

**Експериментальне дослідження ефективності
формування у студентів педагогічних університетів
компетентності в програмуванні на основі функціонального підходу**

Ринкова економіка висуває нові вимоги до випускників ВНЗ. Необхідність підготовки фахівців, які не лише мають ґрунтовні знання з певної галузі, але й володіють здатністю самостійної постановки та вирішення завдань, вміють проявляти ініціативу та навчатись протягом всього життя, зумовила впровадження компетентнісного підходу в освіту. Це вимагає розробки та перевірки ефективності нових методичних систем навчання, адже зміна будь-якої складової методичної системи навчання (в даному випадку – цільової) впливає й на всі інші складові.

Формування професійних компетентностей майбутнього вчителя інформатики розглядається у працях вітчизняних і зарубіжних дослідників: М. С. Голованя, Т. В. Добудько, М. І. Жалдака, В. М. Жукової, А. Г. Кіріллової, Т. П. Кобильника, О. В. Кучай, Г. В. Монастирної, В. М. Монахова, М. П. Лапчика, І. О. Новік, К. П. Осадчої, А. І. Павловського, Ю. С. Рамського, С. О. Семерікова, Є. М. Смирнової-Трибульської, О. М. Спіріна, М. В. Рафальської, А. А. Харківської, В. С. Шелудька, С. М. Яшанов та інших дослідників.

Серед інформатичних компетентностей важливе місце належить компетентностям у програмуванні, адже вона відображає зв'язок між компетентностями з математичної інформатики та інформаційних технологій.

В процесі формування компетентностей в програмуванні можуть бути застосовані різні підходи, найбільш поширені з яких – імперативний та об'єктний – опановуються як послідовно, так і незалежно один від одного. Проте фундаменталізація навчання програмування вимагає посилення ролі математичної інформатики, тому важливим є встановлення зв'язків математичної інформатики з програмуванням, математики з інформатикою з самого початку навчання програмування, що, в свою чергу, вимагає нового підходу до формування компетентностей в програмуванні. У розв'язуванні проблем формування компетентностей в програмуванні студентів педагогічних університетів в умовах фундаменталізації навчання інформатичних дисциплін може суттєво допомогти функціональний підхід до програмування.

Реалізацією функціонального підходу в програмуванні є функціональні мови програмування. Вивченню функціональних мов програмування присвячено праці Х. Абельсона, С. В. Головльової, І. Г. Косової, С. М. Малярчука, С. О. Нігіяна, Д. Д. Сассмана, Дж. Сассман, Р. У. Себести, І. О. Теплицького, А. Філда, П. Хендерсона, Е. Хьюьонена та інших дослідників.

Незважаючи на широку поширеність функціонального підходу до навчання початків програмування у зарубіжних країнах, у країнах СНД переважають імперативний та об'єктний підходи до формування компетентностей у програмуванні. При цьому навчання початків програмування у середній школі із застосуванням діяльнісних середовищ Logo, Scratch, Squeak, Alice та інших відбувається без урахування особливостей реалізації функціонального підходу в мовах програмування.

Мови та середовища функціонального програмування ряд вітчизняних дослідників пропонують використовувати виключно для ілюстрації основних положень теорії програмування та інтелектуальних систем, що відповідає історичним застосуванням мов ML та Lisp. При цьому в якості аргументів того, що мови функціонального програмування не можуть бути використані для формування компетентностей у програмуванні, наводяться переважно технічні обмеження середовищ функціонального програмування, характерні для програмних засобів 20-30-річної давнини (інтерпретована природа мов, обмеження розміру стеку для рекурсивних програм, повільне виконання, неможливість роботи із зовнішніми бібліотеками тощо).

Разом з тим невідповідності

– між педагогічним потенціалом діяльнісних середовищ функціонального програмування та невикористанням функціонального підходу в процесі їх застосування;

– між сучасною тенденцією до формування компетентностей в програмуванні на мультипарадигмній основі та наявним станом навчання початків програмування виключно імперативними засобами зумовлюють необхідність ретельного вивчення функціонального підходу та відповідних програмних засобів, придатних для застосування в процесі навчання початків програмування студентів перших курсів педагогічних ВНЗ. В результаті анкетування студентів І курсу спеціальностей «Математика та інформатика», «Фізика та інформатика», «Інформатика» Криворізького державного педагогічного університету (КДПУ), було виявлено наступне:

- систематично навчалися програмуванню (оцінки 3-5) не більше 50% вступників;
- навчання програмування, за одиничними виключеннями, відбувалося у середовищах імперативного програмування;
- в процесі навчання програмування методи активного навчання та засоби організації спільної роботи не використовувались;
- навчання програмування розглядалося без відповідних математичних основ;
- переважно низький рівень пізнавальної активності в процесі навчання програмування був обумовлений високим порогом входження у предмет (значним обсягом знань та вмій, необхідних для написання особистісно значущої програми).

Виявлені проблеми надали можливість сформулювати вимоги до засобів навчання програмування на основі функціонального підходу: стандартність та локалізованість інтерфейсу користувача; наявність можливості створення імен програмних об'єктів рідною мовою; мале синтаксичне ядро мови програмування; можливість створення коротких програм з розвинутою функціональністю; можливість модифікації програмного коду в процесі його виконання; підтримка багатьох підходів до програмування в межах одного середовища.

Огляд існуючих засобів навчання програмування показав, що жоден з них не відповідає сформульованим вимогам, проте найбільш близьким до їх реалізації було середовище програмування DrScheme. Головними недоліками його були відсутність можливості створення імен програмних об'єктів рідною мовою та нелокалізованість інтерфейсу користувача. Додатковим обмеженням на його застосування була мала кількість джерел українською та російською мовами, присвячених функціональній мові Scheme.

Для усунення виявлених недоліків була виконана локалізація середовищ PLT Scheme та DrScheme українською та російською мовами. За результатами дослідної експлуатації авторської локалізації DrScheme 4.1 у 5 комп'ютерних класах КДПУ було прийняте рішення про її включення до версій PLT Scheme та DrScheme (починаючи з червня 2010 р. – Racket та DrRacket), що розповсюджуються через офіційний сайт проекту (адреса – <http://www.racket-lang.org>).

Створена вітчизняна версія середовища DrRacket задовольняла всім поставленим вимогам до засобу навчання програмування на основі функціонального підходу та може розглядатися як середовище моделювання, а реалізація мови Scheme у ньому – як комп'ютерна інтерпретація математичних моделей функціонального підходу (λ -числення та комбінаторної логіки), що надало можливість використати DrRacket надалі як фундаментальне навчальне середовище, мобільне відносно апаратної та програмної складової обчислювальної системи.

Водночас безпосереднє застосування даного середовища на мобільних пристроях є недоцільним через орієнтацію інтерфейсу користувача на екрани з великими роздільними характеристиками. Це зумовило необхідність розроблення простого мобільного середовища, орієнтованого на комп'ютерні пристрої з середніми та низькими роздільними характеристиками екрану. На основі інтерпретатора мови Scheme, реалізованого на Java (автор – Скотт Міллер), було створено 2 аплету з інтерфейсом українською та російською мовами.

В результаті розробки та локалізації програмного забезпечення було створено умови для організації навчання на основі інтеграції технологій аудиторного, дистанційного та мобільного навчання. Запровадження методів активного навчання вимагало залучення програмних засобів організації спільної роботи у мобільному навчальному середовищі – Документи Google та Skype.

Реалізація процесу формування у студентів педагогічних університетів системи компетентностей в програмуванні на основі функціонального підходу вимагає особливої уваги до змісту навчання основ програмування. Експериментальна робота за методикою, розробленою на факультеті електромеханіки та комп'ютерних наук Массачусетського технологічного інституту (США) та узагальненою в матеріалах навчального курсу «Структура та інтерпретація комп'ютерних програм», показала наступні її особливості:

- 1) в підручнику з курсу та методичних матеріалах для викладача передбачається велика кількість лекцій-семінарів (близько 80 год.), в ході яких основні теоретичні поняття моделюються засобами мови Scheme;
- 2) лабораторні роботи з курсу являють собою, починаючи з першої ж роботи, навчальні проекти, що вимагає попереднього ознайомлення студентів з даною навчальною технологією;
- 3) курс спрямований на формування компетентностей в програмуванні майбутніх інженерів-електриків, фахівців з інформатики.

В навчальних програмах дисципліни «Шкільний курс інформатики» та курсу за вибором «Вступ до програмування» передбачається 18 та 27 год. лекцій відповідно, що не дозволяє організувати лекції-семінари. Крім того, недостатня впровадженість проектних технологій у навчальний процес середньої школи не дозволяє їх застосувати на початку роботи за курсом. I,

нарешті, системи компетентностей майбутніх інженерів та майбутніх викладачів суттєво відрізняються.

Таким чином, виникла необхідність не в модифікації існуючої методичної системи навчання програмування студентів I курсу через зміну її технологічної складової, а в розробленні нової методичної системи, змістову частину якої було узагальнено в навчальному посібнику «Схематичне програмування (початки програмування: функціональний підхід)» [1].

Впровадження розробленої методичної системи та експериментальна перевірка її ефективності були виконані на третьому етапі дослідження. Протягом 2007-2008, 2008-2009, 2009-2010 та 2010-2011 н.р. за розробленою методикою навчалися студенти фізико-математичного, природничого, технолого-педагогічного факультетів КДПУ, факультету інформаційних технологій Криворізького технічного університету, Криворізького металургійного факультету Національної металургійної академії України та Криворізького відокремленого підрозділу Запорізького інституту економіки та інформаційних технологій. Враховуючи спрямованість методичної системи на формування компетентностей в програмуванні студентів педагогічних університетів, результати її впровадження у інших ВНЗ враховувались, проте статистично не опрацьовувались.

Для перевірки ефективності розробленої методичної системи було виконано порівняння розподілів студентів за рівнями сформованості компетентностей в програмуванні за традиційної методикою (на основі імперативного підходу) та розробленою (на основі функціонального підходу).

Кожна складова системи компетентностей в програмуванні оцінювалась за трибальною шкалою (від 0 до 2), що відповідало недостатній, достатній та високій сформованості відповідної складової. Показники сформованості складової діагностувались методами педагогічного спостереження, в процесі контролю знань, захисту лабораторних робіт. Вага кожної складової у загальній сформованості системи компетентностей в програмуванні визначалась за методом експертних оцінок (за результатами опитування викладачів інформатики, які брали участь у VIII міжнародній науково-практичній конференції «Теорія та методика навчання фундаментальних дисциплін у вищій школі» (Кривий Ріг, 2010). Таким чином, числове значення рівня сформованості

системи компетентностей в програмуванні визначалось за формулою $B = \frac{1}{\min_{i=1,4} p_i} \sum_{i=1}^4 S_i p_i$. Множник

$\frac{1}{\min_{i=1,4} p_i}$ перед сумою (в розглядуваному випадку – 10) обирається для того, щоб отримане в результаті

числове значення рівня сформованості системи компетентностей було цілим числом. Таким чином, у відповідності до експертних оцінок ваги всіх складових системи компетентностей в програмуванні, значення B може бути в межах від 0 до 20.

Результати педагогічного експерименту в контрольній (КГ) та експериментальній групах (ЕГ) наведено у таблиці 1. Гістограму порівняльного розподілу студентів (у відсотках, оскільки маємо різну кількість студентів для КГ та ЕГ) за рівнем сформованості системи компетентностей в програмуванні подано на рис. 1.

Таблиця 1.

Кількість балів	Шкала оцінювання	% студентів	
		КГ	ЕГ
		%	%
0-5	низький	19,048	13,043
6-10	достатній	38,095	17,391
11-15	середній	33,333	36,957
16-20	високий	9,524	32,609

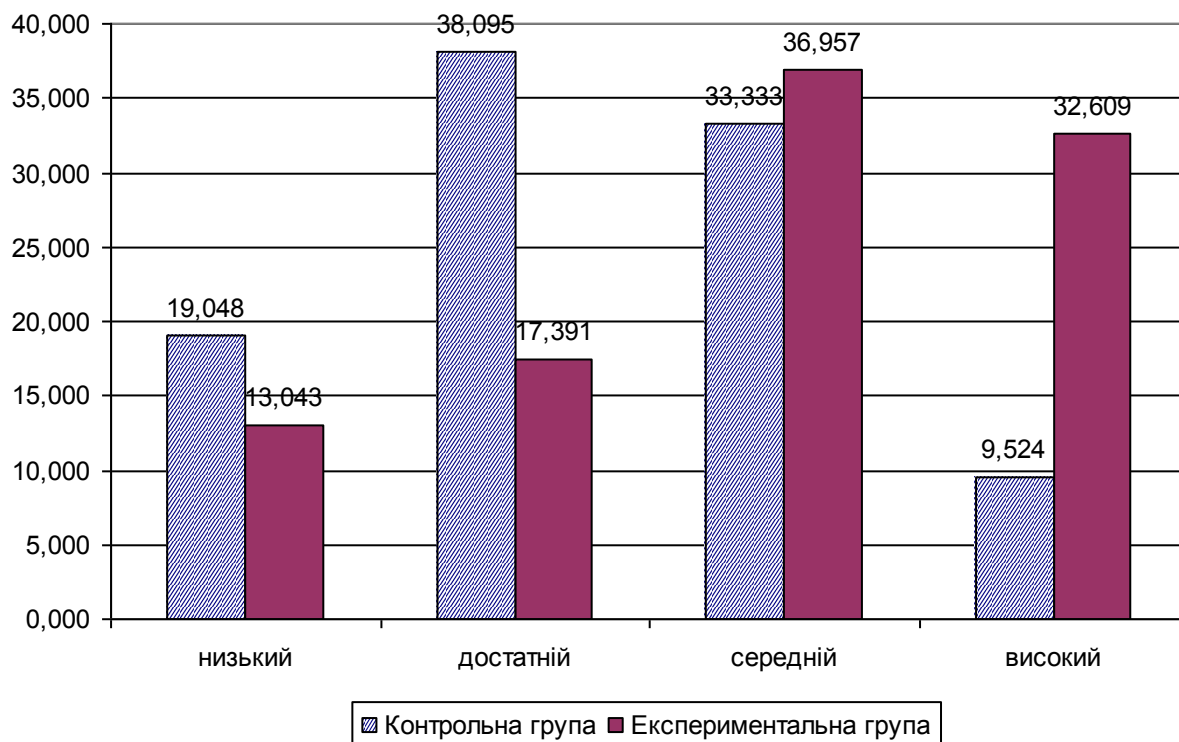


Рис. 1.

Опрацювання результатів експерименту та оцінка ефективності розробленої методичної системи здійснювалась за методами математичної статистики [2]. Оскільки задача полягає у виявленні відмінностей в розподілі певної ознаки (рівня сформованості системи компетентностей в програмуванні) при порівнянні двох емпіричних розподілів згідно [2, 34] можна скористатись χ^2 -критерієм Пірсона, λ -критерієм Колмогорова-Смирнова або ж ϕ -критерієм (кутовим перетворенням Фішера).

χ^2 -критерій Пірсона

Нульова гіпотеза H_0 : ймовірність попадання результатів роботи студентів контрольної та експериментальної вибірки в кожен з i ($i=0, 1, \dots, 20$) категорій однакова, тобто $H_0 : p_{1i} = p_{2i}$ ($i=0, 1, \dots, 20$).

Альтернативна гіпотеза $H_1: p_{1i} \neq p_{2i}$ хоча б для однієї із категорій.

Обчислення значення критерію χ^2 для експериментальної та контрольної вибірки після проведення формувального експерименту показало, що $T > T_{крит}$ ($37,038 > 30,144$). Це є основою для відхилення нульової гіпотези. Прийняття альтернативної гіпотези дає підстави стверджувати, що ці вибірки мають статистично значущі відмінності, тобто *експериментальна методична система більш ефективна, ніж традиційна*.

Для більшої переконливості виконаємо перевірку отриманих під час формувального експерименту вибірок за λ -критерієм Колмогорова-Смирнова. Позначимо:

$F(x)$ – невідома функція розподілу ймовірностей рівня сформованості системи компетентностей у програмуванні студентів в КГ;

$G(x)$ – невідома функція розподілу ймовірностей рівня сформованості системи компетентностей у програмуванні студентів в ЕГ.

Нульова гіпотеза $H_0 : F(x)=G(x)$. Альтернативна гіпотеза $H_1 : F(x) \neq G(x)$.

Коли гіпотеза H_0 справджується, відхилення $D = \sup_x |G(x) - F(x)|$ мале, а коли гіпотеза H_0 не справджується, це відхилення велике.

На рис. 2. подано графічну інтерпретацію розподілів $F(x)$ та $G(x)$.

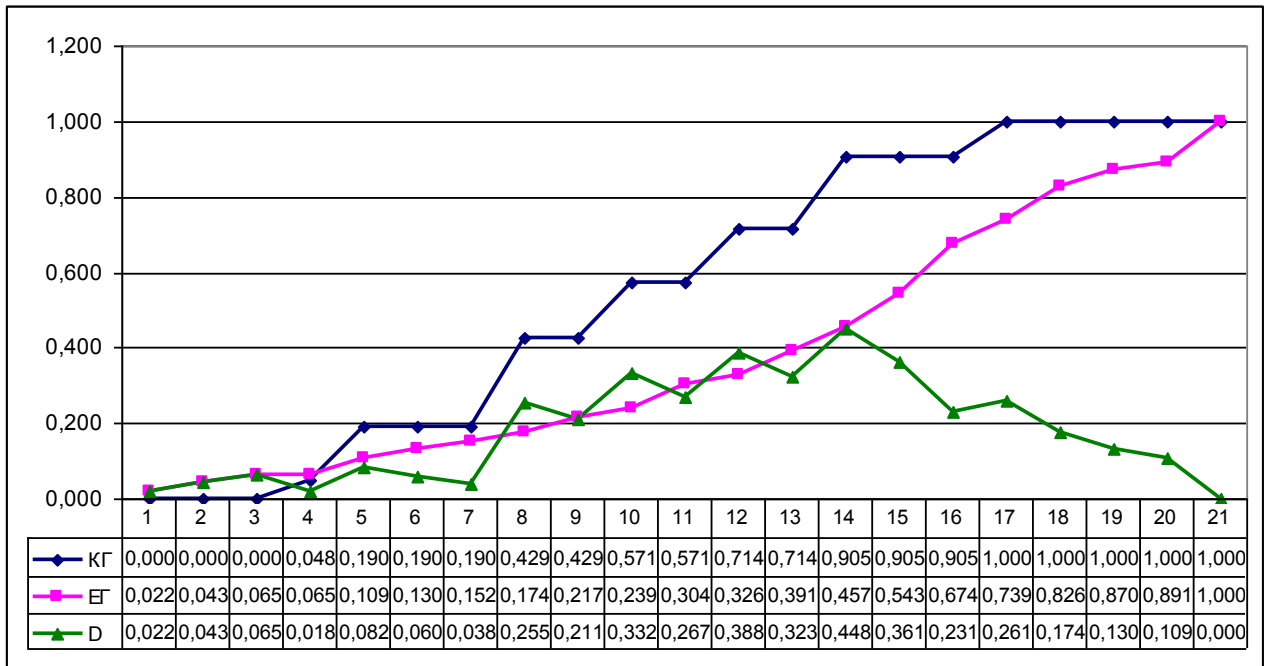


Рис. 2.

У відповідності з λ -критерієм Колмогорова-Смирнова нульова гіпотеза H_0 відхиляється і приймається альтернативна гіпотеза $H_1 : F(x) \neq G(x)$. Це означає, що існує відмінність розподілу рівня сформованості системи компетентностей в програмуванні студентів, які навчалися за традиційною методичною системою і експериментальною. Таким чином, студенти, які навчалися в EG, мали більш високий рівень сформованості системи компетентностей в програмуванні.

Враховуючи, що в EG формування системи компетентностей в програмуванні здійснювалось за розробленою методичною системою, можна припустити, що саме це і дало можливість досягти кращих результатів. Отже, можна говорити про експериментальне підтвердження висунутої гіпотези.

На основі аналізу результатів дослідження можна зробити висновок, що впровадження функціонального підходу впливає на методичну систему навчання програмування на всіх її рівнях: на рівні *цілей* навчання – з'являється мета навчання програмування як комп'ютерної інтерпретації λ - і комбінаторної алгебр та фундаментальних основ теоретичної інформатики; на рівні *змісту* навчання – створюються умови для пропедевтики навчання об'єктно-орієнтованого, подієорієнтованого, візуального і мережного програмування та інтелектуальних систем; на рівні *методів* навчання – з'являється можливість ширше застосовувати методи активного навчання; на рівні *засобів* навчання – виникає можливість застосування мобільних програмних середовищ (DrRacket та Scheme-аплет), та засобів організації спільної роботи (Skype та Документи Google); на рівні *форм організації* навчання – впровадження таких прогресивних форм навчання, як проектна, та поява нових форм, специфічних для мобільного навчання.

Призначенням розробленої методичної системи є формування у студентів педагогічних університетів системи компетентностей у програмуванні на основі функціонального підходу. Для досягнення цієї мети було створено навчально-методичний комплекс з курсу «Вступ до програмування», що включає в себе авторський посібник, відеоуроки, середовища програмування DrRacket та Scheme-аплет, а також інструкції щодо роботи з курсом в цілому та його окремими модулями. У наведених у посібнику [1] проектах ілюструються внутріпредметні та міжпредметні зв'язки різних інформатичних дисциплін, на основі чого створюються умови для опанування різних підходів до програмування в межах єдиного середовища.

Педагогічний експеримент підтвердив гіпотезу дослідження. Аналіз його результатів свідчить про підвищення рівня сформованості системи компетентностей в програмуванні при використанні розробленої методичної системи формування у студентів педагогічних університетів системи компетентностей в програмуванні на основі функціонального підходу, а, отже, і про її ефективність.

Література

1. Мінтій І. С. Схематичне програмування (початки програмування: функціональний підхід) / І. С. Мінтій; за ред. академіка НАПН України М. І. Жалдака. – К. : НПУ імені М. П. Драгоманова, 2010. – 152 с.

2. Сидоренко Е. В. Методы математической обработки в психологии / Е. В. Сидоренко. – СПб. : Речь, 2003. – 350 с.