

Учитель інформатики
Щасливського НВК
Бориспільського району,
ОСТАПЕЦЬ В.С.

(матеріал за 2006 рік, актуальний і сьогодні)

ЯКОЮ БУТИ ШКІЛЬНІЙ ІНФОРМАТИЦІ ЗАВТРА?

Двадцять років тому серед шкільних навчальних дисциплін з'явилась інформатика. Спробуємо провести аналіз сьогоденного стану предмету та розібратись у напрямках його подальшого розвитку. Але спочатку пригадаємо, що необхідність введення курсу інформатики в шкільні навчальні плани була продиктована часом і полягала в набутті комп'ютерної грамотності учнями, студентами, викладачами і, в перспективі, всіма активними верствами населення. Це завдання мало глобальний характер, його актуальність не зменшилась зі зміною суспільного ладу та виникненням незалежної України, адже в кінці ХХ століття стали говорити про нову інформаційну еру і формування відповідної їй інформаційної культури.

У певному розумінні сьогодні можна вважати, що так звана комп'ютерна неграмотність ліквідована. Дійсно, учні масово користуються комп'ютерами, як у стінах школи, так і поза її межами, при вступі у вузи мають достатній для здобуття вищої освіти рівень володіння комп'ютерними технологіями, останні вже досить широко застосовуються у навчально-виховному процесі, зроблено великі кроки для переходу до електронного діловодства, комп'ютерна техніка широко застосовується у науці та господарській діяльності. То, можливо, шкільна інформатика виконала поставлені завдання і потреби в ній більше немає? Нам думається, навпаки - сьогодні поряд із традиційними, виникли нові завдання. Перш за все, про ліквідацію комп'ютерної неграмотності можна говорити не остаточно, а лише в певному розумінні, адже до цього часу значна частина шкільних викладачів не можуть практично і не вважають за потрібне використовувати комп'ютерні технології у своїй роботі, а нові покоління учнів, широко знайомлячись із комп'ютером у побутових умовах, отримують спрощене і спотворене уявлення про нього.

Для глибшого аналізу піднятого питання слід обговорити ще його один, на наш погляд, надзвичайно важливий аспект. Він впливає з принципу, що ліг в основу шкільної інформатики, за висловом її “батька“ академіка А.П. Єршова “шкільна інформатика повинна йти від математики“. Безперечно, знайдуться ті, хто заперечуватиме. Мовляв, у часи Єршова комп’ютерна техніка була на такому примітивному рівні розвитку, що йшлося взагалі не про комп’ютер, а про електронно-обчислювальні машини, а вся інформатика зводилась до алгоритмізації та програмування, тому і математична компонента була такою важливою. Спробуємо поміркувати про роль розділу “Алгоритмізація і програмування“. Щойно ми згадали, що колись це був основний розділ всього курсу інформатики, але поступово, особливо протягом останнього десятиріччя, його роль звелась майже до нуля, адже згідно універсального профілю кількість навчальних годин на вивчення алгоритмізації і програмування зменшена в 4 рази у порівнянні з попередньою програмою і становить всього 12 навчальних годин.

Безперечно, шкільну інформатику не можна зводити до безсистемної маси знань про комп’ютер та його застосування, адже, як і в інших шкільних предметах, тут мова іде про знайомство з наукою. Але відомо, що будь-яка галузь знань лише тоді може у повному розумінні стати наукою, коли вона включає математичні методи дослідження. Хочеться спитати, де математичний апарат і математичні методи у “чистому курсі користувача“? Звідси перший висновок: розділ “Алгоритмізація і програмування“ ніколи не втратить актуальності в курсі інформатики.

Але поглянемо на питання ще з одного боку. За відомим висловом М.В. Ломоносова, математику ще й тому слід вивчати, що вона розум до ладу приводить. Це – постулат, який ніким не ставився під сумнів. Математику, хоча б в обсязі загальноосвітньої школи, вивчають всі люди, бо вона є невід’ємною складовою середньої освіти. На наш погляд, інформатика, як прикладна галузь знань, дає *своєрідне уточнення та узагальнення не тільки таких понять, як інформація та алгоритм, а й багатьох загальноосвітніх понять, від лінгвістичних, наприклад, поняття алфавіту та мови, до природничих і математичних, як, наприклад, поняття числа, величини, типу, або виду, виразу, формули, фун-*

кції, процедури тощо. Іншими словами, *інформатика*, покликана робити наступний за математикою крок у “*приведенні розуму до ладу*”.

Про що тут ідеться? Візьмемо поняття “алфавіт”. У початкових класах його формують лише як множину знаків, що позначають звуки конкретної національної мови. Чи правильно, коли дорослі люди так розуміють поняття “алфавіт” до кінця життя? А як бути із висловом Р. Декарта “Дайте точні визначення поняттям і ви звільните людство від половини його помилок”. Навіть у ХХІ столітті ми спостерігаємо, до яких тяжких для людства наслідків приводить довільне трактування історичних, політичних, економічних і взагалі – наукових понять. Тому без перебільшення можна стверджувати, що все починається з дуже приблизного уявлення про алфавіт! Учні дивуються і ... трохи не вірять, коли на уроці інформатики при вивченні мови програмування їм повідомляють, що алфавіт – це набір *усіх* символів, необхідних для висловлення думок. А потім згоджуються, що, до українського алфавіту на рівних правах, наприклад, з буквою “а”, входять арабські цифри, розділові знаки і т.д. Якщо формування такого уявлення про алфавіт не передбачено в початкових класах та на уроках рідної мови, то вивчення мови програмування вже оправдано хоча б цим. Але, на наше глибоке переконання, будь-який учитель-класовод чи мовник повинен у свій час узагальнити учням поняття алфавіту, бо неможливе розуміння процесу пізнання без принципу “від споглядання до узагальнення”, адже аналіз та синтез – це α і ω мислительського процесу.

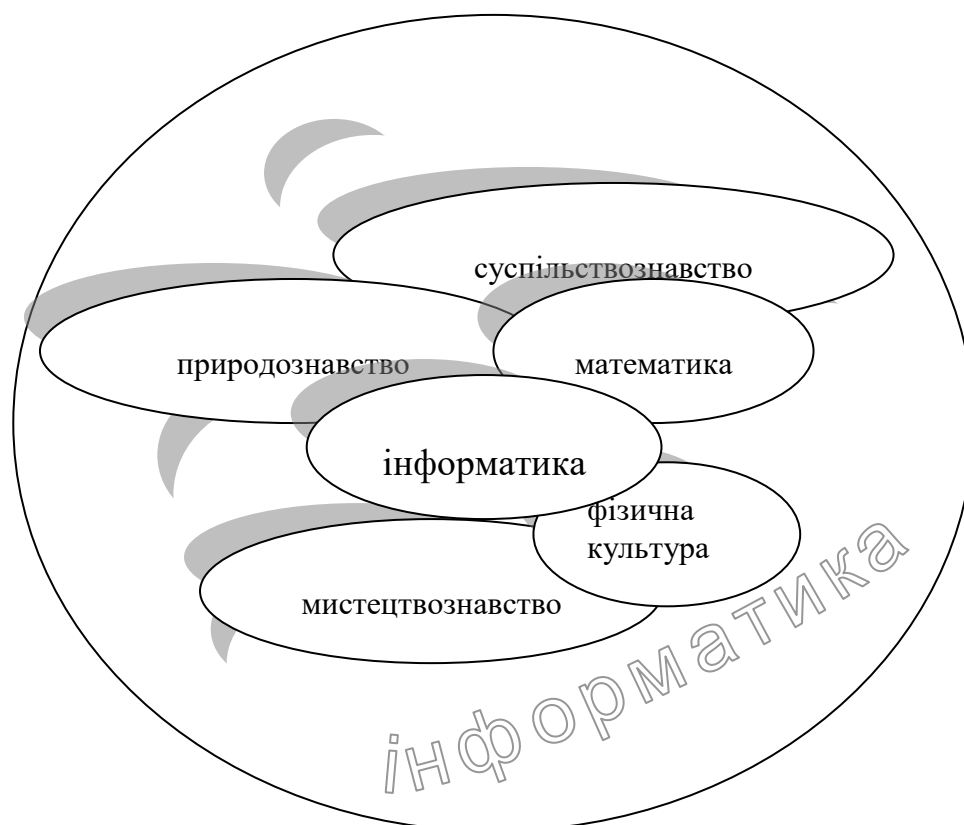
Давайте порівняємо поняття величини чи функції у математиці та фізиці – з одного боку, та в інформатиці – з іншого. Величина у першому випадку має характеристики ім’я та значення, про тип майже не згадується, адже з точки зору математики і фізики величини фактично бувають лише числові, правда є розділення на скалярні і векторні. З програмування учні дізнаються про кодування величин, про потребу їх оголошення та опису, про різноманіття типів, особливо у мові Паскаль, у якому тільки цілочисельних типів кілька. На перший погляд це дивує, але коли пояснити про представлення величини у пам’яті комп’ютера, здивування проходить. Ім’я величини в математиці не обговорюється. Час – t , швидкість – v і т.д. Мови про позначення величини, як деякого

набору допустимих символів, нема. Про це говориться в інформатиці. Як виявляється ім'я, чи ідентифікатор може бути не тільки у величини, а й у файла, папки, людини врешті. Коли пояснюєш, що ім'я, в тому числі і ваше – це лише позначення, деякий набір символів, зникає абсурдна і шкідлива для наукового світогляду містика імен. Учні перестають дивуватись і погоджуються, коли їм говорять, що ім'я “Володимир” для американського індіанця так само є лише набором символів, як для нас “Чингачгук”. У одній із передач “Что, где, когда?” говорилося, що в момент пізнання істини людина відчуває здивування. У нас ідеться саме про це – про процес пізнання через здивування.

Необхідно хоча б коротко сказати про поняття функції. Пригадаймо, як складно було його усвідомити нам самим за класичним означенням у курсі математичного аналізу. Математика розглядає дуже формально і обмежено уявлення про функцію, бо не акцентує, як інформатика, що функція, не лише залежність між двома змінними, при якій кожному значенню однієї відповідає єдине значення другої, а й дія, яка певне значення аргументу перетворює у цілком визначене значення функції. Для учнів цікаво, що термін “дія” тут слід розуміти, як “автомат”, подібно тому, як електром'ясорубка – це автомат для перетворення м'яса у фарш. У процесі таких міркувань у свідомості учнів виникає колосальний світоглядний скачок, багато різних за формою явищ зводяться, так би мовити, “до спільного знаменника”. А якщо пригадати, що математика, навіть вища, практично обмежується розглядом функцій двох числових аргументів, а інформатика, точніше розділ “Алгоритмізація і програмування”, розглядає поняття функції багатьох аргументів, причому, не тільки числових типів, а саме поняття функції у певному розумінні зводиться до поняття процедури, то можна сказати, що абстрактне математичне поняття функції в інформатиці набуває живий, зрозумілий без складного математичного апарату образ. От і спробуй не погодитись, що інформатика, як сказано вище, дуже доречно доповнює математику у “приведенні розуму до ладу”, тобто у формуванні наукового світогляду.

Так склалось, що сьогодні домінують дискретні способи зберігання, опрацювання і передачі інформації з допомогою комп'ютерних технологій, а тому інформатика, не зважаючи на свій дуже молодий вік, опинилась буквально в центрі уваги. Хочеться акцентувати на тому, що *курс інформатики поклика-*

ний не тільки забезпечувати реалізацію дуже важливої функції цементування в єдине ціле всієї сукупності сучасних базових знань, а й формувати окремі компоненти загальної освіти - логічне, алгоритмічне і структурне мислення, необхідні тепер будь-якій пересічній людині. Можна уявляти всю сукупність знань разом із їх складовими, необхідними сучасній людині, у вигляді діаграм або кругів Ейлера (див. мал.), де інформатика відіграє роль і окремого компоненту, і пов'язуючого матеріалу.



Безперечно, ідучи в ногу з часом, слід враховувати стрімке удосконалення обчислювальної техніки, доступність сучасних комп'ютерів, нове і ефективне програмне забезпечення, ширше використовувати дидактичні можливості інформаційних технологій. Але сподіваємось, що вищенаведені міркування переконали читача, що шкільну інформатику ні в якому випадку не слід зводити тільки до курсу користувача. Глибоко помилкова також думка, яку висловлюють інколи навіть авторитетні та знані люди, про те, що на часі зміщувати акценти від так званої "предметизації", характерної для старої, ще радянської школи, до формування сучасної інформаційної культури, яка повинна забезпечити оперативність добування і використання інформації. Але, відштовхуючись від уявлення про комп'ютер, як інструмент для інтенсифікації та автоматизації

розумової діяльності, слід не забувати, що крім стандартних функцій, пов'язаних із використанням ППЗ загального призначення, він володіє важливою властивістю – здатністю до розширення обсягу власних можливостей завдяки програмуванню. У зв'язку з цим *необхідно не скорочувати, а, навпаки, нарощувати присутність алгоритмізації, причому, як шляхом збільшення відведеної кількості навчальних годин, так і наданням їй домінуючої функції у всіх темах курсу.*

Складається враження, що в Україні прийнято концепцію інформатизації суспільства та стандарти в освітній галузі “Інформатика“ не для керівництва а так, щоб було, бо діючі програми, особливо, призначені для звичайних класів та пересічних учнів, мало відповідають і концепції, і стандартам. Так, згідно “Державного стандарту загальної середньої освіти в Україні“, освітня галузь “Інформатика“ до основної школи включено розділ “Алгоритмізація і програмування“, що передбачає такий обсяг вивчення:

Поняття алгоритму. Основні властивості алгоритмів. Способи опису алгоритмів. Виконавець алгоритму. Базові структури алгоритмів. Алгоритмічні мови. Навчальна алгоритмічна мова. Основні елементи мови (символи, слова, вирази, команди). Опис алгоритмів навчальною алгоритмічною мовою. Величини та їх типи. Опис алгоритмів роботи з величинами. Алгоритми створення і опрацювання графічних зображень. Програма і мови програмування. Поняття транслятора. Інтегровані середовища програмування. Мова програмування Алфавіт. Основні поняття мови: ідентифікатори, числа, рядки, описи, оператори. Структура програми. Типи даних. Вирази. Оператори. Оператор присвоювання. Оператори введення, виведення. Опис вказівок повторення і розгалуження мовою програмування. Опис умов. Структурний підхід до розробки алгоритмів і програм. Процедури і функції. Використання готових програм з бібліотеки. Графічний екран. Вказівки для роботи в графічному режимі. Масиви. Алгоритми і програми роботи з масивами. Методи впорядкування та пошуку елементів лінійного масиву. Робота з рядками.

До цього ж розділу в старшій школі віднесено:

Системи візуального програмування. Об'єктно-орієнтоване програмування. Поняття про числення висловлень. Поняття предиката, кванторів. Процедурне і декларативне програмування. Поняття про логічне програмування.

Давайте зіставимо вищенаведене з діючою програмою курсу основ інформатики (універсальний профіль) і спробуємо дати відповідь, чи можливо за нею

забезпечити виконання державного стандарту з інформатики. Ось що пропонується в доданому до цієї програми орієнтовному календарному плануванні (візьмемо обсяг вибраного навчання одного з уроків, наприклад, № 7.9 тієї ж теми “Алгоритмізація і програмування”):

Алфавіт мови програмування. Величини та їх опис мовою програмування. Змінні та константи. Ідентифікатори величин. Типи величин. Вказівки введення й виведення. Вказівка присвоювання.

Зрозуміло, що за 45 хвилин ніяк не можливо дати одночасно поняття про алфавіт мови програмування, весь спектр величин з характеристиками, їх оголошення та опис, команди введення, виведення та присвоювання і практично закріпити це на прикладах лінійних програм.

Який вихід із такого становища? У Щасливському НВК вирішено його наступним чином. У зв'язку із переходом до системи профільного навчання у 10-11 класах, згідно діючих програм з основ інформатики з'явилась можливість перемістити базовий курс інформатики у 7-9 класи (*Програми для загальноосвітніх навчальних закладів Інформатика 7-9 класи. Автори: Жалдак М.І., Морзе Н.В., Науменко Г.Г.*). Базовий курс за цією програмою передбачає 105 навчальних годин замість 70 (по 35 годин у 7-9 класах). Для профільного вивчення інформатики у 10-11 класах вибрано фізико-математичний профіль (по 2 години). Таким чином для вивчення курсу інформатики у Щасливському НВК відведено всього 245 навчальних годин у 7-11 класах. Щоправда це стосується тільки спеціалізованих класів фізико-математичного профілю (5-9 класи) та ліцейних класів (10-11 класи). У загальноосвітніх класах вивчення інформатики здійснюється згідно програми універсального профілю в 10-11 класах. Проте, збільшивши кількість навчальних годин на вивчення інформатики за рахунок шкільного компоненту, або ввівши факультатив по профільному вивченню інформатики за фізико-математичним профілем, описану схему можна впровадити і у загальноосвітніх класах.

Але повернемося до класів фізико-математичного профілю. Перш за все, слід зауважити, що у Щасливському НВК фізико-математичними вони є у великій мірі умовно через незначні можливості конкурсного відбору та ротації

учнів. Тому для вивчення курсу інформатики взято фізико-математичний профіль, а не якийсь інший, з більшою кількістю тижневих годин і потужнішою програмою. Всі, хто знайомі з програмою з інформатики згідно фізико-математичного профілю, помітили, що вона не дуже відрізняється від базового курсу, особливо в частині так званого курсу користувача. Тому виникла проблема уникнення дублювання навчального матеріалу у 7-9 класах (базовий курс) і 10-11 класах (профільний курс). У той же час природно вважати вивчення інформатики в 7-11 класах, як єдине ціле. Отже була складена єдина програма для 7-9-11 класів фізико-математичного профілю на базі двох вищезгаданих програм. Години, вивільнені за рахунок уникнення дублювання, були надані для більш детального вивчення курсу алгоритмізації і програмування, частково розподілені між іншими найважливішими темами, наприклад, вивченням Інтернету (введено ознайомлення з FrontPage та Macromedia), а також включено додаткові, на наш погляд, сьогодні актуальні теми, зокрема ознайомлення з настільними видавничими системами на прикладі Publisher, ознайомлення з комп'ютерною графікою, зокрема CorelDraw та Adobe Photoshop. У розділі "Алгоритмізація і програмування" з'явилась можливість детально познайомити учнів не тільки з традиційним для середніх навчальних закладів структурним програмуванням, а й візуальним проектуванням (ВП) та об'єктно-орієнтованим програмуванням (ООП), дати уявлення про модульне програмування та створення додатків для операційної системи Windows. Ця програма складається із таких тем:

7 клас.

1. *Вступ. Історичні відомості про розвиток інформатики (1 год.)*
2. *Інформація та інформаційні процеси (5 год.)*
3. *Інформаційна система (3 год.)*
4. *Операційні системи (9 год.)*
5. *Програмні засоби загального призначення (16 год.)*
 - 5.1. *Графічний редактор (3 год.)*
 - 5.2. *Текстовий редактор (11 год.)*
6. *Комп'ютерні мережі. Електронна пошта (3 год.)*

8 клас

7. Комп'ютерні презентації (5 год.)
8. Програмні засоби навчального призначення (5 год.)
9. Електронні таблиці (12 год.)
10. Телекомунікаційні системи. Інтернет. (10 год.)
- Резерв навчального часу (3 год.)

9 клас

11. Бази даних. Системи управління базами даних (12 год.)
12. Основи алгоритмізації та програмування. Вступ (1 год.)
13. Алгоритми й алгоритмічні структури (3 год.)
14. Знайомство з навчальною алгоритмічною мовою (6 год.)
15. Складені команди та структуровані дані (12 год.)

10 клас

1. Середовище програмування Delphi. Робота з консольним додатком (5 год.)
2. Процедури і функції (9 год.)
3. Структуровані величини (28 год.)
 - 3.1 Повторення масивів (8 год.)
 - 3.2 Рядкові величини (5 год.)
 - 3.3 Файли (5 год.)
 - 3.4 Множини (5 год.)
 - 3.5 Записи (5 год.)
4. Динамічна пам'ять (7 год.)
5. Практикум по складанню алгоритмів та програм (6 год.)
6. Візуальне проектування (год.)

11 клас

7. Мова програмування Object Pascal (16 год.)
8. HTML-конструювання (27 год.)
9. Видавничі системи. Microsoft Office Publisher 2003 (9 год.)
10. Комп'ютерна графіка. (18 год.)

Програма передбачає дотримання усіх традиційних вимог державних стандартів загальної середньої освіти в Україні стосовно освітньої галузі “Інформатика” щодо обсягу знань, умінь і навичок.

При вивченні курсу інформатики згідно цієї програми доцільно дотримуватись, на наш погляд, таких важливих принципів:

- 1. забезпечення міжпредметних зв'язків, у першу чергу з математикою;*
- 2. єдності всіх тем курсу, незалежно від того, відносяться вони до курсу користувача, чи алгоритмізації і програмування;*
- 3. формування сучасної інформаційної культури та усвідомлення предмету інформатики, як невід'ємної частини загальної системи знань;*
- 4. орієнтації учнів на індивідуальну дослідницьку роботу при вивченні предмету.*

Коротко опишемо ці принципи. Перший полягає у практичній реалізації вищезгаданої тези академіка Єршова, що інформатика повинна іти від математики. В процесі вивчення всього курсу слід максимально можливо застосовувати математичні задачі, уроки інформатики при будь-якій нагоді проводити комбіновано з математикою. Що мається на увазі? Наприклад, вивчення теми № 2 (7 клас), де розглядається кодування інформації, можна поєднати із вивченням систем числення, особливо детально - двійкової системи, з розглядом алгоритмів переведення чисел та перевіркою з допомогою стандартного додатка Windows “Калькулятор”. При вивченні теми 5.2 (7 клас), зокрема введення поняття фреймів у текстовому документі варто зосередитись на послужі Equation. Бажаємо виробити навички запису математичних виразів, наприклад ланцюгових дробів. При цьому учні бачитимуть великі перспективи вивчення текстового редактора Microsoft Word. При вивченні теми № 7 (8 клас) для практичної роботи можна вибрати створення навчальної презентації, наприклад, на опис вписаних і описаних чотирикутників. При вивченні теми № 9 (8 клас) дуже ефектно виглядатиме створення математичних таблиць значень тригонометричних функцій на зразок відомих таблиць Бредіса. При цьому на уроках інформатики не зашкодить розгляд уявлення про радіанну міру кута та формули переведення між градусною і радіанною мірами, що буде зайвим зразком поєд-

нання інформатики та математики. При вивченні логічних функцій у Microsoft Excel природно складати таблиці розв'язування квадратних і бікватратних рівнянь.

Необхідність другого принципу випливає із того, що традиційний курс інформатики штучно і не оправдано поділений на дві практично непересічні та надто непропорційні частини – ”Курс користувача” та ”Алгоритмізація і програмування”, у яких, до того ж, дуже мало передбачено практичне застосування та інтеграцію із іншими предметами, в першу чергу з математикою. Уявіть собі курси алгебри і геометрії, які вивчаються не синхронно і взаємопов'язано, без достатнього практичного застосування, без зв'язків із фізикою. Це повний абсурд. В інформатиці також очевидна доцільність паралельного і взаємопов'язаного вивчення обох складових курсу, узгодження з програмами з математики, фізики, української, російської та англійської мов, як наслідок, можливість широкого використання бінарних уроків інформатики з цими предметами. Отже, ми дотримуємось думки, що *курс інформатики повинен бути цільним, при домінуванні математичної компоненти.*

Зупинимось на третьому принципі. Необхідно поставити акцент на понятті “інформаційна культура” (означати це поняття немає потреби, досить зауважити, що його можна вважати *узагальненим предметом вивчення наукової галузі “інформатика”*). Разом з іншими видами (матеріальної, духовної, інтелектуальної, фізичної і т.д.) інформаційна культура повинна відповідати епосі, тому кожне суспільство завжди її модернізувало, адаптувало до своїх потреб і в цьому розумінні інформаційну культуру слід розглядати, як консервативне поняття. Але сьогодні, в зв'язку із стрімкою трансформацією самого середовища нашого існування, в інформаційний простір, ми не можемо чекати, коли нова інформаційна культура поступово з'явиться, так би мовити, еволюційним шляхом, її слід формувати дуже швидко і оперативно, адже вона, власне, для молодого покоління вже існує, щоправда в дикому, стихійному вигляді, що є небезпечним. Майбутнє людства, як природної популяції, прямо залежить від набутої кожним із нас в процесі досвіду інформаційної культури та її відповідності із сучасними глобальними вимогами до неї.

Особливої уваги вартий четвертий принцип, адже він є загальнопедагогічним. Безсумнівно, що учитель не здібніший чи талановитіший за своїх учнів, він тільки має більше знань та досвіду. Тому його *право учити* - сумнівне. Крім того, по великому рахунку, *навчити неможливо, можливо лише навчитись*. Звідси роль учителя можна звести до трьох основних функцій: *стимулювання шляхом зацікавлення, орієнтація на самостійне удосконалення і консультування*. А процес самостійного удосконалення, або самонавчання зводиться до пошуків запитань та відповідей на них.

Як відомо, сьогодні інформатика викладається переважно у машинному варіанті, причому всі уроки – практичні, тобто на кожному з них учні частину часу працюють за ПК. Сьогодні дуже модні розмови про інформаційні технології, використання ліцензованих програмних навчальних засобів (ПНЗ). Та на практиці рідко застосовують готові ПНЗ. На жаль всі, навіть найкращі, ліцензовані ПНЗ мають три дуже суттєві хиби. Перша – *консервативність*, адже їх важко адаптувати до потреб учителя з конкретними педагогічними ситуаціями, навпаки, використання ПНЗ змушує учителя самому пристосовуватись до них. А як тоді бути із творчим характером навчального процесу? Друга – *переважно інформаційний характер ПНЗ*. Учителю ж необхідні динамічні, із експериментальними, а не тільки демонстраційними та тестуючими можливостями програмні засоби. Нарешті, третя - *всеоб'ємність*, тобто поверхове охоплення всього курсу. Якщо мати на увазі уроки інформатики, то вони по своїй сутності є "живим" спілкуванням з комп'ютером.

Але роль унаочнення у процесі навчання також важлива. Тому вихід бачиться у виробленні деякої "золотої середини", з одного боку якої - громіздке, позбавлене динамічності ПНЗ, з іншого – безпосереднє використання стандартних додатків Window та додатків пакету Microsoft Office. Ми бачимо таку "середину" в так званих *комплексних електронних дидактичних засобах (КЕДЗ)*¹, під якими розуміємо файли чи групи файлів, що містить будь-яку інформацію навчального характеру, дидактичний або наочний засіб, що розсилається на РМУ через локальну мережу для використання учнями під час роботи з

¹ Назва "*комплексний електронний дидактичний засіб*" – це умовний робочий термін, введений автором.

комп'ютером. КЕДЗ не слід плутати із ліцензованими програмними продуктами навчального призначення, бо він виготовляється учителем, гнучкий за характером будови та застосування, може оперативно змінюватись, включати гіперпосилання, анімаційні чи звукові ефекти, адаптований до конкретної ситуації, має, як правило, локальний зміст, володіє багатофункціональністю. Прикладами КЕДЗ можуть бути таблиці та опорні схеми, електронні конспекти до уроків чи тем, презентації PowerPoint тощо. Перш за все це документи, які умовно можна назвати *електронними конспектами*. Вони можуть містити теоретичні відомості з необхідними питаннями, завданнями, тестами, бути оснащеними таблицями, схемами, малюнками. Завдяки системі гіперпосилань ці документи будуть мати динамічний характер, а розсилка на РМУ по локальній мережі дасть змогу використовувати їх оперативно і комплексно з системами програмування та навчальними пакетами. Бажано, щоб кожен КЕДЗ охоплював вузьку тему, тоді створювати його легко, при потребі завжди можна доопрацювати чи удосконалити. Крім того, робота з ним урізноманітнить урок, зробить більш динамічним, створить атмосферу позитивної мотивації вивчення навчального матеріалу, дасть можливість відчувати курс інформатики, як єдине ціле, учні майже безперервно працюватимуть з комп'ютером, не відчуваючи нав'язливої ролі учителя. Такі форми роботи на уроці будуть спонукати і вчителя підходити до більшості тем не формально, строго відповідно програмі, а із спрямуванням на наступне практичне застосування, все щойно вивчене при першій нагоді використовувати практично.

Логічною вершиною курсу інформатики є вивчення візуального проектування та об'єктного програмування, що, як видно із характеристики нашої програми, включено в курс "Алгоритмізація і програмування". Спочатку, у 9-му класі вивчали Visual Basic, це давало вихід на створення макрокоманд у пакеті Microsoft Office. Та через два роки я прийшов до висновку, що слід зосередитись виключно на Delphi. По-перше, тоді нема потреби вивчати дві мови програмування – Basic та Pascal, по-друге, можна обійтись без Turbo Pascal, як додатку морально застарілої ОС MS/DOS, адже у Delphi є можливість створювати консольні додатки. Таким чином, у 9-му класі вивчаємо основи алгоритмізації

та візуального проектування, спираючись на навчальні імітатори мови Pascal ALGO, ABC Pascal та систему програмування Delphi, а у 10-у класі проходимо систематичний курс програмування із ознайомленням з мовою програмування Object Pascal та об'єктним програмуванням.

Додамо, що учні, які додатково цікавляться програмуванням, мають можливість вивчати факультативно (у суботи) поглиблений курс програмування на мові Turbo Pascal. Для цього фрагментарно використовуємо програму “Школа олімпійського резерву з програмування” (автор Лисенко Т.І.) та програму “Прикладна математика” для 8-11 класів (автор Рудик О.Б.). Додатковим джерелом для поглибленого вивчення програмування є також Регіональна програма відкритого безперервного навчання обдарованих учнів з інформатики у Київській області “Комп'ютер – шлях до успіху” при КОПОПК. По суті – це постійно-діюча Інтернет-олімпіада.

Вище викладено погляди на шкільну інформатику, як самостійну навчальну дисципліну і, водночас, як невід'ємну складову загальної середньої освіти у найближчій перспективі. Слід мати на увазі, що ми не претендуємо на абсолютність своїх міркувань, навпаки, усвідомлюємо їх полемічність.

Міркування про роль шкільного предмету інформатики були б не повними без окреслення ще й так би мовити “технологічної” ролі інформатики у системі середньої освіти. Нам вона бачиться так. Уроки інформатики повинні бути взірцем по використанню інформаційних технологій навчання. Ще раз наголосимо, що *використання комп'ютера у навчальному процесі лише в якості засобу демонстрацій тільки частково реалізує можливості інформаційних технологій*, бо учень, що є центральним суб'єктом і об'єктом навчально-виховного процесу, при цьому виступає, як пасивний споживач нової інформації, а її добування і опрацювання на сучасному рівні, про що говорилось вище, залишається поза ним. Тому слід погодитись, що *функція ПК, як технічного засобу навчання, є лише супутною послугою*.

Звідси випливає несподіваний і дивний висновок, що *першою і обов'язковою складовою інформаційних технологій навчанняв цілому є повноцінний курс шкільної інформатики*. Якщо ця складова не присутня у повному

обсязі, то всі інші разом узяті не вирішують проблеми. Учням необхідно оволодіти засобами навігації у потоках інформації, а комп'ютер, подібно компасу у морській справі, - можна розглядати тут як головний інструмент.

Згадаймо відомі слова А. Ейнштейна: "Здається майже чудом, що сучасні методи навчання ще не зовсім задушили святу допитливість". Отже, можливо, слід менше опікуватись впровадженням не активізуючих учня демонстраційних форм інформаційних технологій навчання, а краще створювати ширші можливості для мислительської діяльності безпосередньо за дисплеєм, тобто частіше практикувати чисельний експеримент?

Навіть перенесення традиційного програмованого навчання на нові, інформаційні, засоби, є наочним прикладом консервативності системи освіти й однією з причин того, що тривалий (більш ніж 15-річний) етап впровадження комп'ютерів у навчально-виховний процес не дав очікуваних результатів. Точка зору відомого ученого С. Пейперта полягає в тому, що *дитина повинна програмувати комп'ютер і, роблячи це, не тільки опанувати часточкою найсучаснішої техніки, але й прилучається до багатьох глибоких ідей природознавства, математики, а також до мистецтва інтелектуального моделювання*. Резюмуючи, можна ще раз сказати, що комп'ютер повинний бути в першу чергу одним з інструментів розвитку інтелектуальних здібностей дитини.

Деяким позитивним кроком у інформаційних технологіях навчання стала благодійна програма "Інтел[®] Навчання задля майбутнього", в основі якої лежить метод проектів з використанням комп'ютера та інтернету для добування знань. Але форми навчання на базі методу проектів важко вписуються у календарно-тематичні рамки, їх доцільніше використовувати в позаурочній роботі, зокрема в секціях МАН.

Вихід бачиться у такій схемі: *використання на уроках ліцензованих ПНЗ з допомогою проєктуючих засобів ⇒ курс інформатики у вищеприписаному бащенні ⇒ повноцінне використання можливостей систем програмування та офісних пакетів для моделювання навчальних ситуацій і чисельних експериментів на уроках з інших предметів ⇒ дослідницька позаурочна робота з допомогою комп'ютера*.

