

ПІДВИЩЕННЯ РІВНЯ ПРОФЕСІЙНОЇ ПІДГОТОВКИ СТУДЕНТІВ В ПРОЦЕСІ НАВЧАННЯ МАТЕМАТИКИ

В статті на основі аналізу навчальних програм з усіх математичних та комп'ютерно-орієнтованих дисциплін циклу фахової підготовки студентів спеціальності „Програмування для електронно-обчислювальної техніки і автоматизованих систем”, запропоновано шляхи підвищення ефективності навчання математики та рівня професійної підготовки майбутніх програмістів засобами реалізації міжпредметних зв'язків.

Сьогодні проблема використання НІТН в процесі математичної освіти спеціалістів різного фаху та кваліфікаційного рівня є надзвичайно популярною. Опубліковано велику кількість робіт [1, 4, 5, 6, 9, 11, 12, 13, 14] та захищено ряд дисертацій (наприклад, [8, 10, 15]), в яких досліджуються роль, місце і значення НІТН в процесі навчання математики. Всі автори обґрунтовують, що НІТН потрібно, дуже корисно і просто необхідно використовувати в навчанні математики. Але це і так не викликає сумнівів. Сьогодні говорити про використання НІТН, особливо про специфіку цього використання, при підготовці спеціалістів певного фаху, можна тільки відповівши на наступні питання:

1. В якому місці навчального курсу (при вивченні яких тем, розділів, розв'язуванні яких задач) доцільно використовувати НІТН?
2. В якому обсязі здійснювати комп'ютерну підтримку процесу навчання?
3. З якою метою використовувати НІТН (які дидактичні цілі планується досягти)?
4. Яким методичним рекомендаціям потрібно слідувати, щоб досягти поставленої мети в найповнішому обсязі?

Спробуємо розв'язати деякі з поставлених проблем, взявши за основу процес навчання математичних дисциплін студентів. Метою будь-якого нововведення в процес навчання, як правило, є підвищення рівня підготовки майбутнього фахівця, а отже, і використання НІТН повинно сприяти досягненню даної мети.

Зрозуміло, що програміст повинен вміти розробляти певний програмний продукт для розв'язування поставленої перед ним задачі будь-якої практичної спрямованості.

З огляду на це можна дещо уточнити мету використання НІТН в процесі підготовки програмістів:

використання НІТН при навчанні математики повинно сприяти підвищенню рівня інформаційної культури, тобто як

користувацького рівня студентів, так і рівня їх вмінь та навичок щодо написання програм.

Оскільки особливістю математики є високий рівень абстракції та дедуктивний метод побудови висновків, то це не дає можливості говорити про використання НІТН при вивченні всіх тем та розділів математичних курсів в однаковій мірі, необхідно проаналізувати окремо кожен розділ та тему. При цьому на кожному етапі навчання математики з використанням НІТН необхідно враховувати специфіку спеціальності та рівень навченості студентів.

При використанні НІТН в процесі навчання математики майбутніх програмістів можна реалізовувати два підходи:

1) використання існуючих математичних програмних пакетів під час лекційних та практичних занять з метою ілюстрації теоретичного матеріалу та автоматизації окремих етапів розв'язування певних математичних задач.

2) проведення лабораторних робіт, на яких перед студентами буде ставитись задача створення програм для розв'язування певних математичних задач.

Перший підхід може бути використаний в процесі навчання математики студентів різних спеціальностей (технічних, економічних та інших). Його основною метою є підвищення ефективності навчання математики, при цьому студенти виступають в основному як користувачі, яким не потрібно навіть розуміти, яким чином (за яким алгоритмом) отримано результат.

Другий підхід можна і доцільно реалізовувати при навчанні студентів-програмістів. Його метою є підвищення професійного рівня за рахунок:

1) формування алгоритмічного мислення та алгоритмічної культури;

2) формування навичок створення програм;

3) підвищення рівня засвоєння теоретичного матеріалу;

4) формування вміння застосовувати отримані на лекційних та практичних заняттях знання, вміння і навички до розв'язування практичних задач, тощо.

Навчальним планом підготовки фахівців за напрямом підготовки „Комп'ютерні науки” спеціальності „Програмування для електронно-обчислювальної техніки і автоматизованих систем”, затвердженим Міністерством освіти і науки України, передбачено вивчення таких математичних та комп'ютерно-орієнтованих дисциплін:

<i>Математичні дисципліни</i>	<i>Комп'ютерно -орієнтовані дисципліни</i>
1 курс	
Математика	Основи інформатики

2 курс	
<ul style="list-style-type: none"> • Лінійна алгебра та аналітична геометрія (I семестр) • Основи дискретної математики (II семестр) • Математичний аналіз 	<ul style="list-style-type: none"> ✘ Основи програмування та алгоритмічні мови ✘ Архітектура комп'ютерів ✘ Операційні системи ✘ Технологія розробки програмного забезпечення ✘ Текстові редактори та видавничі системи
3 курс	
<ul style="list-style-type: none"> • Диференціальні рівняння (I семестр) • Чисельні методи (I семестр) • Теорія ймовірностей та математична статистика (II семестр) 	<ul style="list-style-type: none"> ✘ Основи програмування та алгоритмічні мови ✘ Комп'ютерна схемотехніка ✘ Технологія розробки програмного забезпечення ✘ ЕОМ та мікропроцесорні системи ✘ Табличні процесори
4 курс	
<ul style="list-style-type: none"> • Математична логіка (I семестр) • Математичне програмування (II семестр) 	<ul style="list-style-type: none"> ✘ Основи програмування та алгоритмічні мови ✘ Об'єктно-орієнтоване програмування ✘ Комп'ютерні мережі ✘ Інженерна графіка ✘ Системи управління базами даних ✘ Структура даних та алгоритми проектування автоматизованих інформаційних систем

На першому курсі студенти спеціальності „Програмування для електронно-обчислювальної техніки та автоматизованих систем” вивчають курс математики, який дублює програми 10 та 11 класів загальноосвітньої школи. Оскільки більшість студентів, вступивши до коледжу, не мають базових знань з програмування, то ставити завдання написання програм неможливо. На даному етапі найдоцільніше реалізовувати перший підхід, тобто використовувати математичні програмні пакети для ілюстрації

теоретичного матеріалу, автоматизації окремих етапів розв'язування задач, візуалізації математичних об'єктів та інше.

На сьогодні розроблено значну кількість програмного забезпечення, що дозволяє розв'язувати за допомогою комп'ютера досить широке коло математичних задач різних рівнів складності. Наприклад, можна використовувати прикладні програми загального призначення, які допомагають розв'язувати математичні задачі. Прикладом такої програми можуть бути електронні таблиці Microsoft Excel, які входять до MsOffice 9x/2000/Me/XP/2003. Також зручними для підтримки навчання математики є прикладні програми спеціального призначення, розраховані на розв'язування певного кола математичних задач. До таких програм відносяться: Gran1, Gran2D, Gran3D, Derive, Eureka, MathCad, MathLab, Maple, Mathematika, Maxima, Numeri, Reduce, Startgraph тощо.

Найбільш зручними для підтримки навчання математики на першому курсі є програмні комплекси Gran1, Gran2D, Gran3D, Derive.

Ці програми прості у використанні, мають досить зручний інтерфейс, максимально наближені до інтерфейсу найбільш поширених програм загального призначення. Для використання цих програм не потрібно мати значний обсяг спеціальних знань з інформатики, основ обчислювальної техніки, програмування, досить володіти найпростішими поняттями, які є цілком доступними для будь якого студента. Використання таких програм дає можливість більше уваги приділити побудові математичних моделей, розробці і дослідженню методів розв'язування задач, дослідженню розв'язків, логічному аналізу умов задач, перекласти на комп'ютер чисто технічні та не цікаві операції, виконання яких практично не розвиває інтелекту студента [4].

Методичні рекомендації щодо навчання математики з використанням різних математичних програмних пакетів для першокурсників спеціальності „Програмування для електронно-обчислювальної техніки та автоматизованих систем” суттєво не відрізняється від подібних рекомендацій для фахівців інших спеціальностей.

На сьогодні існує багато публікацій, в яких ґрунтовно розглянуті питання щодо місця та методики використання при навчанні математики таких програмних засобів як Gran та Derive, [4-8, 10, 11, 13], тому просто дамо короткий огляд місця застосування цих програм при навчанні математики на першому курсі.

Програма Gran може бути використана, зокрема при вивченні таких тем:

- × різні способи задання функцій;
- × властивості функцій та їх графіки;
- × обернені функції;
- × дослідження функцій;
- × графічне розв'язування систем рівнянь та нерівностей;
- × побудова січних та дотичних до графіка функції;
- × обчислення та застосування визначених інтегралів;

- для обчислення площ довільних фігур;
- обчислення довжини дуги кривої;
- обчислення об'ємів та площ поверхонь тіл обертання.

Програму Derive можна використати для:

- × розкладу виразів на множники, спрощення виразів;
- × розв'язування рівнянь;
- × розв'язування системи лінійних рівнянь;
- × операції над векторами;
- × обчислення границь послідовностей і функцій;
- × обчислення скінченних і нескінченних сум та добутків;
- × відшукування первісних та похідних в аналітичному виді;
- × побудова графіків функцій в декартовій та полярній системах координат;
- × обчислення визначених інтегралів;
- × обчислення довжин дуг кривої;
- × обчислення об'ємів та площ поверхонь тіл обертання;
- × побудова зображень деяких поверхонь.

На другому курсі студенти починають вивчати „Основи програмування та алгоритмічні мови”, тому на цьому етапі можна говорити про проведення лабораторних робіт з математичних дисциплін, вивчення яких передбачено навчальним планом для 2-4 курсів спеціальності ПОМ (програмування та обчислювальні машини).

В залежності від рівня знань студентів з програмування при навчанні математичних дисциплін у першому семестрі другого курсу, можна реалізовувати:

- поєднання двох зазначених підходів, якщо навчання програмування почалося лише з початком вивчення дисципліни „Основи програмування та алгоритмічні мови”;
- введення лабораторних робіт з математики з перших занять, за умови, що на першому курсі студентам було подано основні відомості з програмування.

При цьому написання програм студентами для розв'язування математичних задач може здійснюватись не лише на лабораторних роботах з математики та при виконанні домашніх завдань, тобто мати переважно навчальний характер, а й здійснювати контролюючі функції під час написання „інтегрованих контрольних робіт”, де перевірятимуться не лише знання з інформатики (практичні вміння використання різних операторів, функцій та ін), а й практичні вміння та знання теоретичного матеріалу з даної математичної дисципліни.

Проаналізуємо тематичні плани деяких математичних дисциплін, що вивчаються студентами на 2-4-х курсах, і наведемо перелік тем та задач, використання яких при проведенні лабораторних робіт є найдоцільнішим.

Курс „Лінійна алгебра та аналітична геометрія”:

<u>Тем</u>	<i>Задачі, які можуть використовуватись на лабораторних роботах з математики</i>	<i>Навички з програмування, які можуть формуватися на лабораторних роботах з математики</i>
1	2	3
Дії з матрицями	<ul style="list-style-type: none"> ✗ Додавання та віднімання матриць ✗ Знаходження добутку матриць ✗ Транспонування матриці ✗ Обчислення визначника матриці ✗ Знаходження оберненої матриці 	<ul style="list-style-type: none"> ▪ Використання процедури введення-виведення
Розв'язування системи лінійних рівнянь	<ul style="list-style-type: none"> ✗ За методом Крамера ✗ За матричним методом 	<ul style="list-style-type: none"> ▪ Використання простих операторів (надання значення, процедури переходу)
Вектори та дії над ними	<ul style="list-style-type: none"> ✗ Обчислення довжини вектора, заданого в координатній формі ✗ Знаходження суми та різниці двох векторів, заданих в координатній формі ✗ Обчислення скалярного добутку двох векторів, заданих своїми координатами 	<ul style="list-style-type: none"> ▪ Використання структурних операторів (умовний оператор, оператор циклу з передумовою, циклу з постумовою, циклу з лічильником) ▪ Використання операцій та функцій для величин основних типів даних:

1	2	3
	<ul style="list-style-type: none"> ✘ Знаходження кута між векторами, заданими своїми координатами ✘ З'ясування колінеарності та компланарності векторів, заданих своїми координатами ✘ Знаходження векторного добутку векторів, заданих своїми координатами ✘ Обчислення площі трикутника, побудованого на заданих векторах або заданого вершинами ✘ Знаходження мішаного добутку векторів, заданих своїми координатами ✘ Обчислення об'єму паралелепіпеда (призми, піраміди), побудованого на векторах, що задані своїми координатами, як на сусідніх ребрах 	<p>для величин цілого типу:</p> <p>множення, ділення, ділення без остачі, знаходження остачі від ділення, додавання віднімання, операції відношення;</p>

1	2	3
<p>Прямокутна декартова система координат та її основні задачі</p>	<ul style="list-style-type: none"> ✳ Знаходження відстані між точками ✳ Знаходження рівняння прямої, що проходить через відому точку і має відомий вектор напрямку ✳ Знаходження рівняння прямої, що проходить через дві відомі точки, задані своїми координатами ✳ Знаходження рівняння прямої, що проходить через відому точку і має відомий вектор нормалі ✳ Знаходження рівняння прямої „у відрізках на осях” ✳ Знаходження рівняння прямої з кутовим коефіцієнтом, що проходить через дві точки, задані своїми координатами, або проходить через відому точку і має відомий вектор напрямку ✳ З’ясування взаємного розміщення двох прямих ✳ Обчислення кута між прямими ✳ Обчислення відстані від точки до прямої ✳ Дослідження та побудова кривих другого порядку 	<p style="text-align: center;"><i>для величин дійсного типу:</i> операції множення, ділення, додавання, віднімання, знаходження абсолютного значення величини, квадрату величини, кореня квадратного, піднесення величини до будь-якого дійсного степеня, обчислення логарифма довольною основою, знаходження значень тригонометричних та обернених тригонометричних функцій;</p> <p style="text-align: center;"><i>для величин булівського типу:</i> операції відношення, логічне множення, логічне додавання, виключаюче „або”);</p>

1	2	3
Аналітична геометрія у просторі	<ul style="list-style-type: none"> ✘ Рівняння площини, що проходить через відому точку та перпендикулярна до заданого вектора ✘ Рівняння площини, що проходить через відому точку і паралельна двом не колінеарним векторам ✘ Рівняння площини, що проходить через три точки, які не належать одній прямій ✘ З'ясування взаємного розміщення двох площин ✘ Знаходження кута між площинами ✘ Обчислення відстані від точки до площини ✘ Задачі, що розглядались для прямої на площині ✘ Обчислення відстані між мимобіжними прямими ✘ Дослідження поверхонь другого порядку 	<ul style="list-style-type: none"> ▪ Робота з даними статичної структури (зокрема, робота з масивом) ▪ Використання функцій та процедур стандартного модуля Graph (ініціалізація графічного режиму, „поточний” вказівник, побудова та задання параметрів для найпростіших графічних фігур, виведення тексту в графічному режимі)

Саме цей курс містить задачі, під час розв'язування яких студенти можуть довести свої програмістські вміння (застосовувати базові алгоритми та типи даних), до автоматизму, тобто сформувати навички написання програм з використанням певних базових знань з програмування.

Під час написання цих програм, студенти мають змогу на практиці переконатись, що при конструюванні алгоритмів поділ кожної вказівки на простіші здійснюється за допомогою тільки трьох базових алгоритмічних структур: слідування, розгалуження, повторення.

Розглянемо тепер курс математичного аналізу. Він має певну специфіку, яка не дозволяє широко залучити студентів до написання програм. Коло задач, які можуть бути програмно реалізовані студентами, обмежене, оскільки рівень знань студентів з програмування не відповідає рівню складності програмної реалізації цих задач.

А саме:

<u>Теми</u>	<i>Задачі, які можуть використовуватись на лабораторних роботах з математичного аналізу</i>
Похідна	<ul style="list-style-type: none"> ▪ Застосування диференціалу до наближених обчислень
Визначений інтеграл	<ul style="list-style-type: none"> ▪ Обчислення визначеного інтегралу за „методом прямокутників” ▪ Обчислення визначеного інтегралу за „методом трапеції”
Застосування визначеного інтегралу	<ul style="list-style-type: none"> ▪ Наближене обчислення площі криволінійної трапеції ▪ Наближене обчислення довжини дуги кривої ▪ Наближене обчислення об’єму тіла обертання
Ряди	<ul style="list-style-type: none"> ▪ Застосування ряду Маклорена до наближених обчислень ▪ Наближене обчислення суми ряду

Свою специфіку має використання НІТН в навчанні чисельних методів. При вивченні цього курсу студентам для написання лабораторних робіт можна запропонувати наступні задачі:

<u>Теми</u>	<i>Задачі, які можуть використовуватись на лабораторних роботах з дисципліни „Чисельні методи”</i>
Розв’язування рівнянь	<ul style="list-style-type: none"> ▪ Метод ділення навпіл ▪ Метод хорд ▪ Метод дотичних ▪ Метод простих ітерацій
Чисельне розв’язування систем лінійних рівнянь	<ul style="list-style-type: none"> ▪ Метод ітерацій ▪ Метод Зейделя
Чисельне диференціювання	<ul style="list-style-type: none"> ▪ Знаходження значення похідної функції в точці
Чисельне інтегрування	<ul style="list-style-type: none"> ▪ Наближене обчислення визначеного інтегралу за „методом прямокутників” ▪ Наближене обчислення визначеного інтегралу за „методом трапецій” ▪ Наближене обчислення визначеного інтегралу за методом Сімпсона
Розв’язування диференціальних рівнянь	<ul style="list-style-type: none"> ▪ Метод простих ітерацій (Ейлера) ▪ Метод Адамса ▪ Метод Рунге-Кутта
Побудова інтерполяційних многочленів	<ul style="list-style-type: none"> ▪ Многочлен Лагранжа ▪ Многочлен Ньютона

Зауважимо, що при програмному розв’язуванні задач з чисельних методів студенти повинні не тільки знайти розв’язок задачі, а й оцінити точність його наближення та збіжність методу, що застосовується.

В курсі теорії ймовірностей та математичної статистики можна запропонувати винести на лабораторні роботи наступні задачі:

Теми	<i>Задачі, які можуть використовуватись на лабораторних роботах з дисципліни „теорія ймовірностей та математична статистика”</i>
Елементи математичної статистики	<ul style="list-style-type: none"> ▪ Обчислення числових характеристик вибірки (математичне сподівання, дисперсія, середнє квадратичне відхилення, мода, медіана) ▪ Знаходження емпіричної функції розподілу ймовірностей
Дискретні випадкові величини (одно- та двовимірні)	<ul style="list-style-type: none"> ▪ Обчислення числових характеристик вибірки (математичне сподівання, дисперсія, середнє квадратичне відхилення, мода, медіана) ▪ Знаходження функції розподілу ймовірностей, побудова її графіка ▪ Знаходження ймовірності того, що випадкова величина набудатиме значення з заданого проміжку
Кореляційний аналіз	<ul style="list-style-type: none"> ▪ Обчислення коефіцієнтів кореляції та детермінації ▪ Оцінювання щільності зв'язку між двома показниками
Регресійний аналіз	<ul style="list-style-type: none"> ▪ Обчислення параметрів лінійної регресійної моделі ▪ Перевірка адекватності моделі ▪ Побудова довірчого інтервалу для прогнозованого значення результату ознаки Y
Перевірка статистичних гіпотез	<ul style="list-style-type: none"> ▪ Перевірка гіпотези про тип розподілу генеральної сукупності; ▪ Перевірка гіпотези про рівність середніх значень, дисперсій та ін.

На основі здійсненого аналізу можна сказати, що навчальні програми математичних та комп'ютерно-орієнтованих курсів дозволяють реалізовувати обидва запропоновані підходи до навчання математики, які дозволять:

- підвищити рівень мотивації навчання математики за рахунок розв'язування професійно та практично орієнтованих задач;
- активізувати пізнавальну діяльність студентів через реалізацію міжпредметних зв'язків між математикою та професійно-орієнтованими дисциплінами;
- підвищити рівень професійної культури майбутніх програмістів;
- покращити організацію та підвищити ефективність самостійної роботи студентів;

- урізноманітнити методи навчання і форми участі студентів у навчальній діяльності;
- ефективніше організувати оперативний контроль і самоконтроль результатів навчальної діяльності студентів.

ЛІТЕРАТУРА

1. Гершунский Б.С. Компьютеризация в сфере образования: Проблемы и перспективы. — М.: Педагогика, 1987. — 264с.
2. Гнеденко Б.В. Математика и математическое образование в современном мире. — М.: Просвещение, 1985.— 192 с.
3. Давидов В.В. Проблемы развивающего обучения. — М.: Педагогика, 1986.- 240с.
4. Жалдак М.І. Комп'ютер на уроках математики: Посібник для вчителів.— К.: Техніка, 1997.— 303 с.
5. Жалдак М.І., Вітюк О.В. Комп'ютер на уроках геометрії: Посібник для вчителів. — К.: РННЦ "ДІНІТ", 2000. — 168с.
6. Жалдак М.І., Пеньков А.В. Нова інформаційна технологія на уроках математики // Радянська школа, 1991, №1. — С.77-80.
7. Жильцов О.Б., Торбін Г.М. Вища математика з елементами інформаційних технологій: Навч. посіб. - К.:МАУП, 2002. — 408с.
8. Клочко В.І. Нові інформаційні технології навчання математики в технічній вищій школі: Дис...д-ра пед.наук (13.00.02) — Вінниця, 1998.- 396с.
9. Машбиц Е.И. Психолого-педагогические проблемы компьютеризации обучения: (Педагогическая наука — реформе школы).— М.: Педагогика, 1988.— 192 с.
10. Морзе Н.В. Система методичної підготовки майбутніх вчителів інформатики в педагогічних університетах: Дис...д-ра пед.наук (13.00.02) — Київ, 2003.- 600с.
11. Новые педагогические и информационные технологии в системе образования / Под ред. Е.С. Полат. — М.: «Академия», 2000. — 272с.
12. Раков С.А. Математична діяльність та її підтримка засобами інформаційних технологій // Комп'ютерно-орієнтовані системи навчання: Зб. наук. праць. — К.: НПУ імені М.П.Драгоманова. — Випуск 3. — 2001. — 287с.
13. Раков С.А., Горох В.П., Олійник Т.О., Гармашова Н.М., Якуба М.О. Інформаційні технології в аналітичній геометрії: Навч. посібник для студентів математичних спеціальностей університетів.— Харків: ХДПУ, 2000.— 189 с.
14. Співаковський О.В. Теорія і практика використання інформаційних технологій у процесі підготовки студентів математичних спеціальностей: монографія. - Херсон: Айлант, 2003.
15. Співаковський О.В. Теоретико-методичні основи навчання вищої математики майбутніх вчителів математики з використанням інформаційних технологій: Дис...д-ра пед.наук (13.00.02) — Київ, 2003.- 535с.