

Про вплив інформаційних технологій на технології освіти

Ми завжди намагаємося виділити основні критерії, що відрізняють людину від тварин. І це не тільки спроба у черговий раз реалізувати свої амбіції. Це спроба усвідомити себе для прямування цивілізації вперед. На наш погляд, принциповим аспектом, що виділяє людину в навколишній природі, є її унікальна здатність змінювати себе і навколишній світ. Хоча на різних підрізках історичного розпіткy ці зміни носили як позитивний, так і негативний характери, у цілому ми маємо очевидну позитивну тенденцію. Важливо також відзначити, що швидкість таких змін постійно наростала, а останнім часом набула яскраво вираженого характеру. У зв'язку з цим ми виділяємо основні тини протиріч, що характеризують історичні відрізки часу, на яких швидкість зміни технологій була практично однаковою:

- протиріччя між технологією передавання знань, яку ми називаємо «словом», і носієм знань, цю має короткий період життя;
- протиріччя між кількістю рукописних носіїв знань і кількісно живих людей;
- протиріччя між кількістю розтиражованих знань і можливостями доступу до них;
- протиріччя між можливостями, що налагаються цифровими технологіями одержання знань, і фізіологічними можливостями людини в їхньому освоєнні.

В історії людства кожне покоління намагалось обґрунтувати свою виняткову роль у світовому розвитку І кожне покоління було правим, оскільки у відповідний момент часу воно було останнім. Але в дійсності дуже важливі як в історичних дослідженнях, так і в самому розвитку людської цивілізації саме ті відрізки часу, на яких протиріччя, зазначені вище, одержували свій розв'язок

Ніхто напевно не знає, як і коли з'явилася перша технологія передавання знань, що називається словом, а разом з ним і людина. На нашу думку, в легенді про яблуко Адама і Єви схована таємниця передавання знань однієї цивілізації іншій, причому настільки нетривіальним способом, що потенціал цієї технології і дотепер нами не реалізований.

Але напевно відомо, то рукопис повною мірою вирішив **перше протиріччя** і дозволив людству в наступних 5 тисяч років істотно змінити і себе, і навколишній світ. Поява рукопису призвела до утворення двох нових інститутів передавання знань: бібліотек, де людина мала і має можливість самостійно отримувати знання, та освітніх закладів, в яких зростаючі покоління засвоювали накопичені суспільством знання під керівництвом вчителя.

Ніхто не зможе заперечувати тон факт, що друкарський верстат, винайдений людиною 500 років тому і який дозволив тиражувати знання, вирішив **друге протиріччя** і вплинув на подальші способи розвитку цивілізації. Поява друкарень створила умови для виникнення нового педагогічного інструменту - інструменту передавання знань, формування вмінь та навичок, який сьогодні прийнято називати класно-урочною системою. Ця система, біля джерел якої стояв Ян Амос Каменський, була новою як за змістом, так і за формою, що і дотепер використовується людством як основна технологія поширення знань. Ця історична подія зайвий раз підкреслює ту залежність, яка існує між виробничими технологіями та технологіями освіти.. Зміна одних обов'язково призводить до зміни інших та навпаки, Масове поширення класно-урочної системи була прямо пропорційно пов'язано з кількістю книг, які видавалися.

Протягом цих 500 років кількість знань, яку розтиражовано в книжках, призвело до появи протиріччя, пов'язаного з системами, доступу до інформації, накопиченої цивілізацією. В 60-70 роки ХХ століття стало зрозуміло, що така ситуація потребує рішення через нові технології, якими стали комп'ютерні та мережеві технології.

Сьогодні ми являємось не лише свідками, але й учасниками розв'язання **третього протиріччя**. Дійсно, еліта людської цивілізації, на основі комп'ютерних і глобальних мережевих технологій Підбирається до розкриття таємниць, які можна порівняти з таємницею яблука Адама і Єви. Конкретним підтвердженням сказаного вище є відкриття людського генома. Якщо припустити, що *перше слово* з'явилося 50 тисяч років тому, то ми одержуємо закономірність один до десяти, що визначає змінюваність інформаційних епох і маємо сміливість припустити, що останнє протиріччя знайде розв'язання в найближчі тридцять - сорок років.

Історичний досвід показує, що в рамках розв'язування інформаційних протиріч людство створювало нові технології передавання знань від покоління до покоління. Аналогія підказує, що і сьогодні нам не уникнути не тільки розробки, але і впровадження нових тех-

нологій передавання знань, як на рівні зміни змісту, так і на рівні зміни форми. Процес це об'єктивний, а терміни визначаються рамками двох поколінь, що і визначає специфіку і не дозволяє повною мірою використовувати звичайний досвід. Сказане вище дозволяє виділити особливості, що, на наш погляд, повинні істотно вплинути на розробку і впровадження нових технологій освіти, обумовленим впливом нових інформаційних технологій:

- людство знаходиться на етапі повномасштабного розв'язання **третього інформаційного протиріччя**;
- передбачувані терміни нової історичної епохи глобальної оцифровки знань і мережевих технологій доступу до них визначаються двома - трьома поколіннями;
- мінімальна переносимість історичного досвіду для паралельної зміни технологій освіти;
- велика інертність традиційної системи освіти;
- відсутність гіпотез для розв'язування проблем **четвертого інформаційного протиріччя**.

Урахування цих положень, на наш погляд, дозволить припинити марні суперечки про необхідність переходу до нових технологій навчання, побудованих на основі використання сучасних інформаційних засобів"; дозволить зрозуміти, що поділ праці у світовому виробництві ще не закінчено, а пріоритети сучасної освіти жорстко прив'язані до цифрових технологій; і, нарешті, що прямі інвестиції в нові педагогічні технології єдина альтернатива для виходу нації на передові позиції.

Загальна зміна інформаційно-комунікаційних технологій відбувається настільки швидко, що це не могло не відбитися на освітніх технологіях. Під впливом цих змін виникає необхідність перегляду тих учбових курсів, які вважалися традиційними та майже незмінними. Зростає обсяг професійної інформації і перед викладачами постає питання, що саме і в якому обсязі вивчати зі студентами. Наприклад, в [1] обговорювалися питання змістового наповнення курсу "Основи алгоритмізації та програмування" та необхідність цього курсу взагалі.

Необхідно зауважити, що реформування системи освіти в Україні оголошене пріоритетною задачею держави, передбачає якісні зміни освітніх технологій на базі інформаційних технологій. У зв'язку з цим набувають особливої актуальності задачі реформування системи інформаційної підготовки школярів і випереджувачого реформування системи підготовки вчителів інформатики та інших предметів. Проблеми, пов'язані з підвищенням якості підготовки майбутніх вчителів інформатики завжди були в центрі уваги в науково-методичній роботі кафедри інформаційних технологій Херсонського державного педагогічного університету. Ця частина статті присвячена одній з таких проблем - проблемі якісної зміни змістового компонента вивчення інформатики в педагогічних вузах.

У підготовці майбутнього вчителя інформатики ми виділяємо два компоненти: фундаментальний і прикладний. (В силу різних суб'єктивних і об'єктивних причин у центрі уваги науково-методичних досліджень зараз знаходяться прикладні аспекти. Разом з тим очевидно, що фундаментальна теоретична підготовка сучасного вчителя інформатики принципово важлива для рішення такої стратегічно важливої задачі, як профорієнтація школярів у сфері інформатики).

Однією з дуже важливих задач, зв'язаних зі зміною змістового компонента вивчення інформатики в педагогічних вузах є зміна змісту центрального фундаментального курсу - «Основи алгоритмізації і програмування» (ОАП).

Дійсно, місце і роль цього курсу в системі дисциплін інформаційного циклу унікальні. З одного боку, він перекидає місток до того курсу, що учні вивчали в школі. З іншого боку, він є сполучною ланкою з такими теоретичними дисциплінами, як «Основи математичної логіки і теорії алгоритмів». «Дискретна математика», «Чисельні методи», «Основи теорії імовірностей і математичної статистики»;', базовим курсом для спеціальних дисциплін «Використання обчислювальної техніки в навчальному процесі». І, нарешті, «Методика навчання інформатики в школі».

Рис 1 Структура і взаємозв'язки дисциплін інформаційного циклу.



Рис 1 Структура і взаємозв'язки дисциплін інформаційного циклу

Саме в ході вивчення цього курсу студенти здобувають знання, вміння і навички складання алгоритмів, їхнього опису структурною алгоритмічною мовою і реалізації в конкретній системі програмування у вигляді комп'ютерної програми.

Зауважимо, що цього року можна відзначати 15-тирічний ювілей появи цього курсу. Звернемо також увагу на те, що зміна технологій, пов'язаних з використанням методів інформатики, з удосконалюванням засобів обчислювальної техніки (як технічної, так і програмної компонент), зробивши революційний вплив на прикладні дисципліни інформаційного циклу, практично не призвела до змін в змістовій частині цього фундаментального курсу.

Дійсно, у багатьох педагогічних вузах як споконвічне вивчали питання, пов'язані з даними і способами опрацювання, з типізацією даних, як використовувалися процедурні мови програмування, так і сьогодні ситуація не змінилася. Ядром цього курсу, як і раніше, є вивчення найпростіших конструкцій, зв'язаних із слідуванням, з розгалуженням, з циклами, з подальшим переходом до вивчення статичних структур даних (масиви, записи, файли, множини), динамічних структур даних (списки, стеки, черги, дерева і т. п.) та засобів структурування програм (процедури і функції). Для опису алгоритмів «на змістовому рівні» при цьому часто використовується мова блок-схем.

Таким чином, протягом останніх 15 років не спостерігалися істотні зміни у змістовій частині курсу ОАП. Ця ситуація характерна для 90 % шкіл і педвузів України. На наш погляд, інерція, властива цьому курсу, швидше за все, пов'язана ще і з тим, що рівень засобів обчислювальної техніки педагогічних вузів практично не набагато випереджає те, що відбувається в школах.

Згадуючи і критично оцінюючи історію зміни змістового компонента курсу «Основи алгоритмізації і програмування» педагогічних вузів, хотілося б нагадати, що 1986-1990 роки характеризувалися тим, що вивчення основних структур даних і способів опрацювання в основному базувалися на таких мовах програмування, як мови програмованих калькуляторів, а також Фортран і Бейсік. Дуже важливо звернути увагу і на той факт, що на цьому етапі скоріше виконувалися "Постанови" різного роду регулюючих інстанцій, ніж відбувалося серйозне вивчення курсу програмування і викладання майбутнім учителям тих основ, які вони могли б повною мірою передавати в школах.

Разом з тим необхідно відзначити, що, починаючи з 90-х років, відбувався дуже бурхливий розвиток виробничих мов програмування і технологій програмування. Дійсно, ще в 1990-1991 роках дуже схвальною манерою було вивчення курсу «Основи алгоритмізації й

обчислювальної техніки» на основі мови Pascal. З одного боку, Pascal дозволяв на досить високому методичному і природному рівні вивчати необхідні структури даних і алгоритми опрацювання цих даних. З іншого боку, це мова практичного програмування, яка дозволяла формувати не тільки теоретичні знання, але й учити майбутніх учителів дійсно вирішувати задачі, реально програмуючи й одержуючи необхідні результати за допомогою комп'ютера. Звичайно, на цьому шляху було дуже багато проблем, таких як недостатня кількість машин у ВНЗах, нелегальне програмне забезпечення, і т.п. і о-му лише кращі студенти мали доступ до використання цих технологій. Разом з цим, починаючи десь з 94-95 року, все активніше на зміну так званій структурній методології програмування почали приходити об'єктно-орієнтовані методології програмування (ООП), підтримувані конкретними технологічними рішеннями. Технології ООП докорінно змінили підхід до розв'язування задач з використанням комп'ютера.

Якщо проаналізувати навчальні програми, у тому числі і базові програми Міністерства освіти і науки, з дисципліни «Основи алгоритмізації і програмування», можна побачити, що аспекти об'єктно-орієнтованих технологій практично не знайшли відображення в тих реальних робочих програмах, якими користуються викладачі. Ми не намагаємося критикувати той стан справ, що фактично має місце в даний час.

Припустимо, що педагогічний ВПЗ має нормальну комп'ютерну техніку, сучасне програмне забезпечення і добру методичну базу. Він планує почати навчання майбутніх учителів роботі із сучасники ОС, такими як Windows 98, Windows 2000, Windows-Me, різними мережевими операційними системами. Передбачається також вивчення сучасних прикладних інформаційних технологій як загального призначення, так і професійно орієнтованих (питання, пов'язані з використанням сучасних баз даних, електронних таблиць для розв'язування предметних задач, систем мультимедіа, тощо).

Надамо тепер реальний прогноз перспектив використання цих знань. Учитель, що закінчив такий педагогічний ВНЗ і прийшов на роботу в реальну периферійну школу, власне кажучи, виявляється не затребуваним. Не затребувані його знання, уміння, навички, тому що сьогодні практично 90% шкіл в силу цілого ряду об'єктивних і суб'єктивних причин орієнтуються на підручник А.П.Єршова 1986 року видання, де базовою є російська алгоритмічна мова - з усіма наслідками, то звідси впливають. Хоча парадигма шкільного курсу інформатики вже давно змінилася.

Основною метою шкільного курсу інформатики є не навчити дитину програмувати, а навчити її використовувати сучасні інформаційні технології для розв'язування тих задач, що зустрічаються в повсякденному житті, в різних предметних галузях, у розв'язуванні тих задач, що цікаві учням. У центрі уваги виявилися прикладні аспекти інформатики.

Обговорюючи коло цих проблем, ми ще раз повертаємося до думки про те, що об'єктно-орієнтовані технології не знайшли відображення в курсі «Основи алгоритмізації і програмування».

Отже, перед нами виникає питання, *чи є необхідність сьогодні, в тій реальній ситуації, що склалася не тільки в педвузах, але й в школі, в першу чергу школі периферійній, у вивченні курсу «Об'єктно-орієнтовані технології програмування»* Якщо «Так», то потрібно чітко розуміти, де і коли ці знання будуть використані майбутнім учителем.

Обґрунтування позитивної відповіді на це питання дано на початку статті: країні-виробникові нових інформаційних технологій, а саме такою країною ми хочемо бачити Україну, необхідна система раннього виявлення і професійної орієнтації здатної молоді. Для цього необхідний і фундаментальний компонент шкільного курсу інформатики, і вчителі, професійно підготовлені до його навчання.

Рішення цієї проблеми кафедрою інформаційних технологій було переведено в практичну площину ще 3 роки тому. Саме тоді ми ввели в програму невеликий (від 20 до 40 годин у залежності від спеціальності) курс «Об'єктно-орієнтовані технології програмування». Нам особливо важливо було одержати відповідь на питання: чи заважає та сукупність знань, що ми формуємо в студентів у базовому курсі «Основи алгоритмізації і програмування» наступному вивченню ООП і чи можливо більш ефективно вивчення цих технологій програмування

І ще: якщо ми вирішили, що цей курс необхідний, то виникає друге, не менш важливе питання, на яке не так просто дати відповідь, а саме: *чи є взагалі необхідність у вивченні курсу «Основи алгоритмізації і програмування»*, що, по суті, представляє структурні техно-

логії програмування. І ось тут не можна ігнорувати думку програмістів, що займаються реальним програмуванням, розв'язуванням великих задач. А вони вважають, що знання процедурних технологій програмування, які ми викладали останні 15 років, негативно позначаються на наступному переході до об'єктно-орієнтованої методології програмування.

Існує стійка точка зору: знання, що спираються на структурний підхід у силу своєї специфіки настільки могутньо впливають на мислення, що становляться бар'єром для сприйняття концепцій ООП, отже, і" на використання 00 методологій програмування. Можливо, ми дійсно підійшли до того моменту, коли варто відмовитися від процедурних технологій програмування і відразу ж починати вивчення основ алгоритмізації і програмування з вивчення об'єктно-орієнтованих технологій.

Чи потрібний той традиційний курс, побудований на базі стандартної мови Pascal, що містить в собі лише елементи об'єктно-орієнтованих технологій і дозволяє працювати з об'єктними типами даних? Чи є необхідність вивчення цього курсу? Якщо є, то яким чином негативно чи позитивно, цей курс буде вливати на вивчення об'єктно-орієнтованих технологій. Можливо, дійсно має сенс починати вивчення програмування з курсу ООП. Адже в самім визначенні об'єкта ми говоримо, що усе, що оточує нас, є об'єкти.

На наш погляд, є необхідність у вивченні і того, і іншого курсу як двох взаємодоповнюючих курсів. Разом ці курси покликані допомогти майбутньому учителю визначитися з тією задачею, що стоїть перед ним, зробити правильний вибір у методі, що дозволяє вирішити дану задачу з використанням комп'ютера. Звичайно, існує великий клас задач, природним чином виникаючих перед вчителем і таких, розв'язки яких може бути отримано і без застосування програмування. Він може просто використовувати той чи інший інструментальний засіб. Наприклад, Excel підтримує розв'язування широкого класу обчислювальних задач, включаючи задачі оптимізації, а Access підтримує вирішення проблем, пов'язаних з використанням простих баз даних. Але зараз мова йде про фундаментальні знання - знання методів програмування. Тут ми визначаємо два блоки знань, умінь, навичок:

перший блок - структурні технології програмування або, як кажуть, "програмування в малому";

другий блок - наступний за ним природним чином - це блок об'єктно-орієнтованих технологій або, як кажуть "програмування об'єктів і систем".

Дійсно, у першій частині ми формуємо базові знання про дані, структури даних, про способи їх опрацювання, про вибір найбільш раціональних алгоритмів для розв'язування стандартних задач, які вивчаються локалізовані задачі, які не потребують побудови деякої великої архітектури, побудови дуже складних структур даних. Ці задачі будуть вирішуватись засобами звичайної мови Pascal, що припускає використання структурних технологій і не вимагає додаткових ресурсів.

Разом з тим стає очевидним, що цих знань явно недостатньо для написання «великих» прикладних програм. Тобто, коли ми підіймаємося на крок угору, ми попадаємо в світ реальних систем взаємодіючих •, '7і'ктів Саме такі системи нас оточують і їхнє моделювання передбачає наявність складних рішень. Те, що ми робили дотепер всього лише деякий компонент, одна з невеликих підзадач, це всього лише кілька простих об'єктів чи методів, то описують розв'язування цієї конкретної задачі. І в цій ситуації нам потрібно відмовитися від використання структурної технології і використовувати зовсім іншу і очку зору на розв'язування задач, де ми описуємо об'єкти, їхню поведінку, взаємозв'язок і на цій основі реалізуємо систему.

Сьогодні існує великий якісний розрив між комп'ютерними програмами, що студенти пишуть самі, і тими, якими вони користуються на практиці, працюючи за сучасним комп'ютером. Ще кілька років тому цей розрив можна було усунути, використовуючи методи модульного програмування і стандартні модулі. Зараз, в результаті переходу до сучасних об'єктно-орієнтованих операційних систем типу Windows і до відповідних об'єктно-орієнтованих технологій, цей розрив уже не можна подолати, залишаючись в рамках старих методів і технологій програмування. При цьому ООП усе таки є логічним продовженням і розвитком процедурного програмування.

Розподіл зв'язків між курсами ОАП і ООП виявляється наступним: в ОАП вивчаються методи реалізації поведінки окремих об'єктів, а в ООП - методи реалізації їхньої взаємодії в процесі функціонування програмної системи.

Таким чином, ми все-таки є прихильниками того, що курси ОАП і ООП повинні слідувати один за одним не як взаємовиключаючі, а як взаємодоповнюючі курси, де дуже

важливо виробити правильне почуття вибору, .те сам студент - майбутній учитель, має можливість визначити, що саме він повинний обрати інструментом для розв'язування поставленої задачі. Досвід цих трьох років показав, що такий варіант є найбільш раціональним у зв'язку з цим було розроблено два навчальних посібники, то позначені відповідним грифом Міністерства освіти і науки: перший посібник – *“Основи алгоритмізації і програмування”*, що представляє традиційний курс «Основи алгоритмізації і програмування». Цей посібник використовувався протягом декількох років окремо. Зовсім недавно, буквально два місяці чому. вийшов в світ другий посібник «Введение в объектно-ориентированное программирование», задуманий і реалізований як природне продовження першого. Цей посібник було видано російською мовою. Зараз в друкарні знаходиться український варіант книги.

Таким чином, паш погляд на взаємозв'язки ОАП і ООП, викладений вище. виявився втіленим у двох цих послідовних посібниках-частинах. Необхідно звернути увагу на те, що другий посібник «Введение в ООП» базується па різних предметних областях. Практично всі приклади та вправи, запропоновані там. не вимагають специфічних знань математики, і це - одна з центральних методичних ідей побудови даного навчального посібника. Книга містить приклади, задачі і вправи, що доступні і філологам, і хімікам, і біологам, і історикам, і представникам інших гуманітарних спеціальностей.

Посібник побудований таким чином, щоб студенти, що сьогодні навчаються в педагогічному Вузі і які мають невдовзі працювати в школах, мали можливість використовувати об'єктно-орієнтовані технології у своїй професійній діяльності, не спираючись на знання математики.

На закінчення відзначимо, що проблеми, поставлені в даній статті носять комплексний характер. Їхнє вирішення не може бути отримане «відразу і цілком». Однак, на наш погляд, ці проблеми настільки актуальні і важливі, що будь-яке їхнє обговорення, тим більше просування вперед, заслуговує на увагу.

Література

1. Львов М.С., Співаковський О.В., Про зміст курсу "Основи алгоритмізації та програмування"// Комп'ютер у школі та сім'ї. 2001. № 4. -с.22-25.
2. Львов М.С., Співаковський О.В., Основи алгоритмізації та програмування. навчальний посібник. - Херсон: Айлант 2001, - ІІ'2 с.
3. Львов М.С., Спываковский А.В., Введение в объективно-ориентированное программирование: учебное пособие. - Херсон: Айлант 2000,-210с.