

середовище. Якщо тема передбачає більше зосередження на практичних уміннях, то доцільним вважаємо додати до курсу короткі відеоуроки чи покрокові інструкції роботи з платформою, що вивчається. Після завершення створення теоретичного блоку варто розробити практичний. Згідно календарно-тематичного планування у вчителя може бути запланована одна або більше обов'язкова практична робота. Інструкції до цих робіт, файли-шаблони, тощо також доречно розмістити в окремому блоці. На наступному етапі вчителю варто переглянути виконану роботу та визначити, які додаткові ресурси до даної теми необхідно прикріпити на своєму курсі. Для цього доцільно створити окремий блок або вкладку. Далі, якщо дана система буде основним засобом комунікації, варто додати блок із термінами виконання робіт та тестуваннями, для визначення рівня знань. І останнім етапом, звісно, буде апробація даної платформи учнями, та виправлення недоліків, якщо такі виникнуть.

Отже, використання смартфонів як навчальних інструментів відкриває нові можливості для сучасного викладання інформатики. М-learning сприяє адаптації процесу навчання до потреб учнів, розвиваючи навички роботи в мобільному середовищі та позбавляючи залежності від стаціонарних комп'ютерів. Проте, успішна інтеграція мобільних платформ в освітній процес потребує продуманого підходу до вибору платформ і ресурсів, що дозволяє створити ефективний, доступний і зручний курс для учнів. Дотримуючись цих рекомендацій, вчителі можуть створити курси, які сприятимуть розвитку ключових навичок учнів і забезпечать високий рівень залученості в процес навчання.

### **Список використаних джерел**

1. Олексюк В. П. Єдина система автентифікації як крок до створення освітнього простору загальноосвітнього навчального закладу. Науковий часопис Національного педагогічного університету імені М. П. Драгоманова. Серія 2: Комп'ютерно-орієнтовані системи навчання, 2012. № 13(20). С. 187–192.
2. Рашевська Н. В., Ткачук В. В. Технології мобільного навчання. *Педагогіка вищої та середньої школи?* 2012. № 1(35). С. 295–301.

## **ПРО ПРИКЛАДНИЙ ТА АКАДЕМІЧНИЙ РІВНІ АБСТРАКЦІЇ У ПІДГОТОВЦІ ФАХІВЦІВ ІТ-ГАЛУЗІ**

### **Стельмащук Людмила Володимирівна**

кандидат фізико-математичних наук, викладач циклової комісії програмування,  
Відокремлений структурний підрозділ «Гусятинський фаховий коледж Тернопільського  
національного технічного університету імені Івана Пулюя»,  
stelyudmilav@gmail.com

### **Стельмащук Володимир Миколайович**

інженер-розробник програмного забезпечення,  
ТОВ «КОМ'ЮНІТІ ФІЛЬМ ПРОДАКШН»,  
jordan29.04.1997@gmail.com

Дозволимо собі висловити не дуже популярну думку, що вища освіта – не завжди є правильним інструментом для навчання майбутнього спеціаліста у галузі

інтернет технологій. Основна задача ВНЗ – дати якісну вищу освіту, і це правильно. Проте, здебільшого, вища освіта це забагато для освоєння професії програміста.

Не просто так в Україні є кілька рівнів освіти [1]: I (початковий, базовий та вищих) рівень – професійна (професійно-технічна) освіта, яка здобувається у технікумах та училищах, інших прирівняних до них закладах та відповідає 2–4 рівню Національної рамки кваліфікацій; II рівень – фахова передвища освіта, що забезпечується коледжами, чи прирівняними до них закладах освіти і відповідає 5 рівню Національної рамки кваліфікацій; III рівень – вища освіта (молодший бакалавр, бакалавр, магістр, доктор філософії та доктор наук), яка здобувається в університетах, інститутах, консерваторія, академіях, та відповідає 5–8 рівню Національної рамки кваліфікацій.

Це нашоє думку, що не для всіх професій потрібно навчатися в закладі одного з вищих рівнів акредитації. Вища освіта – це, в першу чергу, про глибину знань та їх всеосяжність.

Проаналізувавши навчальні програми вищих навчальних закладів, можна зробити висновок що навчання студентів в таких закладах має бути надзвичайно складним, і по закінченню 4–6 курсів вони мають стати надзвичайно якісними спеціалістами з великою шириною та і глибиною знань в своїй сфері. Тут важливо зрозуміти що саме у сфері, а не в спеціальності. Реальність така, що людей, які можуть дійсно якісно вчитися та вивчити програму на такому рівні, надзвичайно мало. Це звучить як погана новина, але це не так. Індустрія не потребує багато людей з такою кваліфікацією.

Це можна порівняти з тим, що кількість людей, які вміють водити автомобіль менша ніж кількість людей, які знають скільки передач в їх авто, а ще менше людей знають, як влаштована коробка передач. Звісно, всі професійні гонцики знають все про коробку передач в своєму гоночному боліді, але цей факт не говорить про те, що всі водії мають знати як працює їх коробка передач.

Для успішного розвитку ІТ навчання в Україні, нам потрібно визначитися з рівнем абстракції для навчальних програм із підготовки програмістів, для того щоб це було максимально швидко та ефективно.

Розглянемо два рівні абстракції, один прикладний та максимально наближений до індустрії та вивчення конкретних технологій. Другий можна назвати академічними, він включає вивчення багатьох деталей різних аспектів комп'ютерних наук.

Для ефективного навчання людей на кожному з цих рівнів, потрібно розділити підходи до їх навчання. Для академічного рівня варто залишити все як є в теперішній системі освіти. Єдине, що потрібно змінити, - це кількість людей, які готує академічний рівень. А саме надзвичайну малу кількість інженерів, які будуть кращими з кращих. Ці люди, за рахунок того що матимуть глибокі та широкі знання, як експерти будуть вирішувати глобальні проблеми індустрії.

Прикладом такої роботи в індустрії інтернет технології була розробка протоколу ipv6, це дійсно проблема світового масштабу, і кращі системні адміністратори займалися її вирішенням. Наприклад, люди такого рівня знань

можуть будуть працювати над розробкою ідей у нових галузях, таких як квантові обчислення тощо.

У прикладному рівні стоять інші завдання. Потрібно навчити багато програмістів, які будуть щодня якісно виконувати прикладні задачі – створення сайтів, сервісів та клієнтських застосунків для різних операційних систем. На нашу думку їх потрібно навчати в навчальних закладах фахової передвищої освіти.

У навчальних програмах потрібно приділити увагу вивченню мов програмування, рушіїв та бібліотек на рівні користування цими інструментами. Проблема у тому, що зараз у навчальних програмах часто неправильний вибір рівня абстракції, на якому подається матеріал. Наприклад, навчальні програми передбачають пояснення деталей реалізації тих чи інших методів з системних бібліотек мови програмування. Чи правильний тут вибір абстракції? З одного боку, ці знання не потрібні для користування методами бібліотеками; з іншого боку це дає оманливе враження студентам, що вони вивчають матеріал на низькому рівні абстракції, що, звісно ж, не так.

Розглянемо простий приклад виклику методу `print` («Hello») в довільній мові програмування. Для того, щоб зрозуміти як насправді працює цей метод, потрібно не просто пояснити, що стрічка «Hello» буде передана в операційну систему, яка виведе її в консоль. Насправді, це все працює набагато складніше. Операційна система повинна перетворити стрічку в набір пікселів різного кольору; а для того щоб розрахувати координати для розміщення цього набору пікселів на екрані операційній системі потрібно знати роздільну здатність монітора. Передача зображення між операційною системою та монітором, та від монітора до операційної системи – це складна задача яка реалізується протоколами передачі відео, такими як HDMI, VGA, DVI, DP та інші.

Все вище описане вже виглядає складніше, а ми навіть не розглядали випадків роботи комп'ютера з кількома моніторами, або з додатковою відеокартою тощо. Отже, відображення стрічки на екрані – це складна інженерна задача з багатьма рівняннями абстракції, і те що студентам пояснюється один з них, не додає студентам розуміння того, як системи працюють насправді. Це, навпаки, шкодить, оскільки дає ілюзію студентам, що вони знають щось на низькому рівні, при цьому не розуміючи як працює система в цілому.

Правильніше, на нашу думку, пояснювати як працює система в цілому, які загальні компоненти системи є, дати схеми, що ілюструють високорівневі компоненти системи. При цьому потрібно опустити складні деталі реалізації кожної з систем.

Курси навчальної програми повинні складатися з вивчення різних прикладних технологій без вивчення деталей реалізації, з можливістю обрати студентом курси, які будуть пояснювати деталі реалізації конкретної технології. Логічно було б поділити курси на основні та додаткові.

Основні курси повинні вивчати актуальні технології на прикладному рівні. Навчальний план повинен охопити вивчення технологій для створення додатків для різних сфер, серверні додатки, браузерні додатки, додаток для мобільних

платформ та курс з розробки програм для мікрокомп'ютерів. Навчання має бути побудоване так, щоб кожний наступний курс дозволив студентам використати знання з попереднього курсу.

Наприклад, можна почати з вивчення прикладних web-технологій, таких як HTML, JS, CSS, де студенти створять сайт без серверної взаємодії, потім вивчити створення серверних додатків, далі – бази даних. Після цього логічно дати курс з інфраструктури, на прикладі одного з хмарних провайдерів.

При цьому конкретний план навчання студентів у закладах фахової передвищої освіти на 2,5 – 3 роки, дозволить підготувати спеціаліста в сфері інформаційних технологій на прикладному рівні.

### **Список використаних джерел**

1. Про затвердження Національної рамки кваліфікацій: офіц. текст. Київ: КМ, 2011. URL: <https://zakon.rada.gov.ua/laws/show/1341-2011-%D0%BF#n12> (дата звернення: 01.11.2024).

## **ІНТЕГРАТИВНИЙ ПІДХІД У НАВЧАННІ**

### **Турчин Ростислав Богданович**

здобувач третього (освітньо-наукового) рівня вищої освіти спеціальності 015 Професійна освіта, Тернопільський національний педагогічний університет імені Володимира Гнатюка  
tyrchunr@gmail.com

Інтегративний навчальний процес є позитивним викликом, який передбачає підготовку студентів до середовища, яке постійно змінюється. Він спрямований на розвиток логічних і міждисциплінарних навичок розв'язування проблем, включаючи постановку запитання, відкриття запиту, проведення експерименту та повідомлення висновку. Завдяки спільному підходу до сталого розвитку в політиці управління та розбудові спроможності, викладанні та навчанні за програмою, партнерстві та стосунках із громадою, а також у діяльності, студенти стають більш обізнаними про спільні соціально-культурні та наукові інтереси [2].

Нове мислення щодо інтегративних шляхів навчання та інтегрованих компетенцій наголошує на цілісному підході до освіти, який сприяє розвитку міждисциплінарних навичок і контекстуального розуміння. Ця зміна парадигми має вирішальне значення для підготовки учнів до складних викликів реального світу.

Інтегративний навчальний процес покращує компетенції для сталого розвитку через структуровану діяльність, включаючи дослідження та експерименти, які сприяють розвитку логічних і міждисциплінарних навичок вирішення проблем.

Концепція «навчання буттям» об'єднує наукове та дизайнерське мислення, просуваючи модель, засновану на компетенціях, яка відрізняє людські можливості від штучного інтелекту [1]. Компетентність розглядається як контекстуальна здатність, що об'єднує знання, навички та ставлення, що відповідає сучасним освітнім потребам [3].