

**ОСОБЛИВОСТІ ВИКЛАДАННЯ ПРОГРАМУВАННЯ У ВИЩОМУ НАЧАЛЬНОМУ ЗАКЛАДІ З
ВРАХУВАННЯМ ВИМОГ СУЧАСНОСТІ**

В статті розглянуто основні особливості викладання курсу програмування з врахуванням існуючих стратегій та психолого-педагогічних особливостей навчання студентів. Проаналізовано досвід професійних товариств ACM та IEEE Computer Society по створенню рекомендацій щодо викладання інформатики в університетах. Окреслено базовий рівень математичної підготовки майбутнього вчителя інформатики.

Постановка проблеми. На початку третього тисячоліття інформатика стає надзвичайно актуальною наукою. З моменту своєї появи вона перетворилась на невід'ємну частину сучасної культури, комп'ютери є рушійною силою економічного росту в усьому світі. Більш того, ця галузь продовжує розвиватися з вражаючою швидкістю. Постійно з'являються нові технології, а існуючі технології стають застарілими майже зразу після виникнення.

Аналіз останніх досліджень. Історично склалося, що українські дослідження в галузі навчання інформатики та використання ІКТ в навчальному процесі, які проводилися під керівництвом провідних науковців В. Ю. Бикова, М. І. Жалдака, Н. В. Морзе, Ю. С. Рамського та інших, розвивались певний час самостійно, незалежно від світових розробок. В одній із своїх праць М. І. Жалдак наголошує, що "в основу інформатизації навчального процесу варто покласти створення та широке впровадження в повсякденну педагогічну практику нових комп'ютерно-орієнтованих методичних систем навчання на принципах ... гармонійного поєднання традицій і комп'ютерно-орієнтованих технологій навчання, не заперечування і відкидання здобутків педагогічної науки минулого, а, навпаки, їх удосконалення і посилення ..." [1: 8].

Проте, зусилля закордонних науковців зі стандартизації навчання інформатики заслуговують уваги. Одним з яскравих проєктів у цій галузі є створення документа Computing Curricula ("Рекомендації щодо викладання інформатики в університетах"). Перша версія цього документа була створена спеціальним комітетом з освіти професійного товариства ACM (Association for Computing Machinery) та опублікована в 1968 році. У 70-х роках минулого століття аналогічний документ опублікувало інше професійне товариство – IEEE Computer Society. Пізніше, ці два товариства об'єднали зусилля, і в 1991 році вийшла в світ оновлена версія рекомендацій Computing Curricula'91. У грудні 2001 року був опублікований останній варіант Computing Curricula 2001. У межах такого проєкту було розроблено вказівки зі складання навчальних планів з інформатики (computer science) для зарубіжних університетів та коледжів [2; 3; 4].

Метою статті є висвітлення особливостей викладання курсу програмування з врахуванням існуючих стратегій та базової математичної підготовки майбутніх учителів інформатики.

Постановка проблеми. Швидкі темпи розвитку інформаційних технологій, парадигм програмування, вдосконалення комп'ютерної техніки, збільшення інформаційного потоку та необхідності його опрацювання змусили провідних науковців переглянути підходи в галузі викладання як програмування окремо так і інформатики загалом. Швидка еволюція інформатики як дисципліни вплинула не тільки на зміст предметів, що викладають, а й на педагогічні методи та технології, що знайшло відображення в документах ЮНЕСКО: "педагогічні технології – це системний підхід створення, застосування та визначення всього процесу викладання та засвоєння знань з врахуванням технічних і людських ресурсів, їх взаємодії, що мають на меті оптимізацію освіти" [5].

Одним із перших, хто в 70-80 роках минулого століття почав працювати над питанням методики викладання інформатики, був академік А. П. Єршов. Його відомі праці "Програмування – друга освіта" та "Звідки беруться люди, що можуть створювати надійне програмне забезпечення" створили фундамент для методики викладання інформатики як науки в Радянському Союзі. Подальше активне обговорення питань викладання інформатики призвело до створення та впровадження державних освітніх стандартів, що відобразили існуючі на той час бачення предмета та відповідних йому знань.

Навчання майбутнього фахівця – це вплив на свідомість та діяльність з метою підготовки до майбутньої професійної діяльності. Проте результати навчання не повинні обмежуватись формуванням лише професійних знань, вмінь та навичок. У процесі навчання повинні формуватися ключові компетентності, інтелектуальні якості – формується особистість майбутнього фахівця, як єдине ціле [6: 31].

Ряд педагогів та науковців у галузі дидактики вищої школи пропонують свої системи принципів навчання. При цьому одні з них переносять принципи загальної (шкільної) дидактики в умови ВНЗ, дещо змінюючи та уточнюючи їх формулювання. Автор однієї з перших монографій, що присвячена навчальному процесу в ВНЗ, радянський науковець С. І. Зінов'єв, сформулював наступні принципи дидактики вищої школи: науковість; зв'язок теорії з практикою, практичного досвіду з наукою; системність та послідовність в підготовці фахівців; свідомість, активність та самостійність студентів в навчанні; поєднання індивідуального пошуку знань з навчальною роботою в колективі; поєднання абстрактного мислення з наочністю у викладанні; доступність наукових знань; ґрунтовність засвоєних знань [7: 6].

Ці принципи були перенесені з дидактики середньої школи так як дидактика вищої школи і дидактика середньої школи по своїй суті відповідають на наступні питання: як навчати, чому навчати, для чого навчати? Проте були враховані особливості організації навчального процесу цієї групи навчальних закладів – у вищому навчальному закладі вивчаються не основи наук, а сама наука в розвитку; залучення студентів до науково-дослідницької роботи кафедр та наукових лабораторій; поєднання наукового та навчального компонентів в діяльності викладачів; орієнтація на майбутню професію закладено у викладанні переважної більшості дисциплін; більший відсоток самостійної роботи студентів; інші форми організації навчального процесу тощо.

За роки свого існування більшість курсів з інформатики фокусувалися переважно на набутті навичок програмування. На поширення такого підходу вплинула низка практичних та історичних факторів:

- вміння програмувати є обов'язковим для всіх студентів, що навчаються за напрямком "Інформатика". Оволодіння програмуванням на перших курсах у певній мірі гарантує наявність необхідних знань у переході до поглиблених курсів;
- інформатика стала академічною дисципліною досить пізно, і тому вона довгий час асоціювалася з програмуванням;
- модель "з орієнтацією на програмування" була підтримана ранніми програмами з "Інформатики".

Проте підхід з орієнтацією на програмування має декілька недоліків:

- зосередження на програмуванні за рахунок виключення інших розділів інформатики дає студентам обмежене розуміння дисципліни;
- теоретичні питання, що повинні закріпити розуміння практичного матеріалу, відкладаються на більш пізні етапи навчання, тоді, коли вони вже не мають необхідного значення;
- курси з програмування, як правило, сфокусовані на синтаксисі та особливостях мови програмування, тим самим занадто спрощують процес програмування для того, щоб він був доступним для студентів перших курсів;
- курси інтенсивного вивчення програмування не адаптовані для студентів, які не мають досвіду роботи з комп'ютерами;
- такий підхід може призвести студентів до переконання, що створення програми є єдиним методом розв'язання проблеми з використанням комп'ютера.

Не зважаючи на вищезгадані недоліки, модель з орієнтацією на програмування довела свою виняткову життєдайність протягом останніх десятиліть.

Одним із найбільш актуальних питань у викладанні інформатики є роль та місце програмування в навчальному плані. Виникла необхідність під час навчання програмуванню використовувати методи підвищеної мотивації навчання студентів комп'ютерних спеціальностей та майбутніх викладачів інформатики для школи.

Розглядаючи різні підходи до навчання програмування майбутніх інженерів-програмістів, Л. В. Гришко пропонує наступні підходи до структуризації змісту навчальної дисципліни: імперативний, об'єктно-орієнтований, апаратно-орієнтований, візуальний, функціональний та логічний [6: 25].

Розглянемо деякі з цих стратегій.

Підхід "з орієнтацією на функціональне програмування" вперше був застосований на початку 80-х минулого століття в одному з технологічних інститутів США та характеризується використанням простої функціональної мови. Перевагами такого підходу є: використання мови, що не використовується в промисловості, зменшує ефект різниці в підготовці студентів, частина з яких завжди має певний досвід в програмуванні; мінімальний синтаксис функціональної мови дозволяє викладачу фокусуватися на фундаментальних питаннях; деякі важливі теми (підпрограми, зв'язані структури даних, рекурсія) доволі просто вписуються в такий підхід і можуть бути викладені на початку курсу.

Традиційно функціональний підхід використовується на курсах теорії програмування, штучного інтелекту та, власне, функціонального програмування. Проте багаторічний досвід впровадження функціонального підходу в процесі підготовки інженерів-програмістів дозволяє зробити обґрунтоване припущення про доцільність навчання програмування на основі зазначеного методу і майбутніх учителів інформатики. На жаль такий підхід вимагає абстрактного мислення у студентів та відповідального ставлення до вивчення мови, яку вони потім ніколи не будуть використовувати.

Підхід "з орієнтацією на об'єктно-зорієнтоване програмування" також сфокусований на програмуванні, але з самого початку ставиться акцент на принципах об'єктно-зорієнтованого проектування та програмування. Головною перевагою даного підходу є раннє ознайомлення з об'єктно-зорієнтованим програмуванням, яке, на даний час, стало актуальним як для академічного середовища, так і промисловості. Проте цей підхід має і недоліки – мови, які використовуються в об'єктно-зорієнтованому програмуванні (C++, Java), набагато складніші за мови функціонального програмування, тому викладач повинен докласти певних зусиль з обмеження складностей в теоретичному матеріалі. Підходи, що використовуються у викладанні програмування, не дозволяють в повній мірі оцінити можливості інформаційних технологій, що розвиваються швидкими темпами. Саме тому необхідно приділяти більше уваги вивченню різноманітних візуальних середовищ програмування на практиці.

Підхід "з орієнтацією на алгоритми" зорієнтований на використання псевдокоду замість реальної мови програмування. За рахунок ознайомлення з основними алгоритмічними концепціями та логічними структурами, незалежно від мови програмування, цей підхід мінімізує зусилля, які витрачаються на вивчення специфічних синтаксичних конструкцій. На противагу цьому від студентів вимагається доведення та роз'яснення алгоритмів, які вони створюють, налагоджують на аркуші паперу за допомогою своєї уяви. Це дає змогу студентам працювати з широким колом типів даних та структур керуючої логіки, без необхідності подолання специфічних особливостей, які притаманні будь-якій мові програмування. Як тільки студенти оволодіють основними алгоритмами та типами даних, вони можуть використовувати одну з поширених мов програмування. Завдяки вилученню з програми навчання часу, який відведено на вивчення синтаксису та деталей середовища програмування, курс можна доповнити додатковими теоретичними темами.

Проте підхід "з орієнтацією на алгоритми" має низку критичних моментів, з яких вагомим є те, що студенти, сповнені ентузіазму, прагнуть "змусити комп'ютер щось робити". Також орієнтація на псевдокод позбавляє студентів необхідності демонструвати те, що їх програми працюють та мають функціональну реалізацію. І останнім із вагомих недоліків такого підходу є оцінювання псевдокоду на коректність – це значно складніша задача, яка потребує залучення асистентів з числа студентів.

Незалежно від обраної стратегії навчання курс програмування повинен містити всі необхідні розділи та теми. Вибір того чи іншого варіанту залежить як від специфіки викладання окремого викладача, так і вищого навчального закладу загалом. В усіх стратегіях передбачені стандартні форми занять – лекції, практичні або лабораторні заняття та самостійна робота, що цілком відповідає навчальним планам, передбаченим державними стандартами України.

Майбутньому вчителю інформатики необхідно мати базовий рівень математичної підготовки незалежно від технічних знань з інформатики, володіти науковими методами, реалізовувати чисельні методи на практиці, працювати в колективі. Математичні методи та формальні міркування є складовими більшості галузей інформатики. Інформатика залежить від математики та її фундаментальних визначень, аксіом, теорем та методів доведення. Математика надає інструментарій для роботи над поняттями, що належать до інформатики, конкретними засобами аналізу та верифікації, а також теоретичну основу для розуміння різного типу ідей інформатики. Функціональне програмування та розв'язування задач з використанням комп'ютера базується на математичних концепціях та аналізі функцій; для аналізу алгоритмів необхідні знання з комбінаторики та теорії ймовірностей, теорії графів; верифікація алгоритмів базується на формальній логіці та дедукції. Таким чином, для розуміння теоретичних основ інформатики доцільно в програму навчання майбутнього вчителя інформатики включати достатній об'єм дисциплін математичного циклу.

Студенти повинні отримати певні знання, вміння та навички з математичного аналізу, лінійної алгебри, статистики, чисельних методів, теорії чисел, геометрії та логіки. Також студенти повинні бути ознайомлені з прийомами дискретної математики.

Використання на лабораторних заняттях сучасних середовищ розробки програмних продуктів суттєво підвищує мотивацію студентів до вивчення програмування. Практична перевірка на практиці засобів середовища програмування дає можливість для формування чіткого уявлення про сучасний процес проектування. Чітка мотивація необхідності вивчення програмування дозволяє реалізувати принцип свідомості та активності студентів.

Психолого-педагогічними особливостями студентського віку є максимальний розвиток пам'яті, уваги, сприйняття, мислення, емоцій та почуттів, мови, активного формування індивідуального стилю навчання. У пізнавальній діяльності пріоритетним стає абстрактне мислення, формується узагальнена картина світу, аналізуються зв'язки між складовими оточуючої дійсності.

Висновки. У процесі навчання у вищому навчальному закладі студент повинен набути здібностей навчатися та самостійно отримувати знання, які визначають його можливості у професійній діяльності. Багаторічний досвід показує, що ефективність навчання програмуванню зростає, коли студент самостійно працює над проектом, а не отримує вже готовий кінцевий продукт відразу. Під час роботи

над проектом студентом краще засвоюються теоретичні знання, відпрацьовуються набуті навички програмування та формуються інформаційно-комунікаційні компетентності.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ ТА ЛІТЕРАТУРИ

1. Жалдак М. І. Педагогічний потенціал комп'ютерно-орієнтованих систем навчання математики / М. І. Жалдак // Комп'ютерно-орієнтовані системи навчання : зб. наук. праць. – К. : НПУ ім. Драгоманова. – Вип.7. – 2003. – С. 13–16.
2. ACM Curriculum Committee on Computer Science. Curriculum'68 : recommendations for the undergraduate program in computer science. Communications of the ACM, 11(3) : 151197, March 1968
3. Computing Curricula'91. Association for Computing Machinery and the Computer Society of the Institute of Electrical and Electronics Engineers / [Allen B. Tucker, Brace H. Barnes, Robert M. Aiken, Keith Barker, Kim B. Brace, J. Thomas Cain, Susan E. Conry, Gerald L. Engel, Richard G. Epstein, Doris K. Lidtke, Michael C. Mulder, Jean B. Rogers, Eugene H. Spafford, and A. Joe Turner] – 1991.
4. Education Committee of the IEEE Computer Society. A curriculum in computer science and engineering. Publication EHOI 198, Computer Society of the IEEE, January 1977.
5. Организация обучения информатике [Електронний ресурс]. – Режим доступу : <http://ict.edu.ru>.
6. Буланова-Топоркова М. В. Педагогика и психология высшей школы : [учеб. пос.]. – 3-е изд., перераб. и доп. – Ростов-н/Д : Феникс. – 2006. – 512 с.
7. Зиновьев С. И. Учебный процесс в советской высшей школе / С. И. Зиновьев.– [2-е изд.]. – М. : Высшая школа, 1975. – 314 с.
8. Гришко Л. В. Методична система навчання основ програмування майбутніх інженерів-програмістів : дис. ... канд. пед. наук : 13.00.02 / Л. В. Гришко // Національний педагогічний ун-т ім. М. П. Драгоманова. – 2009. – 276 с.
9. [Електронний ресурс]. – Режим доступу : <http://se.math.spbu.ru/cc2001>.
10. [Електронний ресурс]. – Режим доступу : <http://www.computer.org/education/cc2001>.

Матеріал надішов до редакції 18.05. 2011 р.

Кривонос А. Н. Особенности преподавания программирования в высшем учебном заведении с учетом требований современности.

В статье рассмотрены основные особенности преподавания курса программирования с учетом существующих стратегий и психолого-педагогических особенностей обучения студентов. Сделан анализ опыта профессиональных сообществ ACM и IEEE Computer Society по созданию рекомендаций преподавания информатики в университетах. Описан базовый уровень математической подготовки будущего учителя информатики.

Kryvonos O. M. The Programming Teaching Peculiarities in Higher Educational Institutions in Light of the Demands of the Present Time.

The article deals with the peculiarities of the teaching programming course in light of the contemporary strategies and psychological, pedagogical trends in students' tuition. The experience of professional societies ACM and IEEE Computer Society in establishing recommendations for the course of Informatics in the universities has been analyzed. The basic level in Mathematics preparation of future teachers of Informatics has been determined.