

В. Дз. Ляўчук, П. У. Бычкоў, А. В. Варуеў
Гомельскі дзяржаўны ўніверсітэт імя Ф. Скарыны

АЛЬТЭРНАТЫВЫ ВЫБАРУ ПЕРШАГА ІНСТРУМЕНТА ДЛЯ НАВУЧАННЯ ПРАГРАМАВАННЮ

У дадзеным артыкуле ў якасці першага інструменту для навучання праграмаванню разглядаюцца выключна мовы праграмавання. Розныя канструктары праграм кшталту Scratch будуць пакінуты па-за ўвагай. На наш суб'ектыўны погляд, іх роля ў першаю чаргу маркетынгавая і матывацыйная, а потым ужо навучальная і метадычная.

Правільны выбар мовы праграмавання, якая найлепш пасуе для фармавання ведаў і навыкаў прафесіі праграміста, дазваляе пачаткоўцу хутка прагрэсаваць. Пры разглядзе пытання пра першую мову варта ўлічваць шэраг фактараў. Яны абмяркоўваюцца ў дадзеным артыкуле.

Варта ўлічваць ўстойлівасць мовы і інструментальных сродкаў распрацоўкі прыкладанняў на ёй. Калі мова ўсё яшчэ развіваецца, то цалкам магчыма, што яе моўныя канструкцыі могуць быць удакладнены і зменены ў будучых версіях. Яшчэ важна, ці пераносіцца мова на розныя платформы, ці створаная падрабязная даведачная служба. Як прыклад, даводзіцца сутыкацца з пунктам гледжання, што мова праграмавання Python, якая мае ўжо больш за дваццацігадовую гісторыю, дазваляе пачаткоўцам праз лічаную колькасць заняткаў ствараць складаныя прыкладанні. Аднак варта ўлічваць, што гэтая мова падтрымлівае дзве розныя версіі, якія несумяшчальныя адна з другой. Нават калі пагадзіцца з высновай аб трывіяльнасці стварэння складаных прыкладанняў, то наяўнасць сінтаксічных адрозненняў у адной і той жа праграме можа выклікаць моцнае расчараванне ў пачаткоўца-праграміста. Яму прыйдзецца перапісваць свае праграмы толькі таму, што на другой платформе або ў іншай дыстрыбуцыі ўстаноўлены несумяшчальныя бібліятэчныя модулі.

Нішавыя вобласці ёсць у кожнай мовы праграмавання. Так, JavaScript паўстаў з патрэбы праграмаваць вэб-сцэнары на баку кліента. Акрамя таго, вэб-праграміст павінен ведаць HTML (мова гіпертэкставай разметкі) і CSS (каскадныя табліцы стыляў), якія з'яўляюцца кантэйнерамі для сцэнарыяў. Але HTML і CSS не ўваходзяць у навучальныя праграмы для пачаткоўцаў. Нельга патрабаваць ад пачаткоўца разумення канцэпцыі кліент-сервернага праграмавання. Гэта задача спецыяльных дысцыплін. Як след трэба звярнуць увагу на

мовы праграмавання, не звязаныя напрамую з вэб, перш за ўсё на Сі і Паскаль.

Сі з'яўляецца магутнай мовай, якая была напісана для аперацыйнай сістэмы Unix. Яна шырока выкарыстоўваецца ў розных галінах: ад сістэмнага праграмавання да напісання інфармацыйных сістэм. У апошняй галіне вельмі трывалыя пазіцыі былі ў мовы Паскаль. З шырокім распаўсюджваннем асяроддзяў распрацоўкі прыкладанняў спачатку Borland Pascal, а затым Delphi гэтая мова ў канцы мінулага стагоддзя на постсавецкай прасторы была безальтэрнатыўным кандыдатам у якасці інструмента рэалізацыі прыкладных праектаў. Цалкам лагічна, што Сі і Паскаль былі выкарыстаныя на працягу больш чым дваццаці гадоў у якасці першых моў праграмавання ў каледжах, інстытутах і універсітэтах.

Абедзве мовы, як Паскаль, так і Сі, маюць дзве версіі: працэдурную і аб'ектна-арыентаваную. Працэдурнае праграмаванне вучыць студэнтаў вырашаць кампутарныя праблемы, падзяліўшы іх рашэнне на комплекс падзадач, якія афармляюцца ў выглядзе працэдур і функцый. Гэта вельмі натуральны працэс для пачаткоўца. Таму рэкамендуецца пачынаць навучанне праграмаванню з працэдурных моў, такіх як Паскаль ці Сі. Гэта трывіяльны шлях найменшага супраціву. Асабліва, калі ўлічваць наяўнасць у аб'ектна-арыентаваным праграмаванні абстрактных тыпаў дадзеных, канцэпцый інкапсуляцыі, успадкавання і палімарфізму.

Спрабуючы адразу даносіць да пачаткоўца абстрактны падмурак, можна зусім адбіць жаданне стаць праграмістам. Гэтую выснову падвяргаюць сумневу. Але ў тых вышэйшых і сярэдніх спецыяльных навучальных установах, дзе пачыналі навучанне з C++ або Object Pascal ў якасці першай мовы праграмавання, значных дасягненняў не назіраецца. Нават нягледзячы на тое, што гэтыя мовы з'яўляюцца прамымі пацярэннямі працэдурных моў Сі і Паскаль.

Вельмі шырокае распаўсюджванне ў апошніх два дзесяцігоддзі атрымала мова праграмавання Java. Яна шмат у чым успадкоўвае сінтаксіс моў Сі і C++. Педагагічная супольнасць падзялілася ў стаўленні да яго на дзве няроўныя групы з пункту гледжання першай мовы праграмавання. Меншасць сцвярджае, што гэтая мова з'яўляецца яшчэ больш складанай, бо першапачаткова мяркуе, што будучы праграміст ужо ведае пра аб'ектна-арыентаваную парадыгму. Яны пазіцыянуюць Java, як «Сі з класамі». Гэтага ўжо дастаткова, каб натуральным чынам зыходзіць ад працэдурнага падыходу.

Акрамя таго, платформа Java ўключае ў сябе на самых ранніх этапах знаёмства з шэрагам тэхналагічных сродкаў, адны абрэвіятуры якіх могуць у пачаткоўца адбіць жаданне праграмаваць: JVM, JDK, JRE, JSE і да т.п. На гэта апаненты пярэчаць, што Java праектавалася як інструмент, пазбаўлены ад шэрагу цяжкіх ў выкарыстанні механізмаў C++: множнае ўспадкаванне, прамы доступ да кіравання памяццю, вызваленне непатрэбных дадзеных, перагрузка аперацый і да т.п.

У аргументацыі другой групы спецыялістаў ёсць пазітыўны сэнс. Працэдурны падыход у цяперашні час не выкарыстоўваецца ў прамысловым праграмаванні. Глобальнае распаўсюджванне атрымала аб'ектна-арыентаваная тэхналогія на платформе Java. Пошук метадыкі, якая дазволіць пачынаць навучанне, выкарыстоўваючы інструментальныя сродкі дадзенай платформы, з'яўляецца перспектыўным напрамкам у галіне падрыхтоўкі шырока запатрабаваных ІТ-кадраў.

Машынныя мовы дазваляюць праграмісту вырашаць задачу на вельмі нізкім узроўні, маючы справу з рэгістрамі і ячэйкамі памяці з дапамогай простых, але шматлікіх інструкцый. Затым з'явіліся мовы высокага ўзроўню, для якіх ствараюцца спецыяльныя праграмы, званыя кампілятарамі, якія пераўтвараюць чытабельны код у машынныя інструкцыі. Праграма, напісаная на мове высокага ўзроўню, павінна прайсці праз шэраг этапаў апрацоўкі, перш чым яна можа выканацца на кампутары. Гэта этапы кампіляцыі і кампануюкі. Калі кампутарная праграма ствараецца па дадзенай схеме, то выкарыстаная мова праграмавання адносіцца да кампіляванага тыпу.

Пачатковец-праграміст большую частку свайго часу на ранніх стадыях навучання марнуе на выпраўленне памылак і ліквідацыю папярэджаняў кампілятара. Праз выпраўленне сінтаксічных памылак праграмавання забяспечваецца эфект паскарэння ў вывучэнні мовы. Аднак кампілятар не будзе ўказваць памылкі логікі праграміста.

Мовы сцэнарнага тыпу былі створаны, каб прапусціць этапы кампіляцыі і кампануюкі праграмы і непасрэдна выконваць напісаны код. Яны таксама адносяцца да моў высокага ўзроўню. На жаль, дастатковай універсальнасцю яны не валодаюць, абмяжоўваючыся ў платформе, праблемнай вобласці, працоўных структурах дадзеных і да т.п. Варта ўлічваць таксама дадатковыя выдаткі рэсурсаў кампутара на інтэрпрэтацыю чытабельных канструкцый і пераўтварэнне іх у рэальным часе ў машынныя інструкцыі.

Распрацоўшчыкі высокаўзроўневых моў, такіх як Сі, Паскаль, як правіла, падтрымліваюць больш шырокі спектр тыпаў дадзеных, чым мовы сцэнарыяў. Аднак найбольшая прадукцыйнасць па кіраванні абстрактнымі структурамі дадзеных забяспечваецца ў аб'ектна-арыентаваных мовах. Object Pascal, C++, Java падтрымліваюць на ўзроўні бібліятэк набор класаў, якія хутка і эфектыўна апрацоўваюць вялікія аб'ёмы дадзеных. Большасць сцэнарных моў падтрымліваюць простыя тыпы дадзеных і, такім чынам, не пасуюць для складанай апрацоўкі.

Варта адзначыць, што распрацоўшчыкі гэтых моў выразна разумеюць дадзены недахоп і ўводзяць складаныя сінтаксічныя канструкцыі, якія дазваляюць кіраваць калекцыямі. Але пры гэтым прыходзіцца расплачвацца хуткасцю інтэрпрэтацыі праграм і гнуткасцю рэалізацыі алгарытмаў.

Такім чынам, неабходна старанна і ўдумліва падыходзіць да выбару першай мовы праграмавання. Наша рэкамендацыя складаецца ў тым, што першая мова павінна:

- падтрымліваць простую інтуітыўна зразумелую працэдурную парадыгму праграмавання,
- быць стандартызаванай і гарантавана ўстойлівай да новых трэндаў у ІТ-галіне,
- мець набор распаўсюджаных бібліятэк, якія дазваляюць вырашаць шырокі спектр кананічных задач,
- з'яўляцца незалежна адносна аперацыйнай платформы,
- не абмяжоўвацца сферай канкрэтнай вобласці (напрыклад, толькі для вэб-праграмавання),
- трымаць у сваёй сінтаксічнай аснове натуральна інтэрпрэтаваныя чалавекам канструкцыі, прычым пажадана, каб яны з'яўляліся шаблонамі для стварэння спецыфічных для некаторай канкрэтнай галіны сцэнарных моў інтэрпрэтаванага тыпу.

З нашага суб'ектыўнага пункту гледжання, мовы Сі і Паскаль адпавядаюць дадзеным крытэрам, прычым мова Сі ў большай ступені. Аднак яна больш цяжкая для пачаткоўцаў, чым Паскаль, так як патрабуе на парадак больш намаганняў па кіраванні апэратыўнай памяццю. Паколькі аб'ектна-арыентаваны падыход у прамысловым праграмаванні вяршэнствуе, то далей неабходна вывучыць альбо C++, альбо Java. З чатырох магчымых камбінацый (Сі і C++, Сі і Java, Паскаль і C++, Паскаль і Java) найбольш эфектыўны старт для пачаткоўца дасягаецца пры выбары апошняга варыянта – Паскаль і Java.