

## ЯКУ МОВУ ПРОГРАМУВАННЯ ВИВЧАТИ У ШКОЛІ

**Від редакції.** Колегія МОН України розглянула і надала гриф «Затверджено Міністерством освіти і науки» навчальній програмі з інформатики для поглибленого вивчення інформатики у 8–9 класах (протокол №2/4-2 від 27.06.2013 року). Нижче в таблиці 1 наведено розподіл годин на розділи навчальної програми.

Таблиця 1

№	Назва розділу	Класи і кількість годин		Всього
		8 кл.	9 кл.	
1	Інформаційні технології у суспільстві	0	3	3
2	Математичні основи обчислювальної техніки	6	0	6
3	Кодування даних	10	0	10
4	Комп'ютер як універсальний пристрій для опрацювання даних	12	0	12
5	Інформаційні технології			
5.1	Створення та опрацювання текстових документів (електронний документообіг, комп'ютерні публікації)	8	12	20
5.2	Створення та опрацювання графічних зображень	6	0	6
5.3	Створення та опрацювання об'єктів мультимедіа	8	0	8
5.4	Створення та опрацювання мультимедійних презентацій	8	0	8
5.5	Технології опрацювання числових даних у середовищі табличного процесора	14	0	14
5.6	Система управління базами даних	0	16	16
7	Мережні технології	0	18	18
8	Основи інформаційної безпеки	0	4	4
9	Створення персонального навчального середовища	0	6	6
10	Моделювання	0	8	8
11	Основи алгоритмізації та програмування	56	56	112
12	Розв'язування компетентнісних задач, виконання навчальних проєктів	8	12	20
13	Резерв часу	4	5	9
	Всього	140	140	280

Звернемо увагу на те, що на алгоритмізацію, моделювання, кодування даних, математичні основи обчислювальної техніки відведена значна кількість годин. У зв'язку з цим виникає питання: на основі якої мови програмування вивчати алгоритмізацію у 8–9-их класах?

Редакція звернулася до відомих учителів інформатики, фахівців у галузі навчальної інформатики з проханням взяти участь в обговоренні цієї важливої і складної проблеми.

Нижче публікуються думки і пропозиції авторів (надруковані в порядку надходження до редакції).

### Горошко Юрій Васильович,

*канд. педаг. наук, доцент Чернігівського національного університету.*

Якщо коротко, то я прихильник традиційного підходу — блок-схеми і мова Паскаль.

#### Причини:

- усі вчителі інформатики вивчали цю мову;
- це одна з двох офіційних мов олімпіади з інформатики;

- кросплатформеність;
- мова проста у вивченні;
- величезна кількість методичних і дидактичних матеріалів;
- можливість створювати реальні програми.

### Мельник Валентин Іванович,

*учитель інформатики Полтавського обласного ліцею-інтернату для обдарованих дітей із сільській місцевості при Кременчуцькому педагогічному училищі імені А. С. Макаренка, заслужений вчитель України.*

Як на мене, то краще залишити вибір мови програмування за вчителем.

Мені більш подобається Pascal. Як на мене, він легше сприймається учнями молодших класів. Кожна з мов має як переваги, так і недоліки.

**Ягієв Шаміль,**

*R&D менеджер компанії Playtika Ukraine, неоднаразовий учасник Всеукраїнських олімпіад з інформатики.*

Разумеется, я приветствую подобную программу. Мои мысли по поводу языка программирования следующие.

**1. Сравнение языков**

Я бы рассматривал только такие языки: Free Pascal, GNU C++, Java, C#. С моей точки зрения Basic слишком тупиковая ветвь в плане синтаксиса, чтобы его изучать. Скриптовые языки Perl, Python, PHP, Groovy, Ruby и т.д., ориентированы на быстрое решение каких-то прикладных задач, в основном Web, и не являются базовыми для алгоритмизации. Есть функциональные языки, которые становятся все более популярными в практическом программировании, однако в школе их изучение вероятно только факультативно.

**Pascal** — классический алгоритмический язык, который позволяет сосредоточиться на алгоритмах и реализации своих структур данных. Многие учителя знают этот язык. С другой стороны, не имеет серьезной прикладной реализации (Delphi — не в счет). Нет современной визуальной среды.

**C и C++** — прекрасный язык для реализации эффективных алгоритмов. Имеет прикладное применение для реализации критических с точки зрения производительности модулей, разработки для iPhone/iPad. Прекрасно подходит для изучения ООП. Имеет мощные библиотеки алгоритмов, и более эффективный компилятор, нежели FreePascal, что делает язык наилучшим для применения в олимпиадах. С другой стороны, язык более сложен для изучения — сложная работа с указателями, сложный синтаксис. Язык «позволяет совершать ошибки». Вряд ли у нас много учителей, которые знают C++ и могут ему обучать.

**Java** — очень востребованный на рынке труда язык (как и технологии, с ним связанные), разработки для Android. Может применяться на олимпиадах, но, в основном, со студенческого уровня. Имеет современные среды разработки. С точки зрения обучения более сложен чем Pascal, т.к. требует понимания ООП.

**C#** — востребованный на рынке труда язык (в рамках NET технологии), разработки для мобильных приложений. Допустим в некоторых соревнованиях по программированию со студенческого уровня. С точки зрения обучения более сложен чем Pascal, т.к. требует понимания ООП. Имеет современные среды разработки.

**2. Анализ Цели**

Один из важных вопросов. Какую Цель наша образовательная система ставит перед собой?

(а) ИТ — одно из важнейших направлений экономического развития страны и мы массово готовим профессионалов?

(б) Мы развиваем интеллектуальный потенциал нации и делаем акцент на естественные науки в школе?

(в) Мы хотим более эффективно выбирать лучших людей, способных к информатике, жертвуя при этом средними показателями?

В случае (а) — моя рекомендация: ориентироваться на Java, C#, Web-технологии. Что-то на постоянной основе, что-то на факультативной. C++ — факультатив для олимпиадников.

(б) Наверное, Pascal + C++ — как факультатив для наиболее способных.

(в) C++ как база + углубленная алгоритмика для олимпиадников.

**3. Численный анализ**

У меня есть предположения, что будет легче даваться ребятам, а что будет тяжелее. Но мне сложно дать численную оценку этой разницы. К примеру, оценить сколько людей потеряют интерес к информатике, если учить будут C++, а не Pascal. Поэтому я бы провел так называемое a/b тестирование. На определенной выборке учителя, с примерно одинаковым знанием языков, параллельные группы обучает двум разным языкам и сравнивает результаты по некоему тестированию. На основании этого делается статистический анализ.

**4. Анализ опыта**

Кстати, как с этим вопросом обстоят дела в России, США, Китае, Индии? Кто-то изучал их опыт?

Нужно как-то синхронизироваться с институтами. Чему учат там?

**5. Главный вопрос**

Однако, во всей этой истории с выбором языка есть вопрос, который я считаю самым важным. Мы можем выбрать язык, но кто ему будет обучать детей? Достаточно ли у нас людей, в принципе способных обучать алгоритмике? Как мы обучаем их? Какие программы повышения квалификации для учителей 8–9-ых классов? И еще, если они действительно знают эти языки программирования и могут спокойно перейти на любой другой, который мы скажем, почему они не зарабатывают те же деньги, что и в школе, но в долларах?

С моей точки зрения, нам нужно сосредоточиться на образовательной системе на основании удаленных лекций и курсов по примеру: <https://www.coursera.org/>.

С другой стороны, наверное, на данный момент нужно давать возможность имеющимся учителям давать возможность обучать на тех языках, к которым они привыкли, и делать акцент на углублении знаний и методики, а не изменении языка.

**Бодрик Оксана Олександрівна,**

заступник директора з навчально-виховної роботи, учитель інформатики спеціалізованої школи I–III ступенів №12 з поглибленим вивченням інформаційних технологій Білоцерківської міської ради Київської області.

Уже не перший рік перед учителями інформатики постає проблема вибору мови програмування для вивчення теми «Основи алгоритмізації та програмування». Зокрема, це питання гостро постає у процесі поглибленого вивчення інформатики. Потрібно акцентувати увагу, що саме під час вивчення основ алгоритмізації і програмування, а не для підготовки до олімпіад з програмування. Досвідчені вчителі звертають увагу на те, що мова програмування – це всього лише мова. Мови програмування відрізняються лише синтаксисом. Головне не синтаксис, а алгоритмічне мислення. Якщо його не має, то немає навичок програмування. Якщо наявне алгоритмічне мислення, то й не буде проблемою перехід з однієї мови програмування на іншу.

Під час вивчення основ алгоритмізації в середній школі основна увага насамперед приділяється [1,3]:

- основним поняттям алгоритмізації; властивостям алгоритмів; способам представлення алгоритмів; базовим алгоритмічним структурам; типам алгоритмів;
- поняттю моделі і моделюванню; класифікації моделей; інформаційній моделі; етапам розв'язування задач на комп'ютері;
- аналізу поставленої задачі, методам формалізації моделювання реальних процесів і явищ;
- вибору виконавця поставленої задачі, виходячи з тих міркувань, що він є також певним об'єктом із притаманними йому властивостями і набором дій, які потребують аналізу для правильного й ефективного їх використання;
- методам і засобам формалізованого опису дій виконавця, сучасним засобам їх конструювання і реалізації за допомогою комп'ютера.

Алгоритмізація, як розділ інформатики, який вивчає процеси створення алгоритмів, традиційно відноситься до теоретичної інформатики внаслідок свого фундаментального характеру. Тому актуальним є вивчення саме структурованих мов програмування, з наступним вивченням теоретичних основ, які реалізують загальні поняття і принципи об'єктно-орієнтованого програмування, а також формування навичок програмування на об'єктно-орієнтованих мовах.

На практиці використовуються не мови, а системи програмування, тобто системи для розробки нових програм на конкретній мові програмування. Для популярних мов програмування (C, C++, Паскаль, Бейсик, Фортран та ін.) існує багато таких систем програмування. Почали використовуватись системи програмування на мові Java, які дозволяють створювати програми, що викликаються під час перегляду Web-сторінок у глобальній електронній мережі Internet.

З іншого боку, учасники олімпіади з інформатики II етапу у 2012–2013 навчальному році [4] могли вибрати мову програмування з такого переліку: Pascal,

C або C++. Середовища розробки програм FreePascal 2.4 (чи новішої версії), CodeBlocks 10.05 (чи новішої версії). Для перевірки робіт учасників на IV етапі використовувались такі версії компіляторів: FPC 2.4, GCC 4.6.

Враховуючи все вищесказане, розглянемо такі напрями вибору мови програмування для вивчення основ програмування:

- тип мови;
- простота і доступність мови;
- систему програмування;
- вільне розповсюдження;
- використання на олімпіадах з інформатики (а точніше, з програмування) можливість наступності у вивченні об'єктно-орієнтованого програмування;
- наявність методичного забезпечення та педагогічних кадрів, які володіють вибраною мовою або мають бажання нею оволодіти.

Розглянемо таку класифікацію сучасних мов програмування (рис. 1).

Існує точка зору, що для початкового навчання об'єктно-орієнтовані мови програмування не рекомендуються.

На сучасному етапі найпопулярнішими процедурними структурними мовами програмування для вивчення основ програмування є:

- Basic;
- Pascal;
- C (C+)

Розглянемо особливості кожної з мов.

Бейсик (BASIC, скорочення від англ. Beginner's All-purpose Symbolic Instruction Code — універсальний код символічних інструкцій для початківців; *англ.* basic — основний, базовий) — сімейство високорівневих мов програмування. Був розроблений у 1963 році професорами Дартмутського коледжу Томасом Куртом (Thomas E. Kurtz) і Джоном Кемені (John G. Kemeny).

Характеризується простою засвоєнням і наявністю універсальних засобів для розв'язування науко-

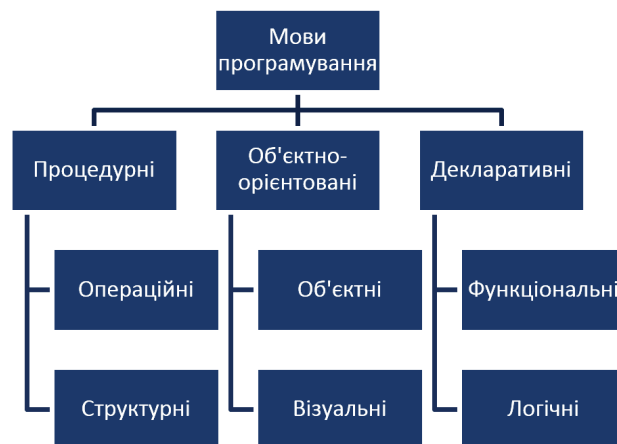


Рис. 1

вих, технічних і економічних задач, а також інших, наприклад, ігрових.

Досить популярною є мова VB.Net (VisualBasic .Net) – об'єктно-орієнтована мова, що входить до комплексу засобів розробки від Майкрософт. Основним аргументом її вивчення є те, що ІТ-компанії вважають мінімально необхідною базою для будь-якого розробника.

У списку посібників для вивчення VB, рекомендованих МОН України [2], є два:

- Інформатика: VisualBasic (автор Бондаренко О.О.).
- Основи візуального програмування (автори Завадський І.О., Заболотній Р.І. за заг. ред. Гуржія А.М).

У переліку мов для використання на олімпіадах цієї мови не має.

Паскаль був створений Нікклаусом Віртом у 1968/9 роках (опублікований у 1970-му). Особливостями мови є строга типізація і наявність засобів структурного (процедурного) програмування. На думку Вірта, мова повинна сприяти програмуванню, тому у Паскалі майже немає синтаксичних неоднозначностей, а сам синтаксис автор намагався зробити інтуїтивно зрозумілим навіть за першого знайомства з мовою. Простота мови дозволяє швидко її освоїти і створювати алгоритмічно складні програми. З іншого боку, розвинуті засоби представлення структур даних забезпечують зручність роботи як з числовою, так і символічною і бітовою інформацією. Завдяки реалізації у цій мові ідей структурного програмування Паскаль дає дуже багато у розумінні суті програмування, вона прививає хороший стиль програмування, великі можливості для розробки і налагодження. Попри це, вона є прототипом таких мов програмування нового покоління, як Ada і Modula-II.

На сучасному етапі є досить багато літератури з програмування саме на Паскалі відомі посібники Караванової Т.В.:

- «Основи алгоритмізації та програмування. 777 задач з рекомендаціями та прикладами»;
- «Інформатика. Методи побудови алгоритмів та їх аналіз. Необчислювальні алгоритми»;
- «Інформатика. Методи побудови алгоритмів та їх аналіз. Обчислювальні алгоритми».

Якщо розглядати можливість наступності, то система візуального проектування Delphi в консольному варіанті це Паскаль, донедавна використовувалась на олімпіадах. Зрозуміло, що в об'єктно-орієнтованому варіанті не використовується на олімпіадах.

Хочеться звернути увагу на FreePascal (повна назва Free Pascal Compiler, FPC) — це компілятор мови програмування Паскаль з відкритими вихідними кодами.

Він сумісний з BorlandPascal 7, ObjectPascal, Delphi, має низку додаткових можливостей, наприклад, підтримує перевантаження операторів. FPC — кросплатформний інструмент, що підтримує величезну кількість платформ. Серед них — AmigaOS, DOS, Linux, \*BSD, OS/2, MacOS X і Win32.

Компілятор поширюється на умовах GNU General Public License, а значна частина бібліотек, у тому числі ядро RTL — на умовах м'якшої GNU Lesser General Public License.

Співробітник фірми Bell\_Labs Деніс Рітчі створив мову C у 1972 році під час співпраці з Кеном Томпсоном як інструментальний засіб для реалізації операційної системи Unix, але популярність цієї мови швидко переросла рамки конкретної операційної системи і конкретних задач системного програмування.

Мова C (початок 70-х років) також відноситься до мов процедурного програмування. Однією з особливостей мови C є те, що розбіжність між виразами й операторами згладжуються. Крім того, у цій мові *відсутнє поняття процедури*, а використання підпрограм ґрунтується на понятті *функції*, яка має можливості процедури. З одного боку, з набору управляючих конструкцій і структур даних C можна віднести до мов високого рівня, а з іншого, — вона має набір засобів прямого звертання до функціональних вузлів комп'ютера, тобто її можна використовувати як операційну мову.

C створювався з метою: зробити більш простим написання великих програм з мінімумом помилок за правилами процедурного програмування.

Вивчення C — це основа, яка дозволить легко перейти на Java або C#, оскільки велика частина синтаксису дуже схожа. Попри це, синтаксис, схожий C, використовується в основних мовах web-програмування.

Мова C++ є засобом об'єктного програмування, яке, можливо, замінить традиційне процедурне програмування. Абстракція, реалізація, наслідування та поліморфізм — необхідні властивості мови C++, завдяки чому він не тільки є універсальним, як і C, але й об'єктною мовою.

Але C, C++ — мови професійного програмування. Для початкового вивчення не рекомендується. Вона потрібна для рівня міжнародних олімпіад, або може мати перевагу для розв'язування деяких класів задач на всеукраїнському рівні.

З посібниками для вивчення цієї мови, рекомендованими МОН, я не знайома.

Її можна вивчати після детального знайомства з Паскалем як курс за вибором або факультатив. У переліку програм, рекомендованих МОН України є програма курсу за вибором «Основи програмування на C#» (автори М.О. Войцеховський, Т.Г. Проценко) та «Основи програмування» (автори С.Д. Вапнічний, В.В. Зубик, В.А. Ребрина). Автори останньої програми пропонують два варіанти: для Паскаля та C++, а також посібники відповідно для кожної з мов ([http://info.hoip-ro.km.ua/do\\_novogo\\_navchalnogo\\_roku.html](http://info.hoip-ro.km.ua/do_novogo_navchalnogo_roku.html)), орієнтованих на підготовку до олімпіад.

Розглянемо аспект підготовки вчителів інформатики. Не є секретом той факт, що спеціальність «Інформатика» вчителі отримують додатково до якогось предмета, наприклад, математика, фізика. Та й під час вивчення програмування у ВНЗ майбутніми педагогами вивчається одна мова програмування, як правило, Паскаль, та й звертається увага, в основному, на оволодіння основами алгоритмізації і програмування (у прямому значенні слова «основи»). З іншого боку, «сильні» молоді програмісти не мають бажання працювати в школі за таку зарплату або на досить висо-

кому рівні займатись самоосвітою для оволодіння новою мовою програмування.

Отож, можна зробити деякі висновки. На даному етапі для вивчення основ алгоритмізації та програмування все таки актуальним залишається використання мови Паскаль, а для підготовки до олімпіад за рахунок варіативної частини – на вибір учителя.

#### Використані джерела

1. Навчальна програма поглибленого вивчення інформатики для учнів 8-9 класів загальноосвітніх навчальних закладів [Електронний ресурс]. — Режим доступу: [http://www.mon.gov.ua/ua/activity/education/56/general-secondary-education/educational\\_programs/1349869497/](http://www.mon.gov.ua/ua/activity/education/56/general-secondary-education/educational_programs/1349869497/).

2. Перелік навчальних програм, підручників та навчально-методичних посібників, рекомендованих Міністерством освіти і науки, молоді та спорту України для використання в основній і старшій школі у загальноосвітніх навчальних закладах з навчанням українською мовою у 2012/13 навчальному році закладів [Електронний ресурс]. — Режим доступу: <http://mon.gov.ua>.

3. Копяев О.В. Влияние современных информационных технологий на изучение основ алгоритмизации в средней школе // Компьютер у школі та сім'ї. — 2000. — №2. — С. 24–27.

4. Лист Інституту інноваційних технологій і змісту освіти від 19.11.2012 р. №14.1/10-3187 Про проведення III етапу Всеукраїнської учнівської олімпіади з інформатики 2012/2013 н.р. закладів [Електронний ресурс]. — Режим доступу: <http://old.lito.kr.ua/>.

#### Мица Олександр Володимирович,

*доцент Ужгородського національного університету.*

Я думаю, що основну увагу треба приділити алгоритмічній складовій та підбору цікавих різнопланових задач, які б могли зацікавити широкий прошарок учнів. Вибір мови програмування (на сьогоднішній

день — це Паскаль або С++) треба залишити за вчителем. А також необхідно забезпечити вчителів якісними посібниками і допоміжним матеріалом з використанням обох мов програмування.

#### Зеленяк Олег Петрович,

*канд. педагогічних наук, вчитель математики та інформатики Олександрійського колегіуму, Соросівський вчитель, Заслужений вчитель України.*

На мой взгляд, суть проблемы в том, что преподаватели программирования и реальные программисты-разработчики были и, наверное, останутся на разных позициях. Действительно, TurboPascal устарел, не служит сегодня средой разработки приложений, DOS не вернется, повсеместно мобильные устройства, all about cloud computing и т.д.

Кроме того, под Windows 7 или 8 нужно использовать FreePascal, Pascal ABC, эмулятор DOSBox и т.п. Это создает неудобства. Но как начинать обучение с «нуля» на С или визуального создания «форточек» я, имея огромный опыт преподавания, не представляю.

Начинать нужно с данных и алгоритмов, а потом уже их кодирования. При этом последнее в школе должно быть по возможности ясным и понятным для детей соответствующей возрастной группы.

Завадский И., агитируя за объекты, пишет: «Программа состоит из многих алгоритмов, исполняющихся вследствие наступления того или иного события ...».

Здесь ни первая, ни вторая часть не правильны!

Учебная программа (мы говорим об обучении!) может кодировать всего один алгоритм и, естественно, может исполниться без внешних интерактивных реакций (как, например, тесты программой олимпиадника).

И важно понять, что алгоритмы изучают по одному, а уже потом объединяют в необходимом порядке. Возможно прав Глинский Я. (див. журнал «Компьютер у школі та сім'ї» №8, 2011, с. 9), что Паскаль и не нужно было начинать изучать в учебных заведениях.

Действительно, я очень хорошо знаю среду QuickBasic 4.5, практически не уступающую (кроме скорости компиляции) Паскалю. В ней сочетается интерпретатор и компилятор, простейший переход в графический режим (SCREEN 12), имеется операция возведения в степень и т.п. И главное: с этой среды переход на VisualBasic естественен.

Но сегодня именно по Паскалю больше всего хорошей различной учебной литературы.

Кстати, сейчас я веду переговоры с Московским издательством ДМК о пятом издании моей книги по Паскалю, которая за 10 лет из раздела дополнительной литературы переместилась в раздел основной литературы для студентов (вообще, я писал ее для школьников!). И Паскаль, по-прежнему, изучается сегодня в большинстве вузов Украины и России.

Сегодня по этой проблеме имеются полярные точки зрения. Этот спор, наверное, бесконечен. Вообще, с опытом у меня утвердилось мнение, что важнее как учить, а не чему учить.

И все же, примерная схема школьного обучения может быть такой:

1. Алгоритмы.

2. Начальная среда (Logo или аналог Кенгуренок+Пылесосик под Windows или Scratch).

3. Среда программирования (TurboPascal, QuickBasic).

4. Объектно-ориентированная среда программирования (ознакомительно).

**(Далі буде)**

