

УДК 130.2: 811.93

ЛЮДСЬКІ МОВИ ТА МОВИ ПРОГРАМУВАННЯ

Ілля Ухов

У цій статті описуються деякі візуальні схожості між мовами програмування та звичайними мовами на прикладі простих прикладів текстів, написаних деякими мовами програмування. Було виявлено, що в обох типах мов присутні схожі синтаксичні й семантичні одиниці, а також конструкції, що виражають як сутності, так і дії. Також, обох типах мов можливе написання текстів, повністю аналогічних за змістом, однак у більш стислій, лаконічній формі та у більш детальній.

Ключові слова: мова, програмування, порівняння.

ВСТУП

Мова — абсолютно звична та повсякденна річ, якою користується кожна людина. Водночас, кожна мова має багатовікову історію, та є чи не найвеличнішим культурним надбанням свого народу. Та що таке мова із суто практичної точки зору? Це — спосіб передачі та збереження інформації за допомогою слів.

Мова програмування. Перша половина цього терміну — слово “мова” — означає, що ми також маємо справу із засобом передачі інформації за допомогою слів. Та друга половина — слово “програмування” — робить уточнення. Уточнення, що ця мова призначена не для спілкування людей із людьми, а для спілкування людини із комп'ютером.

Очевидно, що “мови” та “мови програмування” мають багато спільного та багато відмінного. Однак, порівняння цих двох явищ не популярною темою наукових праць. Насправді, я не знайшов ніяких наукових праць з цієї теми взагалі. Це закономірно — дослідження в цьому напрямі не надають ніяких видимих перспектив щодо, наприклад, винаходу надсвітлових космічних кораблів чи довершеного штучного інтелекту.

Метою моєї статті є доповнення власними думками того, що вже висловлено, авторів деяких з думок буде згодом згадано. Можливо, нечисленні наразі дослідження із даної теми в майбутньому стануть фундаментом для великих теорій.

Активний розвиток мов програмування почався лише у середині колишнього століття. Водночас, історія мов, якими ми спілкуємось, сягає сотен та тисяч років, у них накопичено багатовіковий досвід усього людства. Можливо, завдяки подібним дослідженням цей досвід допоможе майбутнім розробникам створювати все більш довершені мови програмування. Які, в свою чергу, дозволять майбутнім програмістам все ефективніше створювати програмні продукти.

Звичайно ж, увагу деяких людей (зазвичай, програмістів), привертає спільність слова “мова” у термінах “мова” та “мова програмування”. Вони висловлюють різноманітні, іноді полярно протилежні думки.

Так, користувач portquest2016 на інтернет-форумі linux.org.ru висловив думку про те, що мови програмування насправді не є мовами та термін є некоректним[1].

У 2004 році російське радіо “Свобода” влаштувало інтерв'ю з двома професійними програмістами Владіміром Губайловським та Александром Сергеевим щодо спільних та відмінних рис мов програмування та “живих” мов[2].

Англомовний сегмент інтернету дещо багатший на думки із цієї теми: декілька користувачів форуму quoga.com висловили свої критерії, за якими мови програмування принципово відрізняються від людських мов[3]. Деякі користувачі у своїх міркуваннях зверталися до математики.

Іноді до порівняння людських мов із мовами програмування звертаються цілком випадково. Так, у 2015 році, внаслідок появу пропозицій у деяких штатах США замінити вивчення іноземної мови вивченням мови програмування, журнал Language Magazine опублікував невелику статтю на подібну тему[4].

Відносно великою за об'ємом є стаття, опублікована на своєму персональному сайті програмістом Раджешем Кумаром (Rajesh Kumar)[5]. Він приділяє велику увагу численним спільним рисам мов програмування та людських мов.

Природно, що теми торкаються також лінгвістичні організації. На форумі linguistics.stackexchange.com автор публікує тему із питанням, чи наявні взагалі якісь чіткі відмінності між людськими мовами та мовами програмування[6].

Ще одним зазначеним мною джерелом буде невелика, та серйозна, інформативна і водночас легка для сприйняття стаття, написана співробітниками голландського Інституту психолінгвістики імені Макса Планка (Max Planck Institute for Psycholinguistics) Джулією Удден (Julia Udden), Гаральдом Гаммарстрьомом (Harald Hammarström) та Ріком Янсенем (Rick Jansen) як відповідь на запитання користувача про спільні та відмінні риси “природних” мов та мов програмування[7].

Спільною думкою багатьох дослідників можна вважати схожість принципів синтаксису та семантики: подібно до того, як в звичайних мовах кожна літера й слово мають займати чітко визначені місця в словах та реченнях, так і в мовах програмування неправильно записана команда чи команда, розміщена в неналежному місці стане синтаксичною помилкою. Водночас, кожне слово в людських мовах та мовах програмування має власний зміст.

Така наука, як семіотика, узагальнює це явище під термінами знак — письмове позначення, і денотат — значення або зміст, що воно має. Наявність знаково-денотатної системи в мовах програмування зазначають такі автори, як Сергій Орлов у книзі “Теорія та практика мов програмування. Підручник для вузів” (глава 1) та Віталій Кауфман у книзі “Мови програмування. Концепції та принципи” (глава 1.15)[8][9].

Отже, в основі людських мов та мов програмування полягає одна система. На мою думку, це очевидно, адже не просто так головне слово в обох термінах — “мова”. Це слово начебто натякає, що обидва явища мають спільну природу. Та

поглянемо дещо детальніше на синтаксис і семантику мов програмування і з'ясуємо, чи схожі вони на такі у людських мовах.

АТОМ

З чого складається мова? Усі знають, що мова складається зі слів, речень, літер, звуків, абзаців, словосполучень. Та що з цього можна вважати найменшою частинкою мови, її атомом? Абзац — це точно надто велика річ, щоб бути найменшою частиною. Може, літера? Спробуємо. Та якщо ми зберемо до купи багато випадкових літер, чи отримаємо ми мову? Навряд чи. Також, не в усіх мовах літери взагалі є.

Хоч речення і зазвичай набагато більші за літеру, та саме вони визнані як одиниця мови. Тому, що речення виражає закінчену думку. Якщо ви випишете з цього тексту будь-які окремі слова і прочитаєте їх, то навряд чи ви зможете щось собі уявити, чи щось зрозуміти. Якщо ж виписати речення, то кожне з них про щось вам розповість, донесе до вас якусь інформацію.

А що вважати атомом мови програмування? У мовах програмування немає речень. Та в них також існує базова одиниця — інструкція. Комп'ютерна програма складається з інструкцій. Типова інструкція:

```
human.run()
```

Саме таким нескладним чином виглядає більша частина коду в сучасному програмуванні. "human" — це об'єкт, тобто сутність, що має властивості та методи. Метод — це деякий блок коду, який виконується тоді, коли цей метод викликається. Для цього в більшості мов програмування після назви об'єкта ставиться крапка, потім пишеться назва методу із круглими дужками наприкінці.

Легко здогадатися, що ця інструкція викликає метод "run" в об'єкта "human". "Human run". "Людина біжить". Нічого не нагадує? Людина біжить — це є просте, повне, непоширене речення, де "людина" — це підмет, а а "біжить" — це присудок!

Отже, як ми здогадалися, стандартна інструкція мови програмування за синтаксисом та змістом дуже схожа на прості непоширені речення речення. А як щодо інших варіантів?

У програмі також можуть зустрічатися ще простіші інструкції у двох випадках:

По-перше, коли об'єкт, чий метод викликано, існує, але явно не записується для лаконічності коду. Це часто використовується, якщо метод викликано всередині блоку коду, що описує цей самий об'єкт, який має на увазі:

```
run()
```

Така інструкція схожа на речення з одного присудка, коли підмет можна встановити з контексту. Наприклад:

```
— Що Іван робить?  
— Працює.
```

По-друге, іноді метод може взагалі не мати конкретного об'єкту, в котрого його викликано:

```
getting_darker()
```

"Темніє". А "темніє" — це просте безособове речення! За визначенням, сутність-підмет не визначена чітко в безособових реченнях. Так само, як не визначений і об'єкт у попередньому коді.

Тепер навпаки, розглянемо складніші речення та програмні інструкції:

```
human.go(park)
```

Ви вже здогадалися, що уявна людина йде у парк. "Людина йде до парку". Це речення, на відміну від попередніх, є поширеним і має додаток. У коді слово "park" є аргументом метода.

До речі, у багатьох сучасних мовах програмування є можливість додавати надписи перед аргументами методів:

```
human.go(to: park)
```

Код став ще більш зрозумілим і ще більш схожим на речення "Людина йде до парку".

Аргументи можуть нагадувати не лише додатки у звичайній мові, а й інші другорядні члени речення:

```
builders.build_house(color: red)  
Peter.meet(Emily, at: evening)
```

"Будівники будують червоний будинок" — аргумент методу символізує означення. "Пітер зустріне Емілі ввечері" — аргументи символізують додаток та обставину.

Як відомо, у нашому мовленні ми користуємося також складними реченнями. Мені не вдалося знайти в мовах програмування інструкцію, схожу на складносурядне речення, однак дуже часто застосовуються інструкції такого виду:

```
human.go(to: gps_navigator.point())
```

“Людина йде туди, куди їй вказує GPS-навігатор”. Це — складнопідрядне речення. І так само, як складнопідрядне речення можна розділити на два простих речення — “GPS-навігатор вказує місце, куди йти. Людина йде в це місце.” — так і вищевказаний код можна записати у вигляді двох інструкцій:

```
let place = gps_navigator.point()
human.go(to: place)
```

Цей код має абсолютно такий самий зміст, як і попередній. Строго кажучи, попередній код також складається з двох інструкцій, але там вони вкладені одна в іншу, і візуально виглядають, як одна. Відміна в тому, що в цьому випадку явно створюється новий об’єкт “місце”. Однак, як ви могли помітити, отриманий текст з двох простих речень видається громіздким та недотепними в порівнянні із початковим складним реченням. Так само і в програмі — перший варіант є більш лаконічним та легким для читання і розуміння.

РОЗМАЇТТЯ?

У світі людських мов головною мовою міжнародного спілкування стала англійська. Та усі європейські мови схожі на англійську — в них усіх слова складаються із літер, а в реченнях є підмети і присудки, іменники для позначення сутностей і дієслова для позначення дії.

Так само і в сучасних мовах програмування — більшість них дуже схожа. Усі приклади коду тут були написані мовою програмування Swift. Зараз будуть наведені приклади нашої класичної інструкції “Людина йде до парку” на деяких інших сучасних мовах програмування, та продубльований приклад мовою Swift:

```
Swift:      human.go(to: park)
Kotlin:    human.go(to = park)
Python:    human.go(to=park)
```

Не надто велике розмаїття. До речі, в прикладі на мові Kotlin можна записати інструкцію точно так само, як і в прикладі мовою Python, вона від цього не втратить зміст та коректність. Пробіли навколо знака рівності є лише елементом стилю програмування цією мовою.

Як ви, можливо, здогадуєтесь, список прикладів можна було б зробити у багато разів довшим. Та це було б не надто цікаво — приклади б повторювали ті, що вже наявні.

Щоб додати хоч трохи розмаїття, я звернуся до більш старих та консервативних мов програмування PHP та C++:

```
PHP:       $human->go($park);
C++:      human->go(park);
```

В цих мовах немає підписів аргументів методів. Як бачимо, навіть ці приклади не так вже й сильно відрізняються від попередніх.

Ця схожість для мов програмування є цілком закономірною — більшість сучасних мов успадкували свій синтаксис від однієї мови — Сі (англ. “С”). Якщо в синтаксисі якоїсь комп’ютерної мови з’являється вдале нововведення, то й інші мови починають його переймати.

З людськими ж мовами, вочевидь, дещо інша ситуація — навряд чи можна згадати приклад, щоб елементи синтаксису активно переходили з однієї мови до інших, народи світу ніколи не мали бажання робити свою мову схожою на якусь чужу мову. Активно запозичуються лише слова. Також, синтаксис людських мов незрівнянно більш складний, формувався протягом в сотні разів довшого проміжку в історії.

Через ці причини, на мою думку, синтаксис людських мов значно більш різноманітний — досить згадати, що навіть загалом схожі європейські мови можуть мати багато відмінностей у синтаксисі. А от коли згадати про мови народів Азії, Африки, Океанії...

ВИСНОВКИ

Щодо першої частини, де порівнювалися речення у людських мовах з інструкціями мов програмування, потрібно зробити два уточнення:

По-перше, подібного виду інструкції наявні лише в так званих об’єктно-орієнтованих мовах програмування. Не будемо вдаватися в деталі терміну, головне — це такі мови, програмування якими означає створення уявних об’єктів реального світу, як приклад — наша уявна людина, що “йшла” до уявного парку. Таких мов програмування наразі більшість. Різноманітні та часом чудернацькі не-об’єктно-орієнтовані мови програмування на мою думку надто специфічні, щоб шукати в них схожості з людськими мовами та згадувати їх в цій статті.

По-друге, у мовах програмування багато також зовсім іншого виду інструкцій, які не схожі на речення. Але інструкція виду “об’єкт—виклик-методу” (аналог “підмет-присудок”), приклади якої розглядалися, є основною в сучасному програмуванні.

Ознайомившись зі старими мовами програмування, можна помітити, що їх інструкції менше схожі на людську мову. Тенденція мов програмування до набування схожості із людськими мовами з’явилася ще на витоках їхньої історії — спочатку у мовах програмування просто з’явилися короткі слова-команди, що задавали розміщення інформації у комп’ютерній пам’яті, а вже пізніше вони набули сучасного наочного вигляду. Можна зробити висновок, що розвиток мов програмування і надалі буде йти шляхом підвищення легкості сприйняття коду за рахунок схожості із людськими мовами.

Список ітератури

1. portquest2016. Сравнение ЯП с естественными языками [Электронный ресурс] / portquest2016. – 2016. – Режим доступа до ресурсу: <https://www.linux.org.ru/forum/talks/12998355/page1>.
2. Губайловський В. Языки программирования и живые языки: сходства и различия [Электронный ресурс] / В. Губайловський, О. Костинський // Радіо Свобода. – 2004. – Режим доступа до ресурсу: <https://www.svoboda.org/a/24203650.html>.
3. What are the fundamental differences between a human language and a computer language? [Электронный ресурс] / [A. Cheng, T. Serviam, G. Kemnitz та ін.]. – 2014. – Режим доступа до ресурсу: <https://www.quora.com/What-are-the-fundamental-differences-between-a-human-language-and-a-computer-language>.
4. World Languages Vs. Programming Languages [Электронный ресурс] // Language Magazine. – 2015. – Режим доступа до ресурсу: <https://www.languagemagazine.com/2015/10/21/world-languages-vs-programming-languages/>.
5. Kumar R. Natural vs programming languages [Электронный ресурс] / Rajesh Kumar. – 2012. – Режим доступа до ресурсу: <http://meetrajesh.com/posts/natural-vs-programming-languages.html>.
6. ThreeFX. Any difference between natural and programming languages? [Электронный ресурс] / ThreeFX. – 2014. – Режим доступа до ресурсу: <https://linguistics.stackexchange.com/questions/8612/any-difference-between-natural-and-programming-languages>.
7. Udden J. What is the similarity between a natural languages and programming languages? [Электронный ресурс] / J. Udden, H. Hammarström, R. Jansen – Режим доступа до ресурсу: <https://www.mpi.nl/q-a/questions-and-answers/wat-zijn-de-overeenkomsten-en-verschillen-tussen-natuurlijke-talen-en-programmeertalen>.
8. Орлов С. А. Исходное определение языка программирования / Сергей Александрович Орлов // Теория и практика языков программирования. Учебник для вузов. 2-е изд. Стандарт 3-го поколения / Сергей Александрович Орлов. – СПб: Питер, 2018. – С. 26–27.
9. Кауфман В. Язык программирования как знаковая система / Виталий Кауфман // Языки программирования. Концепции и принципы / Виталий Кауфман., 2018. – С. 37–38.

Iliia Ukhov

HUMAN LANGUAGES AND PROGRAMMING LANGUAGES

Comparison of common languages and programming languages is not an urgent global problem. A man sufficiently uses common language for thousands of years and also sufficiently uses programming languages for last little more than half of century. And man will still use both kinds of languages no matter how similar or different they are.

But programming languages are pretty young, and new programming languages make their appearance every year. Every new programming language has its own unique features, its own set of advantages and shortcuts. Some of these new languages appear to be noticeably useful in some area of informational technologies, so they become popular. So, mankind manages to continue its various experiments with programming languages. And it can be possible to use experience of common languages to make new better programming languages.

The problem was previously mentioned by some anonymous users of internet forums as well as by some professional programmers. Short articles can be found on some popular

internet-magazines as well as on some universities' websites. Finally, some theoretical information about structure of programming language as it seen by science can be found in books about programming.

This article describes some visual similarities using simple examples of texts, written in several object-oriented programming languages. It is clearly seen that common and programming languages both have similar syntactic and semantic units, and in both kinds of languages there are special kinds of units to describe entities as well as processes. Also, in both kinds of languages semantically identical texts can be written in different forms: shorter brief form or longer and somewhat more formal-looking.

In result of this little research, it was figured out that new programming languages generally appear to be more similar to common languages than older ones. This trend started at very beginning of programming languages' history and may remain further.

Keywords: language, programming, comparison.

References

1. portquest2016 (2001), "Comparison of PL and common languages", available at : <https://www.linux.org.ru/forum/talks/12998355/page1> (accessed 4 December 2018).
2. Gubaylovskiy V. and Kostinsky A. (2004), "Programming languages and live languages: similarities and differences", available at : <https://www.svoboda.org/a/24203650.html> (accessed 4 December 2018).
3. Cheng A., Serviam .T, Kemnitz G., Reddy R., Vö T., Kearsley L. and Cha M. (2014), "What are the fundamental differences between a human language and a computer language?", available at : <https://www.quora.com/What-are-the-fundamental-differences-between-a-human-language-and-a-computer-language> (accessed 4 December 2018).
4. Language Magazine (2015), "World Languages Vs. Programming Languages", available at : <https://www.languagemagazine.com/2015/10/21/world-languages-vs-programming-languages/> (accessed 4 December 2018).
5. Kumar R. (2012), "Natural vs programming languages", available at : <http://meetrajesh.com/posts/natural-vs-programming-languages.html> (accessed 4 December 2018).
6. ThreeFX (2014), "Any difference between natural and programming languages?", available at : <https://linguistics.stackexchange.com/questions/8612/any-difference-between-natural-and-programming-languages> (accessed 4 December 2018).
7. Udden J., Hammarström H. and Janssen. R "What is the similarity between a natural languages and programming languages?" available at : <https://www.mpi.nl/q-a/questions-and-answers/wat-zijn-de-overeenkomsten-en-verschillen-tussen-natuurlijke-talen-en-programmeertalen> (accessed 4 December 2018).
8. Orlov S. (2018), Theory and practice of programming languages. Manual for high schools, 2nd ed, Piter, St. Petersburg, pp. 26–27.
9. Kaufman V. (2011), Programming languages. Concepts and principles, Litres, Moscow, pp 37–38.

Стаття надійшла до редакції 30.06.2018