

*institutions were observed in non-conformity of learning to the requirements of the society, the practical needs of the industrial region, etc. Question is investigated in the context of historical and pedagogical situation in Ukraine in the second half of XIX - early XX centuries.*

**Keywords:** *Mariupol classical secondary school for man, Mariupol classical secondary school for woman, Feoktist Khartakhay, charter in 1871, national education, reports on the state of Mariupol school.*

УДК 378.016 : 004.42

І. С. Мінтій, І. В. Тарасов, С. О. Семеріков

### МЕТА НАВЧАННЯ ТА ЗМІСТ КУРСУ «ВСТУП ДО ПРОГРАМУВАННЯ» ДЛЯ МАЙБУТНІХ УЧИТЕЛІВ ІНФОРМАТИКИ

*У статті розглянуто складові компетентностей з програмування, описано зміст кожної зі складових та виокремлено рівні сформованості компетентностей з програмування. Розглянуто принципи відбору змісту навчання основ програмування та визначено зміст курсу «Вступ до програмування», метою якого є формування у майбутніх учителів інформатики компетентностей з програмування на основі функціонального підходу.*

**Ключові слова:** *компетентнісний підхід, методична система навчання, мета навчання, зміст навчання, компетентності з програмування.*

**Постановка проблеми.** Методична система навчання – сукупність взаємопов'язаних компонентів: цілей навчання, змісту, методів, засобів і форм організації навчання, що утворюють єдину цілісну функціональну структуру, орієнтовану на досягнення цілей навчання [8, с. 247]. Упровадження компетентнісного підходу у вищу освіту обумовлює необхідність перегляду всіх складових компетентнісно-орієнтованих методичних систем навчання студентів.

**Аналіз останніх досліджень і публікацій.** Питанню розробки та впровадження компетентнісно-орієнтованих методичних систем навчання присвячено праці Т. Г. Крамаренко, Т. В. Колчук, Т. П. Кобильника, С. А. Ракова, М. В. Рафальської, Л. Д. Шевчук та ін. У дослідженнях Л. В. Гришко, С. О. Семерікова, І. О. Теплицького, Ю. В. Триуса, та ін. розглядаються принципи відбору змісту навчання програмування для студентів інформатичних напрямів.

У попередніх публікаціях [3] розглянуто засоби для навчання майбутніх учителів інформатики основ програмування на основі функціонального підходу.

**Метою даної статті** є визначення мети навчання програмування майбутніх учителів інформатики та змісту курсу з основ програмування на основі компетентнісного підходу.

**Виклад основного матеріалу.** *Мета (ціль) навчання* – ідеальне передбачення кінцевих результатів навчання; те, до чого прагнуть студенти, викладачі [8, с. 229].

У зв'язку з швидкими темпами оновлення засобів сучасних ІКТ, зменшенням періоду застарівання фактичного матеріалу, вчитель інформатики повинен бути готовий до самонавчання та саморозвитку впродовж всього життя. Тому, одним з необхідних сьогодні результатів здобуття освіти у педагогічному університеті є набуття майбутніми вчителями інформатики інформатичних компетентностей на основі здобутих знань з фундаментальних та прикладних розділів інформатики [7, с. 49].

Таким чином, *метою навчання програмування майбутніх учителів інформатики* за компетентнісного підходу є формування у студентів компетентностей з програмування.

Компетентності з програмування мають такі взаємопов'язані складові [5, с. 352]:

– *когнітивно-змістову* (гносеологічну) – знання основних форм для керування виконанням програми; знання простих типів даних та функцій для роботи з ними; знання похідних типів даних, способів їх утворення з простих типів даних, функцій для роботи з ними та пріоритетних напрямів їх використання; знання основних етапів розв'язування прикладних задач; знання основних етапів проектування програм; знання складових мови програмування;

– *операційно-технологічну* (праксеологічну) – вміння пояснити призначення та функції існуючої програми, знайти помилки в логіці розв'язання задачі, описати етапи розробки програм, розробити функції та обґрунтувати пріоритетність використання того чи іншого виразу для їх створення, створити документацію до програми, пояснити та продемонструвати процес створення похідних типів даних, спроектувати, описати, перевірити та проаналізувати результати виконання програми; оцінити переваги різних способів розв'язання однієї задачі; вміння обирати засоби для розв'язання задачі та обґрунтовувати свій вибір; уміння використовувати можливості обраних засобів (довідка, налагодження програми, налаштування необхідних параметрів та ін.);

– *ціннісно-мотиваційну* (аксіологічну) – емоційно-ціннісне ставлення до процесу розробки, опису, налагодження, тестування програм та аналізу результатів їх роботи; уміння знаходити нові, нестандартні рішення задачі; внутрішня мотивація до опанування програмування; готовність до активного застосування гносеологічних та праксеологічних складових у практичній діяльності; прагнення до самовдосконалення, потреба у саморозвитку гносеологічних та праксеологічних складових; уміння самостійно приймати рішення, критично ставитись до чужих впливів, здатності за власним почином організовувати діяльність, ставити мету, в разі необхідності вносити в поведінку зміни; вміння постійно і тривало домагатися мети; наполегливість у досягненні мети, прагнення до поліпшення отриманих результатів, незадоволеність досягнутим, намагання домогтися успіху; внутрішня потреба у створенні програмних продуктів;

– *соціально-поведінкову* – здатність до співпраці у процесі розробки, опису, налагодження, тестування програм та аналізу результатів їх роботи, використання засобів для організації спільної роботи над проектом; відповідальність за власну поведінку, за виконання завдань; комунікабельність; здатність до адаптації; схильність до дискусії.

Кожна складова має різні «вагові внески» в загальний рівень сформованості компетентностей з програмування [5, с. 353].

Для оцінки рівня сформованості у студентів компетентностей з програмування виокремлено низький, середній, достатній та високий рівні [4, с. 85].

*Низький рівень* характеризується негативним ставленням до процесу розробки, опису, налагодження, тестування програм та аналізу результатів їх роботи; поверхневими, несистемними знаннями з програмування, відсутністю вмінь. *Середній рівень* відзначається індиферентним ставленням до процесу розробки, опису, налагодження, тестування програм та аналізу результатів їх роботи; слабкою мотивацією до опанування програмування; середніми за об'ємом, фрагментарними знаннями, наявністю окремих, розрізнених вмінь. *Достатній рівень* передбачає виявлення інтересу до процесу розробки, опису, налагодження, тестування програм та аналізу результатів їх роботи; упорядкованими, структурованими знаннями, достатніми вміннями; проявленням здатності до співпраці у процесі програмування, використанням засобів для організації спільної роботи над проектом; здатністю до самонавчання. *Високий рівень* характеризується позитивним ставленням до процесу розробки, опису, налагодження, тестування програм та аналізу результатів їх роботи;

стійкими, ґрунтовними знаннями, творчим підходом, уміннями до нестандартного розв'язання завдань, умінням відстоювати власну думку, постійною здатністю до співпраці у процесі програмування, використанням засобів для організації спільної роботи; здатністю до самонавчання.

Особливістю формування компетентностей з програмування є їх взаємозв'язок з іншими, як спеціальними професійними (компетентностями з математичної інформатики, з інформаційних технологій та з фундаментальних природничо-математичних дисциплін), так і з загальнопрофесійними (ІКТ-компетентностями) та ключовими (навчальними, соціальними, ІКТ та підприємницькими).

Формування компетентностей з програмування має акумулюючий характер: нові знання, що здобувають студенти, доповнюють їхню систему знань; уміння, навички, способи діяльності, яких набувають студенти, інтегруються з уже сформованими [7, с. 62].

Формування компетентностей з програмування у майбутніх учителів інформатики в педагогічному університеті розпочинається в процесі навчання курсу «Шкільний курс інформатики», «Основи інформатики та обчислювальної техніки» або ж розробленому курсі «Вступ до програмування».

Зміст навчання у широкому розумінні – структура, зміст і обсяг навчальної інформації, засвоєння якої забезпечує особі можливість здобуття вищої освіти і певної кваліфікації [2].

Зміст навчання у ВНЗ'кому розумінні (на рівні навчального предмету) – система знань з певної наукової галузі, практичних вмій і навичок та способів діяльності, якими повинен оволодіти студент у процесі навчання [с. 7, 230].

Зміст навчання з програмування визначається основними державними документами про освіту: Законами України «Про освіту», «Про вищу освіту» [2], «Про національну програму інформатизації», Державною національною програмою «Освіта. Україна ХХІ століття», специфікою освітньої системи, освітнього закладу, навчальними планами та програмами.

Зміст освіти є одним із факторів економічного і соціального прогресу суспільства і повинен бути орієнтований на забезпечення самовизначення особистості, створення умов для її самореалізації; розвитку суспільства; посилення та вдосконалення правової держави.

Визначення змісту курсу з основ програмування необхідно здійснювати з врахуванням принципів, спільних як для будь-якого навчального курсу, так і властивих для курсів з програмування:

1. *Принцип відповідності навчальним цілям.* Ціллю навчання основ програмування є формування компетентностей студента, причому не лише з програмування, а й інших спеціальних професійних, загальнопрофесійних та ключових компетентностей.

2. *Принцип науковості і посиленої складності.* Вимога науковості передбачає взаємозв'язок теорії, розробки, аналізу і оцінювання ефективності, реалізації і застосування алгоритмів. Зміст курсу повинен складатися з тих розділів і тем, які важливі для практики програмування незалежно від обраного підходу до навчання програмування. В той же час зміст курсу мусить містити всі необхідні компоненти для його засвоєння, тобто не містити «білих плям», а всі завдання, що пропонуються студентам мають або бути посильними, або ж орієнтованими на зону найближчого розвитку.

3. *Принцип фундаментальності.* За твердженням Мері Шоу навчання інформатики має організовуватись «навколо ідей, а не навколо артефактів.... Машинобудівні інститути не викладають проектування бойлера – вони викладають



термодинаміку». Оскільки компетентнісний підхід є одним із напрямів фундаменталізації, принцип фундаментальності покладено в основу навчання.

З іншого боку, оскільки «основна ідея концепції фундаментального навчання програмування – виокремлення та поєднання у змісті навчання кожної теми відповідних математичних теорій, абстракцій (класичних алгоритмів і структур даних) та їх реалізацій на обраній мові програмування» [9], реалізація принципу фундаментальності досягається за рахунок фундаментальної математичної основи функціональних мов програмування ( $\lambda$ -числення та числення комбінаторів).

4. *Принцип відкритості.* Цей принцип передбачає можливість корекції змісту курсу залежно від освітнього напрямку підготовки, без порушення цілісності фундаментального ядра дисципліни. У розробленому курсі можливість реалізації даного принципу досягається через добір тематики проектів.

5. *Принцип сучасності.* Швидкий розвиток ІКТ вимагає постійного оновлення навчальної програми, що для учителів інформатики є особливо актуальним з огляду на особливості їхньої майбутньої професійної діяльності в умовах широкого впровадження засобів сучасних ІКТ у процес навчання школи.

6. *Принцип перспективності.* Цей принцип передбачає формування у студентів готовності до подальшого навчання протягом всього життя, що надасть можливість їм бути здатними розв'язувати професійні проблеми у майбутньому. У зв'язку із впровадженням компетентнісного підходу в навчанні реалізація цього принципу є безперечною, оскільки навчальні компетентності визначені одними з ключових як вітчизняними, так і європейськими вченими.

7. *Принцип вирівнювання знань.* Зміст курсу з основ програмування повинен включати модуль, під час вивчення якого буде здійснено початкове опанування мови програмування. Як вже зазначалось, особливістю функціональних мов програмування є їх нескладний синтаксис та семантика, тому даний принцип у розробленому курсі реалізується за незначний час: у модулі «Основи синтаксису» під час першої теми «Основи Scheme» студенти отримують відомості про основні складові мови програмування, вчать записувати вирази у Scheme та визначати змінні і функції.

8. *Принцип послідовності і систематичності навчання.* Кожна наступна складова курсу повинна спиратися на попередній матеріал. На рис. 1 наведено послідовність тем курсу «Вступ до програмування». Для реалізації принципу систематичності обов'язковою є формулювання теми та мети на аудиторних заняттях, демонстрація можливостей і способів досягнення мети заняття на конкретних прикладах і задачах.

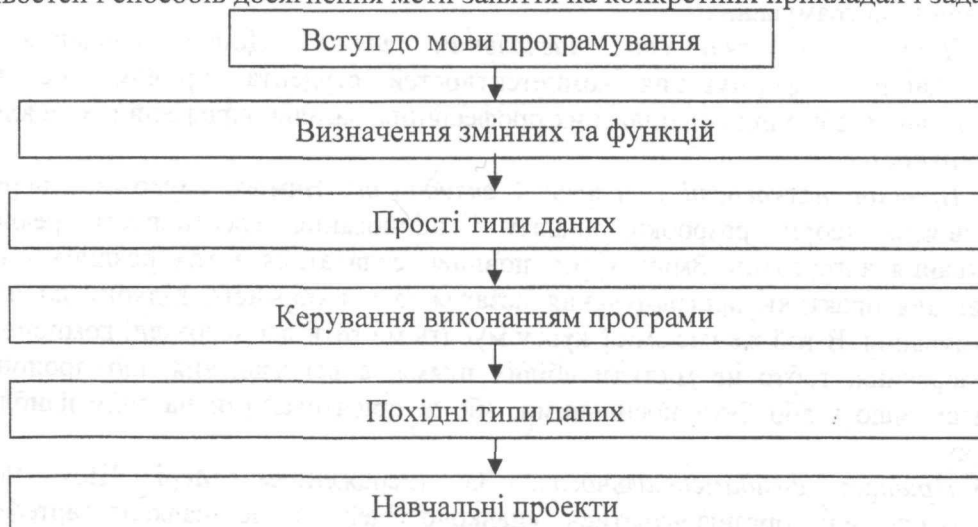


Рис. 1. Послідовність тем курсу «Вступ до програмування».

10. *Принцип наочності змісту і діяльності.* Цей принцип для програмування (а особливо для функціональних мов програмування) реалізувати досить просто (порівняно з іншими навчальними дисциплінами та іншими підходами до програмування): адже обов'язковим етапом розв'язування будь-якої задачі є реалізація алгоритму за допомогою комп'ютера. Оскільки при функціональному підході до програмування функцій можливе ще на початкових етапах вивчення курсу, то практично будь-який фрагмент коду можна визначити як функцію та переконатись у його дієвості. Особливо актуально це при програмуванні графічного інтерфейсу програми: для візуалізації деякими графічними елементами досить кількох рядків коду.

11. *Принцип створення відповідних умов для функціонування навчання.* Особливістю курсів з інформатики є їх переважне проведення у комп'ютерних аудиторіях, що зумовлює підвищені вимоги до навколишнього середовища – необхідно дотримуватись відповідних санітарно-гігієнічних вимог. Але, окрім цього, важливе значення має і створення сприятливої морально-психологічної атмосфери, що цілком залежить від рівня методичних, психолого-педагогічних, організаційно-управлінських та комунікативних компетентностей учителя.

12. *Принцип забезпечення оперативного контролю і самоконтролю в навчанні.* Для виявлення результативності та можливої корекції подальшого навчання необхідно вчасно діагностувати рівень сформованості компетентностей з програмування. Важливе значення має різноманітність форм контролю: в розробленому курсі це і усне опитування на лабораторних заняттях, і проходження тестів в електронній системі підтримки навчання Moodle, і виконання та захист лабораторних та індивідуальних завдань, модульний контроль. Для попередження виникнення стресових станів, що можуть виникнути на занятті під час тестування, студенти мають можливість самостійно пройти попереднє тестування в електронній системі підтримки навчання Moodle.

13. *Принцип індивідуалізації і колективності навчання.* Цей принцип передбачає необхідність врахування індивідуальних особливостей кожного студента: як фізичних (наприклад, студенту з вадами зору доцільним буде при роботі в парі доручити опис алгоритму розв'язування задачі, а реалізацію алгоритму покласти на іншого студента, або ж надавати матеріал у роздрукованому вигляді та ін.), так і психологічних (особливості сприйняття, уваги, мислення).

Принцип колективності у навчанні досягається шляхом використання групових форм організації роботи, однією з яких є і проектна форма. Завдяки реалізації цього принципу відбувається формування соціально-поведінкової та ціннісно-мотиваційної складових компетентностей з програмування.

14. *Принцип свідомості, активності і самостійності студентів.* Для реалізації принципу свідомості необхідно сприяти розумінню студентами сфери застосувань здобутих знань та вмінь, вмінню пояснити переваги того чи іншого методу при розв'язуванні задачі, вмінню обирати засоби для вирішення поставлених задач та обґрунтовувати вибір, тобто сприяти формуванню праксеологічної та аксіологічної складових компетентностей з програмування.

Реалізація принципу активності та самостійності студентів відбувається переважно в процесі формування елементів ціннісно-мотиваційної складової компетентностей з програмування: такі властивості студентів, як наполегливість, критичність мислення, здатність до цілепокладання, внутрішня мотивація до опанування програмуванням, прагнення до самовдосконалення, самостійність та ін.

Окрім розглянутого, добір змісту навчального матеріалу має здійснюватися з врахуванням і інших дидактичних принципів навчання: розвиваючого і виховного

характеру навчання, врахування вікових та індивідуальних особливостей, міцності і системності знань, зв'язку змісту навчання з життям, практикою та ін.

Згідно Computing Curricula 2001 [1, с. 65] будь-якого курсу з інформатики має відповідати таким вимогам:

- застосування методик навчання, що підкреслюють різницю між викладанням і навчанням (самоосвітою) та стимулюють студентів незалежно мислити;
- навчання студентів на творчих задачах та вправах, що стимулюють їх ініціативність;
- постійне оновлення обладнання та забезпечення;
- ознайомлення студентів з інформаційними ресурсами та стратегіями оновлення своїх знань;
- заохочення до навчання та використання ІКТ для забезпечення взаємодії учнів;
- переконання студентів у необхідності продовження професійного розвитку та самовдосконалення протягом всього життя.

Як бачимо, ці вимоги сприяють формуванню у студентів ціннісно-мотиваційної та соціально-поведінкової складових компетентності.

Зміст курсу, спрямованого на формування у студентів педагогічних університетів компетентностей з програмування – «Вступ до програмування», відображений у авторському навчальному посібнику «Схематичне програмування (початки програмування: функціональний підхід)» [6]. Посібник складається з двох частин: *основи програмування мовою Scheme*: опис мови, запис виразів у Scheme, визначення змінних та функцій, прості типи даних: числовий тип (number), логічний тип (boolean), тип знак (character), символний тип (symbol); керування виконанням програми: умовні вирази, циклічні вирази, рекурсивні функції, виконання блоку дій; похідні даних: рядок (string), пара (pair), список (list), вектор (array); введення/виведення, робота з файлами та *практика програмування мовою Scheme* (проекти із застосування моделей та методів математичної інформатики): «Дилема ув'язненого», «Психотерапевт», «Мінімальна система комп'ютерної алгебри», «Експертна система», «Розробка графічного інтерфейсу проекту».

**Висновки.** Отже, метою навчання програмування є формування компетентностей з програмування, до складу яких входять гносеологічний, праксеологічний, аксіологічний та соціально-поведінковий компоненти. Використання функціонального підходу для формування компетентностей з програмування надає можливість врахувати всі основні принципи відбору змісту курсу з основ програмування: відповідності навчальним цілям, науковості і посильної складності, фундаментальності, відкритості, сучасності, перспективності, принципу вирівнювання знань, послідовності та систематичності навчання, наочності змісту і діяльності та ін.

#### Список використаної літератури

1. Computing Curricula 2001 : Computer science – Final Report (December 15, 2001) / [Carl Chang, Peter J. Denning, James H. Cross II et al]. – 2001. – 240 p.
2. Закон України «Про вищу освіту» : за станом на 17.01.2002 р. № 2984-III // Відомості Верховної Ради України від 17.05.2002. – 2002, № 20, стаття 134.
3. Мінтій І. С. Засоби формування у студентів педагогічних університетів компетентності з програмування на основі функціонального підходу / І. С. Мінтій // Вісник Черкаського університету. Серія педагогічні науки. – Випуск 191. Частина І. – Черкаси : Вид. від. ЧНУ ім. Б. Хмельницького, 2010. – С. 86–92.
4. Мінтій І. С. Рівні сформованості компетентності у програмуванні / І. С. Мінтій // Теорія та методика навчання математики, фізики, інформатики : збірник наукових праць. – Випуск X : в 3-х томах. – Кривий Ріг : Видавничий відділ НМетАУ, 2012. – Т. 3 : Теорія та методика навчання інформатики. – С. 82–86.
5. Мінтій І. С. Спеціальні професійні компетентності вчителя інформатики / І. С. Мінтій // Інноваційні інформаційно-комунікаційні технології навчання математики, фізики, інформатики у середніх та вищих навчальних закладах : зб. наук. праць за матеріалами Всеукр. наук.-метод. конф. молодих науковців, 17-



18 лют. 2011 р. – Кривий Ріг : Криворізький держ. пед. ун-т, 2011. – С. 351–354.

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

7. Рафальська М. В. Формування інформатичних компетентностей майбутніх вчителів інформатики у процесі навчання методів обчислень : дис. ... канд. пед. наук : 13.00.02 – теорія і методика навчання (інформатика) / Рафальська Марина Володимирівна ; Національний педагогічний ун-т ім. М. П. Драгоманова. – К., 2010. – 225 с.

8. Триус Ю. В. Комп'ютерно-орієнтовані методичні системи навчання математичних дисциплін у вищих навчальних закладах : дис. ... доктора пед. наук : 13.00.02 – теорія і методика навчання інформатики / Триус Юрій Васильович ; Черкаський нац. ун-т ім. Б. Хмельницького. – Черкаси, 2005. – 649 с.

9. Швецький М. В. Методическая система фундаментальной подготовки будущих учителей информатики в педагогическом ВУЗе в условиях двухступенчатого образования : автореф. дисс. на соискание ученой степени доктора пед. наук : 13.00.02 – теория и методика обучения и воспитания (информатика) / Швецький Михаил Владимирович. – СПб., 1994. – 36 с.

Одержано редакцією 22.08.2013 р.  
Прийнято до публікації 28.08.2013 р.

**Аннотація.** *Мінтій І. С., Тарасов І. В., Семериков С. А. Цель обучения и содержание курса «Введение в программирование» для будущих учителей информатики. В статье определена цель обучения программированию на основе компетентностного подхода, рассмотрены составляющие компетентностей в программировании, описано содержание каждой составляющей и выделены уровни сформированности компетентностей в программировании. Рассмотрены принципы отбора содержания обучения основам программирования и определено содержание курса «Введение в программирование», целью которого является формирование у будущих учителей информатики компетентностей в программировании на основе функционального подхода.*

**Ключевые слова:** *компетентностный подход, методическая система обучения, цель обучения, содержание обучения, компетентности в программировании.*

**Summary.** *Mintiy I. S., Tarasov I. V., Semerikov S. A. The purpose of the teaching and the contents of the course «Introduction in programming» for future computer science teachers. In the article the purpose of the education as a formation of the competences in the programming is defined, the contents of all constituents of these competences, such as gnosiological, praxeological, axiological and social-behavioural are observed. Low, middle and high levels of the competences in programming are defined, standards of their development are described. The main principles of the selection of the topics for the teaching of the programming basis (correspondence to the purposes of the education, scientific nature and feasible difficult, fundamentality, openness, modernity, availability, knowledge aligning, consistency and systematic character, visualization of the contents and the activity) are examined from the point of view of the competence approach to the education and functional approach to the programming. The contents of the subject «Introduction to the programming» with the purpose of the formation of the competences in programming based on the functional approach in future computer science teachers is defined. The description is based on the programming language Scheme.*

**Keywords:** *competence-based approach, methodical system of education, purpose of the education, contents of education, competences in programming.*