

ЯКУ МОВУ ПРОГРАМУВАННЯ ВИВЧАТИ У ШКОЛІ

Продовження, початок у №7 за 2013 рік

Глинський Ярослав Миколайович,

*канд. фіз.-мат. наук, доцент національного університету
«Львівська політехніка».*

Палюшок Лілія Володимирівна,

*завідувачка кабінетом інформаційних технологій навчання
Львівського обласного інституту післядипломної педагогічної
освіти.*



Інформатика в середньому навчальному закладі належить до групи технологічних дисциплін з інформаційно-технологічним підходом до визначення мети, змісту і методики навчання, з яскраво вираженими засобами, спрямованими на здобуття учнями практичних компетенцій, і слабо вираженими засобами, які можуть забезпечити вивчення фундаментальних тем інформатики, що сприяють розвитку основних форм мислення. Джерелом фундаментальних тем у курсі шкільної інформатики є розділ, що стосується питань моделювання, алгоритмізації та програмування.

Застарілі підходи до трактування традиційного розділу «Алгоритмізація та програмування» привели до зменшення його ролі, вагомості й місця в курсі шкільної інформатики і до намагань вилучити його з навчальних програм, замінивши традиційний математико-алгоритмічний підхід до вивчення інформатики на інформаційно-технологічний (інший термін-сленг — «користувацький підхід»).

Зважаючи на те, що майже в кожній сім'ї є чи скоро буде персональний комп'ютер, «користувацький підхід» поступово втрачає актуальність. Наявність комп'ютера в особистому користуванні передбачає володіння певними користувацькими навичками, здобувати чи удосконалювати які в школі чи у вищому навчальному закладі в майбутньому не матиме сенсу.

Нині актуальними є намагання удосконалити методику навчання класичного розділу інформатики «Алгоритмізація та програмування».

Таке удосконалення можливе у декількох напрямках, і тут розглянемо лише деякі з них. Передусім варто розпрощатися зі старими підходами, які зводять навчання алгоритмізації та програмування до вивчення синтаксису і семантики мови Паскаль у середовищі Turbo Pascal, Free Pascal, Object Pascal і з підходами, коли застарілі дидактичні засоби й методи навчання переносяться в сучасні середовища об'єктно-орієнтованого програмування і його різновиди — середовища візуального програмування. Усвідомлення об'єктної моделі світу є важливим фактором формування у суб'єкта навчання гносеологічних основ пізнання. Практично будь-яка робота учня з візуальним середовищем програмування пов'язана з такими поняттями як: об'єкт, властивості об'єкта, множина значень властивості об'єкта, методи об'єкта. Моделю-

вання значною мірою вже наявне на рівні ознайомлення з візуальним середовищем програмування. Замість розгляду задач про найбільший спільний дільник, прості числа й методи впорядкування масивів навчання учнів потрібно базувати на зовсім інших задачах, наприклад, які подані в [1], і максимально експлуатувати переваги візуального програмування щодо креативної і захопливої візуалізації результатів роботи, чого раніше не можна було досягнути засобами процедурного програмування.

У результаті порівняльної апробації різних підходів до вибору мови програмування ми переконалися, що гіпотеза про доцільність застосування Visual Basic (VB) у навчальному процесі є правильною. Пам'ятаючи, що програмістів ми не готуємо, і не заперечуючи методичних досягнень, пов'язаних із застосуванням середовища Delphi і Lazarus, стверджуємо, що навчання на базі Visual Basic є простіше, зручніше, ефективніше.

Як відомо, шкільні програми не прив'язують навчальний процес до конкретної мови програмування. Це дає змогу вчителям вибирати мову, яку вивчатимуть учні, з декількох альтернатив. Але цей вибір лише теоретичний, бо єдиний підручник, що масово надійшов у школи, містить відомості про мову Pascal і середовище Delphi. Те, що в школі на сьогоднішній день перевага надається мові Паскаль, є результатом системного впровадження цієї мови у 90-х роках, це приклад інерційності в освіті, прив'язування мови програмування до умов четвертого етапу олімпіади з інформатики, відсутності можливості у вчителів і учнів придбати відповідну навчальну літературу.

З-поміж усіх середовищ, де реалізовано Visual Basic, перевагу варто надавати середовищу VBA (Visual Basic for Applications). Це середовище у межах двох класичних парадигм (процедурно-орієнтованого і об'єктно-орієнтованого програмування) з найменшими затратами дає змогу реалізувати основні способи програмування: процедурне, візуальне, об'єктне. Головна мета навчання — реалізація засобами VBA традиційних обчислювальних алгоритмів у візуальному середовищі розробки проектів і необчислювальних алгоритмів, де реалізуються основні інформаційні процеси: створення, зберігання, візуалізація, пересилання і пошук даних (простих, складених і мультимедійних).

Головною і незаперечною перевагою використання VBA — це можливість організації самостійної роботи учнів над програмами і проектами у домашніх умовах, чого не можна забезпечити, базуючись на Delphi чи Lazarus. Адже VBA, як середовище програмування, вбудоване в усі офісні програми, які, зазвичай, є на всіх персональних комп'ютерах. Досвід показав, що намагання вчителів переконати учнів встановити на своїх домашніх комп'ютерах Delphi чи Lazarus не дають позитивного результату.

Питання щодо організації олімпіад з програмування із застосуванням VB-засобів може бути вирішене, якщо для цього буде добра воля і бажання осіб, які займаються організацією олімпіад.

На відеосервісі YouTube на каналі hlynsky1 можна ознайомитися з навчальним програмним відеозабезпеченням, яке призначене для навчання основ алгоритмізації і розробки проектів у середовищі VBA.

Відеофільми дали змогу спростити й автоматизувати процес навчання. Оскільки тематика відеофільмів пов'язана з прагматичними питаннями роботи в програмних середовищах, вдалося вивільнити час для висвітлення вчителем теоретичних питань алгоритмізації і програмування, що спрямовано на підвищення рівня фундаментальності навчання інформатики в школі й дало значний педагогічний ефект.

З метою надання методичної допомоги вчителям інформатики у Львівському обласному інституті післядипломної педагогічної освіти, починаючи з вересня 2011 року, на курсах підвищення кваліфікації в рамках інваріантного спецкурсу «Актуальні питання викладання інформатики» проводяться практичні заняття на тему «Методика викладання теми «Основи алгоритмізації та програмування» в середній школі». На цих заняттях вчителі інформатики ознайомлюються із середовищем VBA і набувають навичок створення проектів для розв'язання типових задач і ефектної візуалізації результатів шляхом розробки відповідного інтерфейсу користувача.

У результаті аналізу результатів опитування вчителів можна зробити висновок про те, що більшість з них вважають середовище VBA простішим і зручнішим для навчання учнів, ніж середовище Delphi. Ма-

ючи досвід роботи у двох середовищах, вони надають перевагу навчанню на базі мови програмування Visual Basic. Єдиний недолік цього середовища полягає в тому, що воно не дає змоги зберігати проекти у форматі виконуваних файлів. Але для навчального процесу це несуттєво, а у разі прагматичних потреб є змога перенести вже створені VBA-проекти в середовища професійного рівня на кшталт Visual Basic 2010 Express чи Visual Basic у Visual Studio 2008, які поширюються безкоштовно.

Для навчання за програмами, які передбачають поглиблене оволодіння учнями навичками і знаннями з алгоритмізації та програмування, Visual Basic — це найкращий вибір першої й основної мови, який може забезпечити в подальшому ефективно ознайомлення чи вивчення мов C# чи Java або їх різновидів у разі потреби.

Висновки. Проведені педагогічні дослідження й аналіз їх результатів дають нам змогу рекомендувати мову Visual Basic для використання у навчальному процесі під час вивчення розділу алгоритмізація та програмування в курсі інформатики академічного рівня в 11-му класі, у курсах за вибором, а також у курсах поглибленого вивчення інформатики у 8–9-их чи 8–11-их класах. Якщо в [2] формулювалась гіпотеза про доцільність використання цієї мови (у середовищах VBA чи VB 2010 Express) в навчальному процесі, то зараз ми впевнені, що ця гіпотеза є правильною, тобто використання мови Visual Basic і реалізація удосконаленої методики навчання забезпечать більш ефективну підготовку учнів з даного розділу інформатики й ознайомить їх з корисним інструментарієм для дослідження інформаційних моделей об'єктів, процесів і явищ навколишнього світу.

Література

1. Глинський Я.М. Інформатика. Основи алгоритмізації і програмування мовою Visual Basic: навч. посібн. — Львів : СПД Глинський, 2011. — 272 с.
2. Глинський Я.М., Ряжська В.А. Бейсик повертається // Теорія та методика навчання математики, фізики, інформатики: збірник наукових праць. — Вип. IX.: в 3-х томах. — Кривий Ріг : Вид. відділ НметАУ, 2010. — Т. 3 : Теорія і методика навчання інформатики. — С. 443–449.



Воропай Юрий Иванович,

учитель информатики гимназии №28 г. Запорожья, высшая категория, учитель-методист, Заслуженный учитель Украины.

В контексте изучения темы «Алгоритмизация и программирование», с моей точки зрения, следует работать в среде программирования Delphi. На первом этапе при изучении алгоритмических конструкций и методов программирования необходимо использовать среду в консольном режиме. Старые версии языка Pascal (Turbo, Free и др.) морально устарели и совместимы не со всеми современными операционными системами.

На втором этапе при разработке проектов (объектно-ориентированное визуальное программирование) используются библиотеки объектов Delphi. Никакая другая визуальная среда не дает такого

многообразия объектов и интуитивно понятного алгоритма их использования. Это дает возможность школьнику реализовать в проекте не только функциональные задачи, но и разработать очень качественный интерфейс.

При изучении тем, связанных с информационными технологиями, на мой взгляд, следует использовать язык программирования VBA в том случае, если курс изучают с использованием MS Office. Это объективно обосновывается тем, что он является составной частью данной системы.



Петрович Сергій Драганович,

кандидат педагогічних наук, викладач вищої категорії, викладач-методист Вінницького коледжу Національного університету харчових технологій.



На наш погляд, програмування для школярів має бути захоплюючим процесом. У сьогоденні світу інформаційних технологій учні мають усвідомлювати основні принципи програмування. Однак часто буває ситуація, коли учні не зацікавлені у вивченні програмування. Причини можуть бути різні, але найчастіше — їм це просто не цікаво. В учнів відсутня мотивація до вивчення мов програмування. А інколи можна почути відповідь: «мови програмування мені не потрібні, адже я не буду програмістом». Саме у цьому, на нашу думку, полягає педагогічна майстерність учителя інформатики, щоб використавши відповідні методики, правильно обравши мову програмування для вивчення, зацікавити учнів у вивченні програмування. Процес вивчення програмування має бути цікавим, корисним та зрозумілим для учнів [2].

Визначальним для вибору мови і середовища програмування як засобів навчання є методична підтримка (методичне забезпечення) навчального процесу. Сюди варто віднести наявність, кількість, якість, а для деяких категорій, і доступність:

- підручників і збірників задач;
- методик використання і методичної літератури;
- компетентних педагогів та методистів;
- навчального програмного забезпечення;
- представлення в глобальній мережі Інтернет (як загальноінформаційне, так і в інтерактивних сервісах і службах);
- прикладів і позитивних результатів упровадження.

Ці передумови сильно взаємопов'язані і взаємообумовлені.

Середовища для програмування відповідною мовою функціонують під управлінням тих чи інших операційних систем. Використання операційних систем, як і середовища програмування потребує від комп'ютерної техніки певних апаратних характеристик. І нарешті, для використання техніки в навчально-виховному процесі учителю інформатики і лаборанту комп'ютерного класу слід мати відповідний рівень компетентностей, які стосуються використання комп'ютерного обладнання і програмного забезпечення.

На сьогоднішній день існує великий вибір мов програмування — починаючи від найпростіших (тих, на яких навчаються діти молодших класів), і закінчуючи такими потужними інструментами розробки, як C++, C#, OpenGL, Java і т. д.

Перерахуємо деякі з них.

1. Мова програмування C#. Компанія Microsoft розробила мову C# в кінці 1990-х років. Автором мови є Андерс Хейлсберг. Мова C# є складовою програмної платформи Microsoft.NET.

У середині 2000 року була випущена альфа-версія мови. Хейлсберг, так само як автори мов C++ і Java, не створював нових наборів команд і правил побудови синтаксису, а використав як фундамент існуючі мови, зосередився на покращеннях й інноваціях. C# має багато спільного з широко використовуваними і популярними мовами програмування C, C++, Java та

PHP. Нині практично всі професійні програмісти знають ці мови, тому перехід до C# відбувається без особливих труднощів.

Поряд із цим, знання, набуті у процесі оволодіння мовою C#, безперечно знадобляться під час опанування іншими мовами програмування.

В останні роки мова C# стала активно використовуватись для навчання програмування. Існує надзвичайно багато on-line сервісів, що використовуються для надання допомоги всім бажаючим у вивченні цієї мови програмування.

2. LightBot — середовище програмування для школярів початкових класів (можна використовувати з 5–6 років). Необхідно запрограмувати рух «віртуального» робота. Незважаючи на простоту, за допомогою LightBot можна не тільки формувати логічне мислення учнів, але й вивчати такі поняття як «підпрограми» і «процедури».

3. RoboMind — просте середовище програмування, яке дозволяє програмувати поведінку «машинки-робота». Тут у доступній формі вивчаються популярні методи програмування та основи штучного інтелекту. Робот може програмуватися на різних мовах.

4. LittleWizzard — середовище програмування для дітей, призначене для вивчення основних елементів програмування в початковій школі. Використовуючи тільки мишку, учні отримують можливість складати програми і вивчати такі поняття, як «змінні», «вирази», «розгалуження», «умови» і «логічні блоки». Кожен елемент мови є інтуїтивно зрозумілим символом.

5. Karel, Karel++, Karel J. Robot — мови для початківців, вони використовуються для складання програм управління «роботом». Karel використовує власну мову програмування, Karel++ — мову програмування C++, Karel J. Robot — версію Karel на Java.

6. Logo — мова, яка була спеціально розроблена з метою навчання дітей програмування. У сучасних реалізаціях Logo віртуальний агент під назвою «черепашка», переміщенням якого можна програмно керувати, використовується для того, щоб зробити програмування привабливим для дітей, зосередити їх увагу на побудові зображень за допомогою «черепашки». Logo створювали, з одного боку, для того, щоб допомогти дітям вивчити основні поняття програмування, а, з іншого, — для їх інтелектуального розвитку у світі, де все більше використовуються інформаційно-комунікаційні технології.

Logo успішно застосовується в різних сферах — від початкового знайомства з комп'ютером і програмуванням у школі до вивчення проблем штучного інтелекту і моделювання екологічних систем в університетах.

7. Scratch — візуальна мова програмування, яка була розроблена Lifelong Kindergarten Group з MIT MediaLab, для навчання програмування дітей.

Основна мета цього проекту — навчання основним поняттям програмування шляхом створення про-

грам-проектів, що містять програмовані об'єкти. Scratch дозволяє створювати інтерактивні додатки шляхом комбінування блоків-команд, використовувати різні графічні об'єкти, зображення, звуки і музику. Завдяки простоті і досить потужним можливостям, навіть початківці можуть створювати власні розробки, тим самим отримуючи мотивацію для подальшого вивчення програмування. Користувачі мови Scratch мають можливість обмінюватись думками у створеному on-line товаристві, членом якого може бути кожен, хто програмує в цьому середовищі.

8. Squeak — сучасна, відкрита, повнофункціональна реалізація середовища і об'єктно орієнтованої мови програмування Smalltalk. Squeak використовується як засіб для створення надзвичайно широкого діапазону проєктів — від мультимедійних додатків і різноманітних освітніх платформ до розробки веб-сайтів. Програмні засоби, розроблені за допомогою Squeak, надзвичайно просто «переносяться» в середовище будь-якої операційної системи, оскільки код програми виконується (інтерпретується) «віртуальною машиною» Squeak. Аналогічна технологія була використана в процесі розробки мови програмування Java.

9. Pascal — одна з найвідоміших мов програмування, яка була розроблена саме з навчальною метою. Починаючи від кінця сімдесятих до дев'яностих років, Pascal був основною мовою програмування, яку використовували учителі в процесі вивчення інформатики (програмування) як у США, так і в Європі. Її використання для навчання програмування і на сьогоднішній день широко поширене, незважаючи на винятково освітній напрямок цієї мови і на зростаючу конкуренцію з боку C, Java і Python.

10. Середовище програмування Robolab. Даний програмний продукт з успіхом можна використовувати для програмування конструкторів-роботів компанії Lego. Це середовище засноване на мові графічного програмування LabVIEW — потужного середовища програмування, яке використовується інженерами й ученими в дослідницьких інститутах і промисловості. LabVIEW — це провідний інструмент для вимірювання і контролю. Він буде корисним у процесі аналізу реальних результатів у біомедицині, космічних дослідженнях, енергетичних дослідженнях і має ще багато сфер використання. Однією з основних переваг середовища Robolab є можливість наочної демонстрації результату в роботі програми шляхом її завантаження в мікропроцесор, який керує роботом. Робот у свою чергу складається з конструктора Lego [3]. Ідея створення цього робота з'явилася в Массачусетському технологічному інституті (США), але дуже швидко поширилась в інші навчальні заклади як наочний навчальний матеріал.

Учень у процесі складання робота опановує основні конструювання, фундаментальні фізичні принципи і розвиває технічне мислення. На етапі програмування і виконання програми учень може відстежити помилки, які, можливо, були допущені у процесі збирання робота і можливі помилки в алгоритмі програми. Це середовище програмування чудово підходить для фахівців практично всіх категорій, оскільки має можливість як «блокового» програмування, так і програмування на таких мовах, як C, C++, Assembler і т. д. Ця особливість обумовлена тим, що

мікроконтролер, що входить в набір LegoMindstorms має універсальну систему команд, до якої, використовуючи трансляцію, можна звести програму, написану на інших мовах програмування.

11. NXT-G графічне середовище програмування. NXT-G — це графічне середовище програмування, розроблене компанією National Instruments для Lego. Написання NXT-G програм дуже схоже на створення блок-схем (рис. 1). Користувач «пише» програму шляхом поєднання блоків, які описують поведінку певного пристрою, наприклад, обертання двигунів. З використанням різних блоків, можна контролювати напрям руху робота, визначити відстань до об'єкта або змінювати рух керованого робота залежно від зміни кольору перешкоди, а також можна відтворити звук і виконати код програми залежно від стану датчиків і т. д.



Рис. 1. Інтерфейс вікна програми NXT-G

Як правило, учні із задоволенням сприймають такий спосіб вивчення інформатики і заняття проходять у творчій і невимушеній атмосфері. До того ж такий підхід «навчання через дію» має додаткові переваги: учні в результаті роботи отримують «готовий виріб» — робота, який здатний рухатись і реагувати на зовнішні перешкоди. Існує ще одна вага переваги вивчення програмування через використання робототехніки — це можливість взяти участь у Всеукраїнській олімпіаді з робототехніки (<http://robotica.in.ua>), яка по суті є відбірною на Всесвітню олімпіаду.

Графічне середовище NXT-G має простий і зрозумілий інтерфейс (рис. 2), і може бути застосоване в процесі вивчення інформатики практично для учнів будь-якого віку. Переваги вивчення програмування

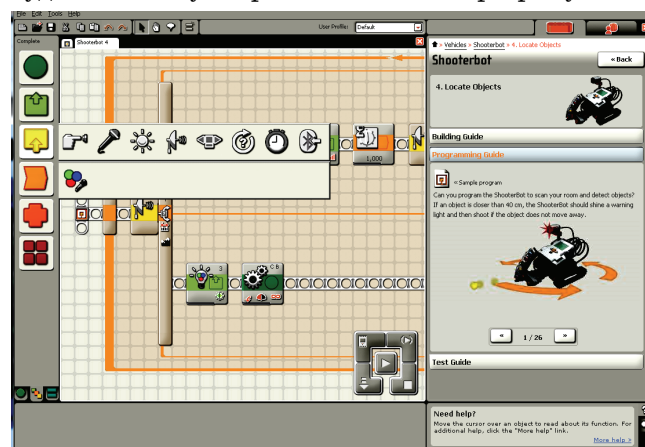


Рис. 2. Приклад програми, розробленої у середовищі графічного програмування NXT-G

у такий спосіб полягають у тому, що учні не просто вивчають інформатику, а стають дослідниками, техніками, розвивають комунікаційні навички, логічне та технічне мислення.

Отже, можна констатувати, що на сьогоднішній день існує велика кількість навчальних мов програмування, кожна з яких володіє своїми перевагами. Однак, яку б мову програмування не обрав учитель інформатики для вивчення у середній школі, він має розуміти, що вибір має ґрунтуватися, перш за все, на пе-

респективних, динамічно розвиваючих напрямках науки і техніки, на розвитку в учнів інженерно-технічного, креативного та логічного мислення.

Література

1. Коломієць А.М. Інформаційна культура вчителя початкових класів : монографія/ А.М. Коломієць. — Вінниця : ВДПУ, 2007. — 379 с.
2. Роботи нового класу [Електронний ресурс]. — Режим доступу до ресурсу: <http://education.lego.com/ru-ru/preschool-and-school/secondary/mindstorms-education-ev3>.



Рудик Олександр Борисович,

кандидат фізико-математичних наук, доцент кафедри методики природничо-математичної освіти і технологій Інституту післядипломної педагогічної освіти Київського університету імені Бориса Грінченка.



Вибір мови програмування, яку використовують для вивчення курсу інформатики, полишено на вчителя. І цей вибір буде зроблено згідно з певною ієрархією цінностей, яких дотримується вчитель. До обговорення питання щодо вибору мови програмування, найпридатнішої для викладання інформатики, природно розглянути загальні питання щодо мети такого вивчення.

Не залежно від того, який предмет природничо-математичного циклу і який рівень вивчення предмету у класі — загальноосвітній, академічний чи профільний (поглиблений до старшої школи), можна зіткнутися з імітацією вивчення. У цьому випадку за мотивацію навчання слугують лише слова навчальної програми. Формально вимоги програми задоволено, стандартні завдання переважна більшість учнів виконує успішно з точки зору неприскіпливого вчителя. Але досить дещо змінити умови завдань, не виходячи за межі опанованого понятійного апарату, і результати будуть іншими, уже не такими втішними.

Проблему сумлінного ставлення вчителя до своїх обов'язків полишимо до іншої нагоди. Звернімо увагу на системні кроки МОН і НАПН України. Мету вивчення кожного предмету сформульовано переважно у термінах відповідної дисципліни. Мету освіти взагалі сформульовано настільки аморфно, що це дає можливість майже довільно розподіляти й перерозподіляти навчальні години. Наприклад, вивчати інформатику по 1 навчальній годині на тиждень, або мати ілюзію щодо вільного вибору мови програмування. Поки це триває, учителі різних предметів грають за своїми правилами у різні ігри, не підсилюючи позитивний вплив один одного.

Свого часу автор пропонував стисло визначити мету освіти як набуття учнями ознак кваліфікованої робочої сили [1]. Дотримуючись цього, він запропонував послідовність [2] вивчення розділів прикладної математики, узгоджену з логічно послідовним викладом математики [3]. Кілька років тому в межах роботи над науковою темою КУ ім. Бориса Грінченка автор скористався нагодою систематизувати своє бачення компетентнісного підходу, спільного для всіх предметів природничо-математичного циклу [4]. Не вдаючись до повного переказу останньої роботи, зауважимо: першими — і щодо часу вироблення, і щодо значимості — названо компетенції щодо мовлення. Врешті-решт, гар-

ний спеціаліст має вміти стисло й переконливо пояснити свої знання і думки. Серед компетенцій щодо опанування алгоритмами виділено таку: спроможність сприйняти й висловити алгоритм. Якщо у початковій і базовій школі вчитель істотно опирається на наочність у поданні матеріалу, то у старшій школі наголос має падати на слово. Заперечення з апелюванням до вищої ефективності сприйняття візуальної інформації у цей час є проявом вчительського популізму й оковамилюванням. У сучасній українській шкільній освіті ці міркування стають істотним гальмом розвитку компетенції щодо мовлення на матеріалі старшої школи. Відомо таке: якщо дитина з 3 до 8 років перебувала поза людським суспільством, то у майбутньому вона вже ніколи не зможе повноцінно опанувати людською мовою. Вимовляти звуки, слова і послідовності слів навчити її можна, а от навчити узгоджувати граматичні конструкції (наприклад, закінчення для різних відмінків) не вдалося ще нікому. Якщо у віці з 3 до 8 років ми говоримо про опанування основами розмовної мови, то довершене опанування мовою (щодо стислості, несуперечливості й чіткості викладу думки) природно здійснити пізніше. Але, мабуть, не пізніше, ніж завершується дозрівання людського організму — до 18 років.

Найкращий спосіб для досягнення цієї мети — розв'язування змістовних завдань на багато кроків до програмної реалізації включно. Найкращий, бо є можливість об'єктивно (незалежно від волі чи старанності окремої людини) оцінити кінцевий результат — програму. Хоча аналогічні спроби можна робити, використовуючи змістове наповнення будь-якого предмету. Навіть богослів'я. Але і за кількістю завдань, і за довершеністю перевірки з її демонстративною об'єктивністю змагатися з алгоритмікою решті предметів зась.

Наразі потрібно пам'ятати застереження Дейкстри про приреченість спроб перетворити програмування на простий і доступний кожному процес. До речі, так само, як і заняття математикою, яку вивчають протягом усіх шкільних років. Колись для успіху достатньо було грамотності й лічби. Зараз мова йде про грамотність у взаємодії з комп'ютерними пристроями й високої культурі формулювання алгоритмів.

Розглянемо проблему вибору мови програмування з цієї точки зору. Ми говоримо про загальну освіту, коли ще не потрібно вибирати мову, зручну для виконання певних завдань. А потрібно, попри набуття певного базового рівня компетентності з усіх дисциплін, максимально розвинути здібності людини щодо набуття кваліфікації (бажано, якомога вищої) і можливої перекваліфікації у майбутньому. Чим ближчою до розмовної мови буде мова програмування, тим швидше буде досягнуто цю мету. Тому природно діяти так:

- до 8-го класу використовувати візуальний засіб програмування Scratch;
- починаючи з 8-го класу, використовувати Pascal;
- в індивідуальному порядку, на заняттях гуртків чи під час вивчення спеціальних курсів окремим учням переходити до вивчення потужнішого (щодо наявності бібліотек) програмного забезпечення — С. Чим вищий рівень олімпіади, тим більша частка носіїв мови С. Причина очевидна: на відміну від носіїв мови Pascal їм непотрібно прописувати деталі алгоритмів, уже програмно реалізовані розробниками ПЗ.

Вибір мов Pascal і С продиктовано використанням цих і лише цих мов на олімпіаді з інформатики (IV етап і Міжнародна). Саме це істотно зменшує кількість мов програмування (до двох!), серед яких можливий вибір у старшій школі. Якби не це, то, на думку автора, поза конкуренцією була би Reduce — платформ-

но незалежна переносна система аналітичних обчислень з можливістю побудови графіків. Вона зацікавить старшокласника і стане у нагоді науковцю.

У школі неможливо і не потрібно вивчати всі можливості мови програмування. Це не потрібно навіть для того, щоб перемогти на предметній олімпіаді. У царині програмування для учня найважливіше своєчасно навчитися у правильному порядку розставляти слова мови програмування з метою отримати програмні реалізації певного кола алгоритмів. Щонайменше всіх тих алгоритмів, з якими стикаються учні під час вивчення природничо-математичних дисциплін. Для цього непотрібно знати багато службових слів чи назв процедур і функцій. Але потрібно навчитися так висловлюватися розмовною мовою, щоб швидко й легко перекладати свої висловлювання мовою програмування. І тут у нагоді стане саме Pascal або схожа за синтаксисом мова.

Висновки

1. Вибір мови програмування однозначно визначений завданнями щодо набуття компетентності щодо мовлення й опанування алгоритмів. Наразі такою мовою програмування, починаючи з 8-го класу, може бути лише Pascal.

2. Основну увагу потрібно приділяти змістовому наповненню задач і їхній складності, а не вивченню структур мови програмування.



Мотурнак Євген Володимирович,

учитель інформатики НВК №100 м. Дніпропетровська, Заслужений вчитель України.

Дивно чи ні, але якою б справою ми не займалися, на що б не організували інших, свідомо чи не свідомо, над нами вирує одне запитання. Навіщо? Якщо ми не отримуємо відповідь на це запитання, то одразу наші зусилля видаються марними, продуктивність праці падає, а то і взагалі процес зупиняється. Так і з програмуванням. Тож почнемо з головного запитання.

Навіщо вчити програмування

Хто з нас, учителів інформатики не чув цього запитання:

- А навіщо ми вчимо програмування? Я ніколи програмістом не буду!
- І я.
- Я також не буду!
- Ні, звичайно такі щасливці є. Але як відповідає дітям переважна більшість учителів?
- Так написано в програмі!
- Це не обговорюється!
- А чому ви не питаєте, чому ви вчите фізику і хімію?

Це не відповіді, це уникнення відповіді. А якщо немає відповіді, то немає і мотивації вчитися. Для школяра «написано в програмі» — це порожній звук. Мотивація для навчання може бути двох типів: або «переваги над іншими, які я можу отримати просто зараз», або «переваги, які я матиму над іншими у самостійному дорослому житті». Є ще «переваги, які я матиму над собою», але це дуже складна мотивація для переважної більшості учнів.



Так дійсно, що відповідати? Ми повинні зрозуміти для себе, для чого саме ми навчаємо програмуванню всіх. Програмування суті планування роботи комп'ютера з виконання поставленого завдання. Тож, навчаючи програмуванню, ми навчаємо плануванню діяльності. Комп'ютер у даному контексті виступає як ідеальний виконавець. Навчившись планувати діяльність ідеального виконавця, людина зможе набагато впевненіше навчитися планувати діяльність далеко неідеальних виконавців — себе, інших людей, підлеглих у майбутньому. Не всі хочуть бути програмістами? Проте переважна більшість бажає бути керівниками. От програмування і вчить вправному керуванню.

Підведемо короткі підсумки щодо мети вивчення програмування у школі.

Програмування для учня

- навчання плануванню діяльності, розвиток алгоритмічного мислення;
- основа для отримання престижної професії;
- можливість творити власні світи;
- програмування — це цікаво.

Програмування для школи

- участь учнів у широкому спектрі інтелектуальних змагань;

- створення навчального програмного забезпечення в школі;
- учнівська ІТ-служба.

Програмування для суспільства

- підвищення ефективності роботи громадянина;
- основа для розвитку ІТ-галузі — економічного козиря України.

Критерії вибору середовища програмування

З огляду на зазначену мету вивчення програмування у школі визначимо критерії вибору середовища програмування.

Доступність ліцензійної версії

Не будемо забувати, що школа повинна працювати в правовому полі, і на будь-яке програмне забезпечення, яке ми обираємо, школа повинна мати відповідну ліцензію. Будемо реалістами. Держава зараз не виділяє коштів на закупку програмного забезпечення, батьківські фонди і так перевантажені, тому зробимо очевидний висновок і звернемося до вільно-розповсюдженого ПЗ.

Простота вивчення

Навчання програмування повинно базуватись на загальних педагогічних принципах: «від простого до складного», «від відомого до невідомого», «від конкретного до абстрактного». Тож середовище програмування повинно забезпечувати можливість виконання всіх цих принципів. Деякі середовища і мови програмування є досить складними для новачків. Навіть написання простої програми в них вимагає певної кількості дій з налаштування середовища. І може статися так, що середовище програмування, яке забезпечує високу ефективність роботи професіоналів, зовсім не підходить для початкового вивчення в школі.

Юні програмісти схильні до великої кількості помилок, і це добре. Саме на помилках вони вчаться. Але помилку це треба побачити, знайти. Тому ще один критерій вибору середовища та мови програмування — це легка у користуванні система відлагодження програми, мінімізація вірогідності появи неявних помилок, пов'язаних, наприклад, з використанням неправильних типів даних.

Попит на ринку

Іноді, обираючи ту чи іншу мову програмування, середовище розробки програм, як аргумент використовують попит на відповідних спеціалістів на ринку праці. Але, якщо ми звернемося до мети викладання програмування в школі, ми побачимо, що школа не повинна випускати готових програмістів. У неї інше завдання — розвиток алгоритмічного мислення, навичок планування роботи та закладання основ програмування. Підготовка спеціалістів з конкретних мов програмування, конкретних технологій — це завдання ВНЗ.

Але якщо ми передбачаємо майбутній перехід учня до іншої мови програмування, то є сенс обирати таку мову для початкового вивчення, перехід з якої буде найлегшим до більшості інших мов.

Застосовність для практики

І є ще один критерій для вибору. Якщо планується підготовка учнів до олімпіад з програмування, ту-

рніру юних інформатиків та інших інтелектуальних змагань, слід звернути увагу на ті середовища і мови програмування, які використовуються для виконання завдань у цих змаганнях.

Аналіз середовищ програмування

Тепер розглянемо найбільш поширені в українській школі мови програмування і середовища розробки програм з огляду на визначені нами критерії. Здебільшого нині використовуються такі універсальні мови програмування як Basic, Pascal, C++, C#. Іноді використовуються більш екзотичні для школи мови Java, JavaScript.

Basic і Pascal вважаються найбільш простими для початкового вивчення. Програмування на C++, C#, Java вже для найпростіших програм потребує використання класів. Ці мови є об'єктно-орієнтованими у своїй суті, мають більше можливостей для роботи над реальними проектами. Попри це, більшість ІТ-компаній виставляють вакансії для спеціалістів саме з цих мов програмування. Мова JavaScript цікава тим, що застосовується для веб-програмування, яке теж є досить популярним нині.

Що ж обрати? Незважаючи на популярність і затребуваність C++, C#, Java на ринку праці, складність синтаксису, необхідність роботи з об'єктами вже із самого початку роботи можуть ускладнити сприйняття і без того непростого матеріалу учнями. Нестроге приведення типів, яке притаманне для цих мов збільшує вірогідність появи помилок, які початківцю досить складно знайти. З іншого боку, якщо пройти ці труднощі початкового навчання, можна буде створювати досить професійні програми. Але скільки відсотків учнів дійсно подолають цей рубіж?

Pascal — застаріла мова програмування. Нині майже не створюється нових проектів з її використанням. Хоча, при цьому, існує ціла низка діючих програмних продуктів, створених за допомогою Delphi, і необхідні спеціалісти, які можуть їх супроводжувати. З іншого боку, розпочати програмувати на Pascal'і дуже просто. Найпростіша програма складається всього з трьох рядків і не потребує введення складних понять. Мабуть найбільшою перевагою цієї мови є той факт, що це мова програмування із строгим приведенням типів. Це означає, що середовище програмування не дозволить використовувати рядкову величину для присвоєння числовій змінній без явної вказівки програміста. Таку помилку можна буде побачити ще на етапі компіляції.

Як показує практика, навички, отримані під час вивчення Pascal'ю дозволяють легко перейти на будь-яку сучасну мову програмування.

Basic — це мова програмування, яка розроблена саме для навчання початківців. Досить легко почати програмувати, як і у випадку з Pascal'ем. Крім того, такі реалізації як VisualBasic.NET дозволяють створювати реальні проекти. Але існує думка, що для тих, хто звик програмувати цією мовою, перехід до інших мов несе певні труднощі. Здебільшого це стосується старих реалізацій.

Якщо йде мова про підготовку до олімпіади з програмування, то слід зауважити, що в більшості регі-

онів України на III-му, обласному етапі вже використовується міжнародна система перевірки розв'язків e-judge. Ця система підтримує лише два компілятори FreePascal та C++. При цьому програми учнів повинні працювати у консольному режимі.

Щодо JavaScript, то синтаксис цієї мови нагадує синтаксис C++ та Java. У найпростішому варіанті не треба окремого середовища програмування. Можна використовувати найпростіший текстовий редактор і браузер для виконання програм. Але в такому випадку з'являється багато проблем з відлагодженням коду.

Є цікава альтернатива. У 2012 році Microsoft анонсувала нове середовище озробки програм TouchDeveloper. В основі лежить мова програмування JavaScript, але головне те, що це середовище надає можливість програмувати без клавіатури, лише за допомогою сенсорних пристроїв. Так, можна програмувати за допомогою смартфона або електронної дошки. Процес створення програми відбувається досить швидко. Використання існуючих бібліотек дає можливість як імітувати черепашку з Лого, так і створювати професійні додатки і публікувати їх в магазині Windows-

Store. Але є і певні проблеми. Так, наприклад, робота з масивами ускладнена тим, що як таких масивів у мові немає. Але є інші структури даних, такі як колекції й об'єкти доступу до даних, які можна використовувати як заміну масивам.

Що стосується конкретних середовищ розробки програм, то їх величезна кількість не дозволить розглянути їх у рамках цієї статті. Крім того, існують різні платформи: Windows, Linux, iOS, під кожен з яких розроблені середовища програмування. Тож, вибір буде залежати від комп'ютерної техніки, встановленої в школі, операційної системи, можливостей фінансування та інших причин, що були зазначені вище.

Висновки

Підводячи підсумки, хочеться згадати відому фразу: «Кадри вирішують все». Усі критерії, які ми розглянули, мають велике значення, але останнє слово залишається за вчителем. Найкращим інструментом викладання для нас буде той, яким ми вправно володіємо. Успіхів вам у оволодінні новими цікавими інструментами!



Бодня Ірина Василівна,

вчитель інформатики ліцею «Лідер» Гуляйпільського району Запорізької області, вчитель вищої категорії, вчитель-методист.

Змістова лінія «Основи алгоритмізації та програмування» є наскрізною для всього курсу «Інформатика» у 5–9-их класах.

Вивчення розділу «Основи алгоритмізації та програмування» розпочинається з 6-го класу в такій послідовності: **Алгоритми та їх виконавці** (7 год.); **Алгоритми з повторенням і розгалуженням** (8 год.); **Алгоритми роботи з величинами** (10 год.); **Табличні величини та алгоритми їх опрацювання** (8 год.).

На першому рівні вивчення (6–7-ий клас) передбачається формування алгоритмічного мислення учнів і розробка алгоритмів у навчальному середовищі виконання, на другому рівні (8–9-ий клас) — ознайомлення з мовою програмування і реалізація алгоритмів у навчальному середовищі програмування.

З моєї точки зору, буде доцільним у 6–7-их класах вивчати візуальне об'єктно-орієнтоване середовище програмування Скретч, так як вивчення теми «Об'єкти та події», складання алгоритмів без використання змінних і констант, які за програмою вивчаються лише у 8-му класі, найкраще реалізувати у цьому середовищі.

У 8–9-их класах пропоную здійснити перехід до вивчення мови програмування Pascal, але з реалізацією програм у середовищі програмування «Pascal ABC», так як у ньому багато концепцій програмування свідомо спрощені, що дозволяє використовувати його на більш ранніх етапах навчання, а саме середовище має графічний інтерфейс.



Петрова Олена Миколаївна,

вчитель інформатики ліцею економіки та інформаційних технологій м. Запоріжжя, вища категорія, вчитель-методист, переможець обласного конкурсу «Вчитель року-2013» в номінації «Інформатика».

Давати програмування в основній школі? — Це питання вже не стоїть. Учителям зі стажем викладання саме програмування зрозуміло, що чим раніше, тим краще. Постає наступне запитання — які мови програмування і які середовища використовувати? Для учнів 5–6-их класів, коли ще сильно превалює візуальна сторона процесу, краще всього використовувати об'єктно-орієнтоване середовище Scratch. У цьому середовищі ми маємо і

розвиток творчості, логіки, і першине сприйняття базових алгоритмічних структур. 7–9-і класи — мова програмування Паскаль і поступове занурення в алгоритми розв'язання базових завдань. Синтаксис мови Паскаль простий для запам'ятовування, але має достатньо засобів для розв'язування класичних завдань.

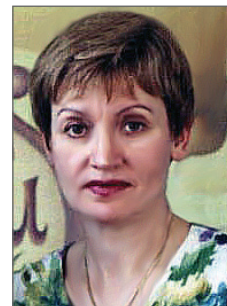


Пасіхов Юрій Якович,

заступник директора, завідувач лабораторії інформаційних та комунікаційних технологій фізико-математичної гімназії №17 м. Вінниці, заслужений учитель України.

Кравець Галина Петрівна,

учитель інформатики Донецького обласного ліцею-інтернату для обдарованих дітей «Ерудит».



Постановка проблеми. Питання, винесене у заголовок статті логічно впливає з відповіді на інше, більш загальне: «А яка інформатика потрібна в школі: користувацька чи «програмістська»? Пошуку відповіді присвячено безліч статей, полум'яних дискусій та наукоподібних обґрунтувань тієї чи іншої точки зору. Свій погляд на проблему автор виклав у статті «Шкільна інформатика: програми, підручники, учителі» [7]. Реалії сьогодення показують, що серед науковців, методистів та «ідеологів від освіти» знайдено певний компроміс у цьому питанні. Попри те, що інформатика у школах України віднесена до навчальної галузі «Технології», у всіх діючих на сьогоднішній день програмах передбачено години на вивчення алгоритмізації та програмування (як на мою думку, замало), а нещодавно затверджена «Програма курсу «ІНФОРМАТИКА» (8–9 класи загальноосвітніх навчальних закладів з поглибленим вивченням інформатики)» [8] приблизно 50% навчального часу відводить саме на цей навчальний напрямок, що варто розглядати, як великий крок уперед.

Отже, основи алгоритмізації та програмування у школі вивчати варто. Як це зробити найкраще? Яку мову програмування обрати? У цій статті зроблено спробу на основі власного 27-річного досвіду викладання курсу у класах як «звичайних», так і «поглиблених» дати відповідь на ці запитання.

Шляхи вирішення проблеми. Один з основних аргументів противників вивчення основ алгоритмізації та програмування (ОАП) у школі вельми прозаїчний: значна частина учнів (інколи — переважаюча) не в змозі засвоїти це непросте мистецтво. «Не всі будуть програмістами, то навіщо всім це вивчати?». Програмістами, дійсно, будуть не всі. Але:

- уже нині відсоток тих випускників шкіл, що стали ІТ-фахівцями, значно нижчий, ніж потреба у таких фахівцях, і різниця у потребі та «факті» має тенденцію до зростання. Беззаперечним висновком буде те, що на ці знання існує соціальне замовлення;
- усі, хто не буде програмістами, отримують під час вивчення ОАП навички структурованого, «впорядкованого» мислення, які є вельми корисні у будь-якій сфері діяльності;
- більшість із тих, хто не зумів, не дивлячись на бажання та наявне соціальне замовлення, оволодіти програмуванням, постраждали через те, що їх або зовсім у школі тому не вчили (характерно для програм останніх років 10–15), або вчили недосконало, а інколи — просто неправильно.

Що ж мається на увазі?

Два різних поняття — вивчення програмування і вивчення мови програмування у нас якось дивно перетворилися в одне. Кожне з них має свої складнощі. Вивчення основних алгоритмічних конструкцій і навіть простеньких алгоритмів справа для учнів вельми непроста, а одночасне вивчення синтаксису тієї чи іншої мови створює свої труднощі. Саме вони виходять для учнів на перший план, саме з ними учні намагаються справитися у першу чергу. Для них пошук відповіді на запитання: «Як це записати на Delphi?» стає актуальнішим ніж роздуми: «А як це слід зробити алгоритмічно правильно?». Часу й розуміння необхідності використання псевдокоду для розробки і запису алгоритмів учителю не вистачає. На практиці — «щось-там, що не зрозуміло» кодували тією чи іншою мовою «за зразком». Від такого вивчення ОАП у середній школі дійсно користі ніякої.

Бажання «осучаснити» шкільне програмування, перейти без знання структурного програмування одразу на об'єктно-орієнтоване породило підміну поняття «візуальне середовище розробки об'єктно-орієнтованих програм» поняттям «візуальне програмування». Того часу, що відводиться на вивчення чинними програмами отого «візуального» програмування, вистачає або на часткове вивчення вельми непростого середовища, або на сліпе копіювання 1–2 прикладів «працюючих» програмок з підручника абсолютно без розуміння «що і до чого». І від такого вивчення «ОАП у стилі ООП» теж абсолютно ніякої користі.

Вихід із положення можна знайти, якщо чітко усвідомити завдання шкільного курсу програмування: розвиток алгоритмічного мислення учня, уміння ним створювати правильні алгоритми для розв'язування певного кола задач (у першу чергу — із тих сфер, що є предметом вивчення в школі), доведення надійності створених алгоритмів і реалізацію їх у вигляді програм однією з мов програмування.

Як бачимо, у цьому переліку мова програмування не є визначальним чинником. Але вибір мови, на яку має спиратися розвивальний шкільний курс ОАП, має велике значення. Обрана мова має відповідати, на думку автора, таким вимогам:

- транслятор мови повинен бути кросплатформним і безкоштовним;
- мова повинна мати простий синтаксис, з одного боку, і потужні засоби, з іншого;
- програми, записані обраною мовою, повинні бути короткими і «прозорими», форма запису повинна бути максимально звичною і зрозумілою для людини;

- мова повинна мати розвинуті бібліотеки, можливості розробки промислових додатків (роботи з мережею, GUI, базами даних, CGI та ін.);
- повинна рівною мірою підтримувати стилі як структурного, так і функціонального та об'єктно-орієнтованого програмування, тобто поняття «функція», «метод», «об'єкт», «клас», «простір імен» повинні бути наскрізними і логічно впливати одне з іншого;
- мова повинна бути сучасною, не «мертвою», входити хоча б у 10 найбільш уживаних в реальному виробництві мов програмування;
- середовище розробки програм не повинно бути перевантажене додатковими функціями і можливостями, складність освоєння яких «затмарює» і відсуває на другий план алгоритмічну складність проекту. Обов'язковою також є наявність консольного транслятора.

На нашу думку, усім перерахованим вимогам відповідає мова програмування Python [1], яка в усьому світі широко використовується як для навчання, так і у промисловому програмуванні. Python займає четверте місце (після Java, C та C++) в рейтингу популярності мов програмування ТЮБЕ Programming Community Index [2]. Мова активно розвивається, має відкритий (Open Source) програмний код, поширюється за GPL — ліцензією, тобто може безкоштовно використовуватися навіть для розробки комерційних продуктів [1].

Python має простий і лаконічний (C-подібний, але значно прозоріший) синтаксис. Прості програми записуються у кілька рядків, відсутні інструкції, що не мають безпосереднього відношення до алгоритму (для прикладу — `int main()` у мові C). Як правило, програми мовою Python є коротшими, ніж на C, C++, Pascal, та, тим більше, C#. Мова сучасна, підтримує високорівневі складені структури даних (списки, множини, асоціативні масиви, інше). Python із самого початку створювався на основі парадигми об'єктно-орієнтованого програмування, але чудово пристосованим для структурного і функціонального програмування. Без прийомів структурного (або, як часто говорять, процедурного) програмування не обійтися під час вивчення основних алгоритмічних конструкцій і початкових методів розробки алгоритмів за технологіями «згори-вниз», а у більшості «об'єктно-орієнтованих від народження» мов (C#, Java) програми, написані у стилі структурного програмування, є так само недолугими, як об'єктно-орієнтовані Pascal, об'єкти у якому є фактично надбудовою.

Python має багату бібліотеку, що постійно поповнюється, а це дозволяє розробляти графічні додатки, web-додатки та ін.

Крім консольного інтерпретатора мови, з якого, на наше глибоке переконання, слід взагалі розпочинати знайомство з програмуванням, Python має кілька різних середовищ розробки програм. IDLE — стандартне середовище, що входить у дистрибутив, просте, лаконічне, зрозуміле. Гарною альтернативою може бути Wing IDE 101 — безкоштовна версія кросплатформеного середовища Wing IDE [3].

Мова чудово підходить для проведення олімпіад школярів і студентів з програмування. На дуже багатьох змаганнях, зокрема на районних і обласних олімпіадах у Донецькій і Вінницькій областях і Всеукраїнських Інтернет-олімпіадах NetOI [4] Python включено до переліку офіційних мов олімпіади. До речі, автоматизована

система перевірки NetOI написана мовою Python. Варто було б включити Python до переліку офіційних мов 4 етапу Всеукраїнських олімпіад з інформатики.

На сьогоднішній день існує дві гілки розвитку мови — Python 2 і Python 3, не зовсім між собою сумісних. Для вивчення в школі варто обрати Python 3 [5] як більш сучасну реалізацію, хоча відмінності між гілками, за великим рахунком, або «косметичні», або лежать далеко за межею досяжності «шкільного» програмування.

До недоліків Python традиційно відносять порівняно невисоку швидкість виконання програм, написаних цією мовою. Дійсно, транслятор Python — «класичний» інтерпретатор, тому програми працюють повільніше. Але із зростанням обчислювальної потужності ПК це стає несуттєвим, а на етапі початкового вивчення ОАП, про що йдеться у статті, інтерпретуючий транслятор стає очевидною перевагою.

У фізико-математичній гімназії №17 м. Вінниці, починаючи з 1999 р., Python використовується як «перша мова» під час вивчення ОАП у 8–11 класах. У кількох школах м. Донецька, як у спеціалізованих, так і звичайних, Python також використовується як основний інструмент вивчення ОАП. При Донецькому обласному інституті післядипломної освіти педагогічних працівників проводяться літні школи з вивчення Python як для учнів, так і для вчителів. За ці роки нами напрацьовано чималий досвід, створено низку методичних посібників і дистанційних курсів, що розміщені, зокрема, у вільному доступі в Інтернеті на ресурсному центрі освітнього порталу гімназії [6]. Є немало дуже цікавих матеріалів з методики використання мови Python для навчання у російськомовному сегменті Інтернету.

Висновки. Пропонується використовувати Python як мову для навчання учнів основам алгоритмізації та програмування, починаючи з першого знайомства у початкових класах на основі модуля Python Turtle, для вивчення основних алгоритмічних конструкцій і базових алгоритмів (консольний інтерпретатор), і аж до свідомого освоєння парадигми ООП під час створення достатньо складних програмних проектів у старших класах.

Якщо ж хтось з учнів зуміє піти далі в програмуванні, то його знання після завершення навчання будуть реально витребувані на ринку праці, адже загальновідомо, що 30% програмного коду Facebook і 80% коду Google реалізовано саме на Python.

Ця мова якнайкраще може бути використана в рамках виконання вже згаданої програми поглибленого вивчення інформатики в 8–9 класах і програм профільного навчання у старших класах.

Література

1. <http://www.python.org> — офіційний сайт Python.
2. http://www.tiobe.com/tiobe_index — ТЮБЕ Programming Community Index.
3. <http://www.wingware.com> — середовище розробки Wing IDE.
4. <http://netoi.org.ua> (www.olymp.vinnica.ua) — Всеукраїнські Інтернет-олімпіади з інформатики NetOI.
5. Саммерфілд М. Программирование на Python 3. Подробное руководство. — М.: Символ-Плюс, 2009.
6. <http://disted.edu.vn.ua> — система дистанційної підтримки освіти школярів.
7. Пасіхов Ю.Я. Шкільна інформатика: програми, підручники, учителі // Комп'ютер у школі та сім'ї. — 2013. — №5. — С. 3–4.
8. Програма курсу «Інформатика» (8–9 класи загальноосвітніх навчальних закладів з поглибленим вивченням інформатики) // Комп'ютер у школі та сім'ї. — 2013. — №6. — С. 40–53.

(Далі буде)