

**Методичні підходи до викладання програмування в педагогічних ВНЗ, як один з початкових кроків до створення вільнорозповсюджуваних програмних засобів навчального призначення.**

*В даній статті зроблено аналіз методичних підходів, які можуть використовуватися при навчанні програмування в педагогічних ВНЗ. Звертається увага, що навчання програмування майбутніх вчителів інформатики, може виступати рішучим кроком по створенню та використанню вільнорозповсюджуваних програмних засобів.*

**Ключові слова:** *Вільнорозповсюджуваний програмний засіб, педагогічний програмний засіб, методи навчання.*

**Постановка проблеми.** Майже одночасно з появою в школі комп'ютерів були створені програми, призначені для навчання школярів програмування. Потім з'явилися програми для навчання іншим предметам. За такими програмами та інструментальними засобами закріпився термін „педагогічні програмні засоби” (ППЗ).

Можна виділити декілька основних проблем використання ППЗ в навчальному процесі, першою з яких є те, що програмні засоби спочатку створюються, а потім розпочинаються способи знайти їм місце в навчальному процесі, причому при створенні таких програмних засобів, розробники не завжди узгоджують свої дії з вчителями практиками. Другою проблемою виступає те, що більшість програмних засобів, що пропонуються для підтримки викладання тих чи інших предметів працюють під пропрієтарними операційними системами, що пов'язане з тотальним «майкрософтоцентризмом» кінця 90-х – початку 2000-х років. Третьою проблемою можна вказати платність таких програмних засобів. Тобто використання пропонованих педагогічних програмних засобів може бути досить дорогим задоволенням для сучасних Українських вчителів та шкіл в цілому.

Зважаючи на вищесказане постає задача в створенні власних програмних засобів, які можуть використовуватися в навчальному процесі і вільно розповсюджуватися серед Українських шкіл, оскільки вільне програмне забезпечення містить у собі чималу кількість важливих, у тому числі й стратегічних, переваг, чи не найголовнішою серед яких є можливість суттєвого заощадження бюджетних коштів. Така задача може бути покладена на вчителів інформатики, що в свою чергу потребує своєрідного навчання студентів педагогічних ВУЗів відповідних спеціальностей.

**Аналіз публікацій з проблеми дослідження.** Класифікацією методів навчання займалися Д.О. Лордкіпанідзе, Є.Я. Голант, Н.М. Верзілін, І.Я. Лернер, М.І. Махмутов, Ю.К. Бабанський, А.М. Алексюк та інші. Разом з тим окремими методичні підходи до навчання програмування зустрічаються в роботах: М.М. Скаткіна, Є.М. Кабанової-Меллер, Г.С. Костюка, А.П. Ершова.

**Метою статті** є огляд методичних підходів, які можуть використовуватися при викладанні програмування в педагогічних ВНЗ.

### **Виклад основного матеріалу дослідження.**

**Класичні методи навчання програмування.** Перш ніж розпочинати огляд методичних підходів, потрібно розглянути поняття класичних методів навчання та їх класифікацією, оскільки саме вони використовуються в будь-яких методичних підходах до навчання, в тому числі і навчання програмування.

Методи навчання (від гр. *methodos* — спосіб пізнання, шлях дослідження) — це упорядковані способи діяльності вчителя й учнів, спрямованих на засвоєння учнями системи знань, набуття умінь і навичок, їх виховання і загальний розвиток. [7].

Відомо, що на вибір методів навчання впливають такі чинники, як специфіка дисципліни, цілі навчання, зміст курсу, складність навчального матеріалу, кількість відведеного часу (кредитів) на дисципліну, рівень підготовленості студентів, рівень мотивації навчання студентів, матеріально-технічна база навчального закладу. Тому в процесі навчання основ програмування доцільно використовувати методи навчання, які обумовлюються видами навчально-пізнавальної діяльності студентів.

Під методом навчання у ВНЗ розуміють впорядковані способи взаємопов'язаної діяльності викладача і студента, спрямовані на досягнення поставлених цілей навчання конкретної наукової дисципліни.

Проблемам розробки та використання методів навчання присвячені численні дослідження. Класифікації методів навчання відрізняються один від одного ознакою, покладеною в основу кожного з них. Так, поширені класифікації методів за джерелами знань (Д.О. Лордкіпанідзе, Є.Я. Голант, Н.М. Верзілін); за характером логіки пізнання (І.Я. Лернер); за рівнем проблемності засвоєння знань (М.І. Махмутов); на основі цілісного підходу до процесу навчання (Ю.К. Бабанський); на основі цілісного підходу до навчальної діяльності (А.М. Алексюк) та інші.

Спільним для всіх підходів є те, що в кожному з них відображається три групи ознак, які характеризують: навчально-пізнавальну діяльність, педагогічну діяльність, предмет спільної діяльності учасників педагогічного процесу.

Розглянемо основні класифікації методів навчання з точки зору застосування цих методів при навчанні курсу програмування. Серед яких можна виділити три: за способом передачі відомостей від викладача до студента, за основними видами дидактичних проблем, що вирішуються на заняттях, за характером пізнавальної діяльності студентів.

І. За способом передачі відомостей від викладача до студента виділяють наступні методи: словесні або вербальні – бесіда, пояснення, розповідь, лекція, інструктаж; наочні або демонстраційні – ілюстрування та демонстрування; практичні – вправи, лабораторні та практичні роботи, дослідні роботи.

Під час навчання курсу програмування словесні методи використовуються при викладанні лекційного матеріалу. В наочних методах головну роль відіграє демонстрація та візуалізація викладачем структур, алгоритмів, що дозволяють в наочній формі динамічно відображати деталі їх роботи. Практичні методи частіше всього застосовуються при розв'язуванні задач, виконанні лабораторних та практичних робіт. При цьому наголос робиться на практичні методи, оскільки під час їх застосування студенти отримують не лише нові знання, але й практичні навички. Роль викладача полягає в інструктажі, постановці цілей роботи, контролі

за її виконанням. В діяльності студентів переважає практична робота під час якої особливу роль відіграє самостійний процес мислення, що дозволяє здійснити пошук підходів до розв'язування задач, перегляд стандартних алгоритмів та підбір потрібного з них.

II. За основними видами дидактичних проблем, що вирішуються на заняттях, можна виділити наступні методи: методи набуття знань, методи формування вмінь, методи застосування знань, методи творчої діяльності, методи перевірки компетентностей.

III. Класифікація методів навчання програмування за характером пізнавальної діяльності студентів: пояснювально-ілюстративний, або інформаційно-рецептивний метод навчання; репродуктивний метод навчання; проблемний метод навчання; частково-пошуковий або евристичний метод навчання; метод проектів або дослідницький метод навчання; метод активного навчання; метод інтерактивного навчання.

Інформаційно-рецептивний і репродуктивний методи забезпечують засвоєння студентами передбачених робочою програмою фактів, методів, алгоритмів, тощо і формування у них вмінь і навичок їх застосування, що є передумовою успішного навчання і творчої діяльності студентів, тоді як методи проблемного навчання – метод проблемного викладу, евристичний метод, дослідницький метод – формують риси творчої особистості студентів.

***Ітераційно-поступальний підхід.*** Аналіз загально-дидактичних і приватно-дидактичних методів навчання показує, що дуже часто навчальна діяльність представляє собою ітераційно-поступальний процес. На основі цього висновку можна запропонувати, ще один приватно-дидактичний метод, який може бути застосований при навчанні програмування – ітераційно-поступальний підхід навчання.

Розглядаючи ітерацію, як покрокове наближення до певної мети, можна застосувати ітераційного підходу, як при викладенні лекційного матеріалу так і в процесі виконання лабораторних робіт. Тим більше, що специфіка задач, призначених для виконання на лабораторному практикумі, повністю відповідає

ітераційно-поступальному процесу, який полягає в побудові ряду моделей, причому кожна наступна модель є уточненням або розширенням попередньої. Кожна лабораторна робота має містити такі розділи, як: короткий теоретичний матеріал (основні відомості); експерименти з програмами (в готові програми необхідно внести деякі зміни і проаналізувати отриманий результат); завдання для самостійної роботи; теоретичний матеріал для допитливих

Таким чином побудова кінцевої моделі представляє собою ітераційний процес, на кожному кроці якого відбуваються певні зміни. Власне це і дозволяє застосовувати ітераційний підхід навчання в процесі навчання програмування.

**Задачний підхід.** На сучасному етапі навчання програмування найпоширенішим являється задачний підхід. М.М. Скаткін відзначає, що “сучасна психологія і дидактика розкрили значну роль задач у навчанні як одного з найважливіших чинників збудження і підвищення пізнавальної і практичної активності учнів” [2].

У роботах Є.М. Кабанової-Меллер, Г.С. Костюка та ін., обґрунтовується концепція формування мислення, яку визначають як „задачний підхід”. Розв’язання задачі розглядається як цілеспрямована діяльність, вагоме місце також відводиться розумовій діяльності в ході якого і відбувається розвиток мислення студентів. Початковим етапом такого процесу найчастіше є проблемна ситуація, оскільки найчастіше, мислити людина починає, коли в неї виникає потреба щось зрозуміти.

Як і в більшості предметів природничо-математичного циклу в програмуванні уміння та навички розв’язування задач формуються безпосередньо у процесі розв’язування задач на основі застосування знань відповідних теоретичних основ.

Відзначимо, що в програмуванні, як і в математиці, під методом розв’язування задач розуміють сукупність прийомів розумової діяльності або логічних дій і операцій, за допомогою яких розв’язується широкий клас задач, а під способом розв’язування задач розуміють сукупність прийомів розумової діяльності або логічних дій і операцій для розв’язування окремої задачі. У програмуванні, як і в математиці, практично кожна задача має декілька способів (алгоритмів) розв’язування.

Особливістю розв'язування задач з програмування є те, що розв'язування задачі з програмування, як правило, передбачає реалізацію алгоритму деякою конкретною мовою програмування, що вимагає від студента не лише знання відповідних мовних конструкцій, а і вміння працювати в певному середовищі програмування.

Можна зазначити, що розв'язання задачі з програмування може складатися з двох частин: побудова інформаційної моделі задачі, пошук і побудова алгоритму її розв'язування та реалізацію цього алгоритму конкретною мовою програмування. Причому для задачного підходу перша з них є більш ваговою. Тобто можна зазначити, що мова програмування є інструментом, за допомогою якого розв'язується задача і який можна поміняти, якщо в цьому виникає необхідність.

Як показує досвід, у процесі навчання основ програмування можна виділити чотири стадії вміння і здатності розв'язування задач студентами: репродуктивне програмування, осмислене програмування, доказове програмування, творче програмування.

*Репродуктивне програмування.* Знання, отримані в результаті використання пояснювально-ілюстративного методу, не формують навиків і вмінь користуватися цими знаннями. Для набуття студентами навиків та вмінь і разом з тим досягнення кращого засвоєння знань, викладач системою задач організує діяльність студентів по неодноразовому відтворенню викладених ним знань і показаних способів дій. Викладач дає завдання, а студенти їх виконують – розв'язують схожі завдання, працюють за інструкцією. Відтворення та повторення розв'язування задач з програмування по завданням викладача є головною ознакою стадії репродуктивного програмування. Сама назва характеризує лише діяльність студента, проте по опису, можна побачити, що дана стадія включає організаційну діяльність викладача.

*Осмислене програмування.* На цій стадії програмування від студента вимагається більше знань ніж на стадії репродуктивного програмування, оскільки студенту необхідно самостійно визначити типи даних, необхідних для розв'язування задачі, вміння побудувати алгоритм, розуміння конструкцій мови програмування для написання тексту програми, можливість протестувати програму. Проте, на цій

стадії навчання, студент найчастіше все ще бачить лише один алгоритм розв'язування задачі, який не завжди є раціональним.

*Доказове програмування.* Ідея доказового програмування вперше була висловлена академіком А.П. Ершовим [3]. Доказове програмування – це створення програм з доказом їх правильності. Складність конструювання і доказ правильності алгоритмів і програм полягає в наступному. Для висновків про наявність помилок в алгоритмі чи програмі достатньо вказати тести, при виконанні яких відбудеться збій, відмова чи буде отримано невірний результат. Для ствердження правильності програми, необхідно довести, що правильна відповідь буде отримана для всіх допустимих вхідних даних. Такі твердження можуть бути доведені тільки шляхом вичерпного аналізу результатів виконання програми при будь-яких доступних даних.

*Творче програмування.* Творче програмування характеризується новизною свого результату для студента, своєрідністю процесу його отримання і істотним впливом на розумовий розвиток. Студенти знаходячись на рівні творчого програмування можуть забезпечувати самостійне розв'язання нових для них проблем, глибоке засвоєння знань, швидкий темп оволодіння новими знаннями, широту їх перенесення в відносно нові умови, здатність самостійно формулювати задачі, використовувати навчальну, наукову, періодичну літературу з програмування в пошуках нових знань. В творчому програмуванні проявляються інтелектуальні здібності студента, його творчий потенціал.

Такий підхід сприяє підвищенню інтересу студентів до навчання, надає їм впевненості у своїй можливості і знання, що, у свою чергу, активізує їх пізнавальну активність, стимулює навчальну діяльність студентів. В той же час розв'язання задачі за готовим алгоритмом, запропонованим викладачем, не вимагає від студента самостійного мислення. А навички вміння розв'язувати задачі відбувається тільки в результаті самостійної праці.

*Методичний підхід до парного програмування.* В процесі навчання програмування доцільно використовувати метод «парного програмування», суть якого полягає в тому, що під час виконання завдань два студента одночасно

працюють за одним комп'ютером. При цьому один із студентів управляє комп'ютером і, в основному, думає над кодуванням в деталях. Другий студент зосереджений на задачі в цілому і постійно переглядає код, що набирається першим студентом.. В цьому випадку студенти, що працюють в парі, постійно спілкуються з приводу навчального завдання, здійснюють взаємонавчання і взаємоконтроль.

Для підходу парного програмування обов'язковою є умова обміну ролями студентів в процесі роботи, тобто періодично студенти, що працюють в парі, повинні мінятися місцями за робочим місцем (комп'ютером).

***Колективний проект як методичний підхід до активного навчання програмування.*** В більшості випадків сучасне програмування є колективним, і вклад кожного окремого програміста в спільний проект впливає на успіх для всієї команди. Необхідно розуміти важливість формування і розвитку у студентів таких важливих якостей, як уміння працювати в колективі і вміння співпрацювати з іншими колективами, наприклад колективом вчителів предметників з інших дисциплін чи колективом методистів.

Учасники команди працюють над спільною задачею, шукають шляхи її розв'язання. Кожен сумлінний учасник несе особисту відповідальність за код програми і вважає доцільним знаходити і відстоювати кращі розв'язки проблеми. Так між студентами можуть виникати суперечки, а як відомо в суперечках народжується істина.

Такий вид навчальної діяльності також спонукує студентів до самостійної творчої роботи і глибокого усвідомленого вивчення курсу програмування.

Якщо розглядати початковий курс програмування, то можна також виокремити такі підходи до навчання програмування: підхід з максимальним охопленням матеріалу, підхід з орієнтацією на алгоритми і підхід з орієнтацією на апаратну частину

***Підхід з максимальним охопленням матеріалу*** полягає в розгляді більшої кількості навчального матеріалу без глибокого занурення в нього. Застосування даного підходу дає студентам більш цілісний погляд на дисципліну, що вивчається, уявлення про цілий ряд важливих тем, замість того, щоб відразу занурювати їх в

деталі однієї конкретної проблеми. Студенти, які навчаються за таким підходом, одержують більш широке уявлення про програмування, що дозволяє їм рухатися далі, впевнено і цілеспрямовано, опановуючи інші комп'ютерні курси.

До недоліків такого підходу можна віднести те, що розгляд широкого спектру теоретичних питань з програмування на ранній стадії навчання, що передбачає такий підхід, ускладнює засвоєння навчального матеріалу студентами молодших курсів.

***Підхід з орієнтацією на алгоритми*** при навчанні основних концепцій програмування передбачає використання псевдокоду замість реальної мови програмування. Застосування цього підходу мінімізує зусилля студента, що йдуть на вивчення специфічних синтаксичних конструкцій конкретної мови програмування. Натомість від студентів вимагається обґрунтування і роз'яснення алгоритмів, які вони створюють. Це дозволяє студентам працювати з широким діапазоном типів даних і базових структур. Після того, як студенти опанують основні типи алгоритмів і типи даних, вони можуть починати використовувати одну з мов програмування. Завдяки виключенню з навчальної програми, відведеного часу на вивчення синтаксису і деталей певного середовища програмування, у початковий курс програми, який побудований на основі такого підходу, можна включати додаткові теоретичні теми. В результаті, студенти починають знайомитися з відповідними аспектами теорії з перших днів навчання.

***Підхід з орієнтацією на апаратну частину*** передбачає вивчення основ інформатики і, зокрема, основ програмування, починаючи з апаратно-логічного рівня. Зазначається, що таким чином студенти можуть вивчити інформатику більш глибоко і послідовно. Тільки після сформування у студентів розуміння структурних особливостей апаратної частини, машинної логіки і математики, в курсі здійснюється перехід до розгляду програмування мовами високого рівня. З погляду розробників такий підхід більш придатний для студентів, що вважають за краще розуміти процес обчислення на ЕОМ у всіх його деталях.

Проте підхід з орієнтацією на апаратну частину не дуже добре узгоджується з сучасною тенденцією все більшого вдосконалення віртуальних машин, що

відділяють процес програмування від апаратних засобів. Цей підхід може бути використаний при підготовці фахівців з проектування комп'ютерної техніки, коли необхідно досягнути раннього ознайомлення з апаратною частиною обчислювальної системи.

**Висновок.** На практиці під час навчання програмування викладачем, найчастіше, використовується не якийсь окремий методичний підхід, а злиття декількох підходів в один для більш ефективного навчання.

#### Список використаної літератури:

1. Балл Г.А. Теория учебных задач: Психолого-педагогический аспект. Педагогический энциклопедический словарь / Г.А. Балл. – М., 2002. С. 321.
2. Дидактика средней школы / Под ред. М.А. Данилова, М.Н. Скаткина. – М., 1975. С. 162.
3. Ершов А. П. Введение в теоретическое программирование: Беседы о методе / А.П.Ершов. – М.: Наука, Главная редакция физико-математической литературы, 1977. – 288 с.
4. Пойя Д. Как решать задачу / Пойя Д. – М.: Либроком, 2010. – 208 с.
5. Гончаренко С.У. Український педагогічний словник / С.У. Гончаренко. – К.: Либідь, 1997. – 376 с.

A.O.Kostuchenko

***Methodological approaches to teaching programming in pedagogical universities, as one of the initial steps to create free software.***

*In this article the analysis of methodical approaches that can be used in teaching programming in pedagogical universities. Also pay attention that teaching programming future science teachers, can act a decisive step in the establishment and use free software.*

***Keywords:*** free software, educational software, methods of education.