-language-swift-by-apple-has-become-the-second-most-popular-among-freelancers/
4. Рейтинг языков программирования № 7: PHP уходит с пьедестала [Електронний ресурс]. – Режим доступу: https://dou.ua/lenta/articles/language-rating-jan-2016/
5. Мысли о Python 3 [Електронний ресурс].- Режим доступу: https://habrahabr.ru/post/147281/
6. LLVM [Електронний ресурс]. – Режим доступу: https://en.wikipedia.org/wiki/LLVM
7. Офіційний сайт мови програмування Swift [Електронний ресурс]. – Режим доступу: https://swift.org/
8. Swift. Язык программирования с открытым кодом. Мощь, простота и потрясающие приложения [Електронний ресурс]. – Режим доступу: http://www.apple.com/ru/swift/

О перспективах использования языка swift в обучении программированию

Ю.В. Горошко, А.Е. Цыбко

Аннотация. В статье анализируется отбор языков программирования для обучения основам алгоритмизации и программирования в школе и высшем педагогическом учебном заведении. Обосновывается возможность использования в учебном процессе языка программирования Swift. Swift — мощный открытый язык программирования, широко используемый при написании программ для операционных систем от фирмы Apple и доступный в операционных системах на основе GNU/Linux. Описываются основные виды величин, способы их представления в Swift, детально рассматриваются особенности типизации данных. Отмечается, что распространению языка Swift в учебных заведениях Украины будет способствовать активное создание соответствующих свободно распространяемых сред разработки для Windows і GNU/Linux.

Ключевые слова: обучение программированию, языки программирования, язык программирования Swift.

About prospects of the use of language swift in the learning of programming

Y.V. Horoshko, H.Y. Tsybko

Resume. In the article the selection of programming languages for learning the basics of algorithmization and programming in school and teachers' training university is being analyzed. The applicability of programming language Swift in education is being substantiated. Swift is a powerful open programming language widely used for applications development within the operating systems from the company Apple and is available in operating systems based on GNU / Linux. Main types of variables, ways of their description in Swift, features of data typing are being described in detail in the article. It is noted that the spreading of the language Swift in schools of Ukraine is possible with the active creation of the free integrated development environments for Windows and GNU / Linux.

Keywords: learning a programming language, programming languages, programming language Swift.