

**Шляхи удосконалення курсу «Основи алгоритмізації та програмування» у педагогічному вузі.**

Реформування системи освіти в Україні, оголошене пріоритетною задачею держави, передбачає якісну зміну освітніх технологій на базі інформаційних технологій. У зв'язку з цим здобувають особливу актуальність задачі реформування системи інформаційної підготовки школярів, і випереджуючого реформування системи підготовки вчителів інформатики. 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 стаття присвячена одній з таких проблем – проблемі якісної зміни змістовного 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 змістовного 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 ймовірностей і математичної статистики», базовим курсом для спеціальних дисциплін «Використання обчислювальної техніки в навчальному процесі», і, нарешті, «Методика викладання інформатики в школі».

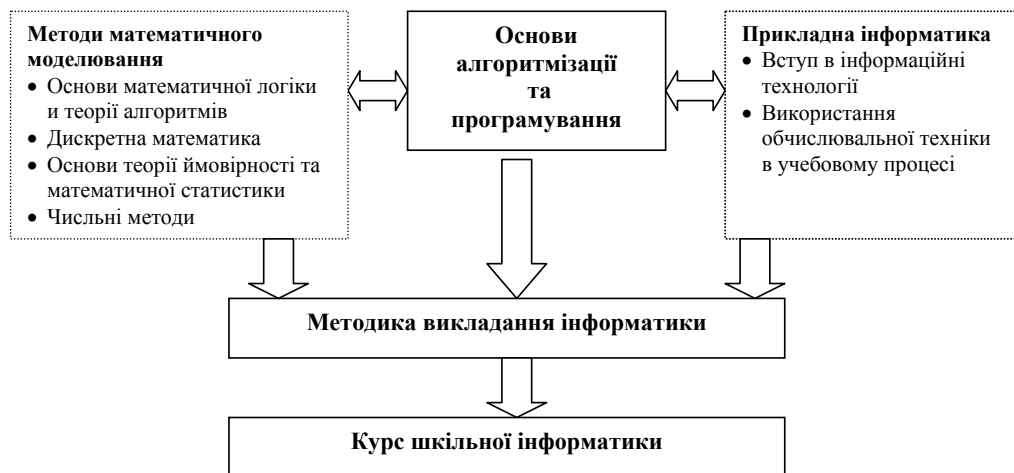


Рис 1. Структура і взаємозв'язки дисциплін інформаційного циклу

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.

Відмітимо, 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, не практично не призвела до змін в змістовній частині цього фундаментального курсу.*

Дійсно, у багатьох педагогічних вузах як споконвічно вивчали питання, пов'язані з даними і способами їхньої обробки, 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 наслідуванням, 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 змістовному рівні» при цьому часто використовується мова блок-схем.

Таким чином, протягом останніх 15 років не спостерігалися істотні зміни у змістовній 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 випереджали те, що відбувалося в школах.

Згадуючи і критично оцінюючи історію зміни змістовного компонента курсу «Основи алгоритмізації і програмування» педагогічних вузів, хотілося б нагадати, що 1986-1990 роки характеризувалися тим, що вивчення основних структур даних і способів їхньої обробки в основному базувалися на таких мовах програмування, як мови програмування 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.

**Примечание [К.1]:**  
Заменить на – преподавание  
будущим учителям ?

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, що обробляють ці дані. 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. Тому 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 в корні змінили підхід до рішення 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 користаються викладачі. Ми не намагаємося критикувати те положення справ, що фактично має місце в даний час.

Припустимо, що педагогічний ВНЗ має нормальну комп'ютерну техніку, сучасне програмне забезпечення і добру методичну базу. Він планує почати навчання майбутніх учителів роботі із сучасними 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, зв'язані з використанням сучасних баз даних, електронних таблиць для рішення 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 незатребуваним. Не затребувані його знання, уміння, навички, тому що сьогодні практично 90% шкіл дотепер у 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, що звідси випливають. 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 рішення тих задач, що зустрічаються в повсякденному житті, предметній області, у рішенні тих задач, що їй цікаві. 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 витребувані майбутнім учителем.

Обґрунтування позитивної відповіді на це питання дано на початку статті: країні-виробнику 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 викладання.

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 викладанні курсу «Основи алгоритмізації і програмування»*, що, по суті, представляє структурні технології програмування. І ось тут не можна ігнорувати думку програмістів, що займаються реальним програмуванням, рішенням 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 нам використання ОО 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? Є чи необхідність вивчення цього курсу? Якщо є, то яким чином, негативно чи позитивно, цей курс буде впливати на вивчення об'єктно-орієнтованих технологій. Можливо, дійсно має сенс починати вивчення програмування з курсу ООП. Адже в самому 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 образом виникаючих перед вчителем і таких, рішення яких може бути отримано і без застосування програмування. Він може просто використовувати той чи інший інструментальний засіб. Наприклад, Excel підтримує рішення широкого класу обчислювальних задач, включаючи задачі оптимізації, а Access вирішує проблеми, пов'язані з експлуатацією 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 образом – 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 оптимальних алгоритмів для рішення стандартних задач. У ньому 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 у гору, ми попадаємо в світ реальних систем взаємодіючих об'єктів. Саме такі системи нас оточують і їхнє моделювання передбачає наявність складних рішень. Те, що ми робили дотепер, у цьому світі, всього лише деякий компонент, одна з невеликих підзадач, це всього лише кілька простих об'єктів чи методів, що описують рішення цієї конкретної задачі. І в цій ситуації нам потрібно відмовитися від використання структурної технології і використовувати зовсім іншу точку зору на рішення 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 назад 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 обов'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 взаємовиключні, а взаємодоповнюючі курси, де дуже важливо виробити правильне почуття вибору, де сам студент – майбутній учитель, має можливість визначити, що саме він повинний обрати інструментом для рішення

поставленої задачі. Досвід цих трьох років показав, що такий варіант є найбільш оптимальним і, у зв'язку з цим було розроблено два навчальних посібники, що 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 тому, 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, наш 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 на різних предметних областях. Практично всі приклади та вправи, запропоновані там, не вимагають специфічних знань математики і це – одна з центральних методичних ідей побудови нашого 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 ВНЗі 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 дійсній статті, носять комплексний характер. Їхнє рішення 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., Спиваковский О.В., Основы алгоритмизации та програмування, навчальний посібник. – Херсон: Айлант, – 122 с.
2. Львов М.С., Спиваковский А.В., Введение в объектно-ориентированное программирование: учебное пособие. – Херсон: Айлант, – 210 с.