

УДК 378.14:372

DOI:

Ірина Бардус, кандидат педагогічних наук, доцент, докторант
кафедри креативної педагогіки та інтелектуальної власності
Української інженерно-педагогічної академії

МЕТОДИКА ФУНДАМЕНТАЛІЗОВАНОГО НАВЧАННЯ ОБ'ЄКТНО- ОРІЄНТОВАНОМУ ПРОГРАМУВАННЮ МАЙБУТНІХ ФАХІВЦІВ У ГАЛУЗІ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Стаття присвячена проблемі підвищення якості навчання об'єктно-орієнтованому програмуванню майбутніх фахівців у галузі інформаційних технологій на засадах фундаменталізації освіти. Розроблено систему навчальних, розвивальних та виховних цілей навчання об'єктно-орієнтованому програмуванню за умови фундаменталізації освіти. Визначено умови фундаменталізації змісту навчання об'єктно-орієнтованому програмуванню майбутніх ІТ-фахівців. Описано методику організації продуктивної навчально-пізнавальної діяльності студентів під час лекційних та практичних занять.

Ключові слова: методика навчання; фундаменталізація; ІТ-фахівець; об'єктно-орієнтоване програмування; цілі навчання; зміст навчання; методи навчання; дидактичні засоби; продуктивна діяльність.

Рис. 2. Табл. 1. Літ. 8.

Iryna Bardus, Ph.D. (Pedagogy), Associate Professor,
Doctoral Student of the Creative Pedagogic and Intellectual Property Department
Ukrainian Engineering and Pedagogical Academy

METHOD OF FUNDAMENTALIZED TEACHING OF OBJECT-ORIENTED PROGRAMMING FOR FUTURE SPECIALISTS IN THE FIELD OF INFORMATIONAL TECHNOLOGIES

The article is devoted to the problem of improving the quality of teaching of the object-oriented programming for future specialists in the field of information technologies based on fundamentalization of education. The conditions of fundamentalization of the object-oriented programming's learning, a content for the future IT specialists are determined. The first condition of fundamentalization is that the content of learning the object-oriented programming should contain the concepts about past, current and future technologies and methods of creating the software products, as well as the general-scientific and industry-specific fundamental laws and categories. The list of basic concepts of object-oriented programming and basic general-scientific and branch laws and categories for their substantiation is given. The second condition of fundamentalizing the content of learning the object-oriented programming is the presentation of basic concepts indicating the purpose, structure, principle of action, characteristics and the fundamental general-scientific and branch laws and categories. The third condition of the fundamentalization of education is the provision of productive educational and cognitive activity of students, which reflects the process of designing, modeling and developing the new or improved existing software, both in practical and in lecture classes. The organization method of productive educational and cognitive activity of students during lectures and practical classes is described. The method and didactic means of training for the improvement of existing ones and the creation of new IT products are developed. The implementation of the method and didactic means for improving the basic IT product during the learning of object-oriented programming during the lecture classes is considered on the example of the topic "Organization of inheritance mechanisms in classes". An example of applying the method of creating new IT products for organizing the productive educational and cognitive activity of students at practical classes during the studying of the topic "The concept of virtual functions and polymorphism" is given. Consequently, the goals, content, methods and didactic means of learning the object-oriented programming based on the fundamentalized approach will allow students to develop the professional competence of IT professionals to perform the productive professional activities in designing and creating new computer software.

Keywords: teaching methodology; fundamentalization; an IT specialist; the object-oriented programming; learning goals, a content of training; teaching methods; didactic means; productive activity.

Постановка проблеми. Підвищення якості професійної підготовки майбутніх фахівців у галузі інформаційних технологій (ІТ-фахівців) до продуктивної діяльності вимагає розроблення методики фундаменталізації комп'ютерних

дисциплін, зокрема навчальної дисципліни "Об'єктно-орієнтоване програмування (ООП)".

Аналіз основних досліджень і публікацій. Нами в роботі [6] обґрунтовано концепцію диференційно-інтегративної фундаменталізації

професійної підготовки майбутніх ІТ-фахівців до продуктивної діяльності на основі філософських, природничих та математичних законів і понять. В роботах [3 – 5] нами розроблено зміст, методи та дидактичні засоби фундаменталізованого навчання майбутніх ІТ-фахівців. Перелічені компоненти методичної системи фундаменталізованої професійної підготовки майбутніх ІТ-фахівців розроблені у вигляді узагальнених структурно-функціональних моделей, які для застосування при навчанні об'єктно-орієнтованому програмуванню потребують конкретизації відповідно до предметної галузі навчальної дисципліни.

Метою статті є розробка методики фундаменталізованого навчання об'єктно-орієнтованому програмуванню майбутніх фахівців у галузі інформаційних технологій.

Виклад основного матеріалу. На сьогодні найпопулярнішою технологією створення програмних продуктів є програмування на основі об'єктно-орієнтованої парадигми, яка дозволяє просто і швидко підвищити їх продуктивність і надійність [1]. Тому при підготовці майбутніх ІТ-фахівців навчальна дисципліна “Об'єктно-орієнтоване програмування” є фундаментальною та обов'язковою для вивчення.

Метою фундаменталізованої професійної підготовки майбутніх ІТ-фахівців з об'єктно-орієнтованого програмування є формування у них професійних компетентностей виконувати продуктивну професійну діяльність з проектування програмного забезпечення, розробки архітектури, модулів та компонентів програмних систем.

Відповідно до поставленої мети та системи професійних компетентностей ІТ-фахівця [4] нами розроблено навчальні, розвивальні та виховні цілі навчання об'єктно-орієнтованому програмуванню.

Навчальними цілями дисципліни “Об'єктно-орієнтоване програмування” є: засвоєння необхідних знань з опанування сучасними технологіями аналізу, проектування та об'єктно-орієнтованої розробки програмних систем, вивчення складових частин об'єктно-орієнтованої парадигми, об'єктно-орієнтованої моделі та синтаксису мови об'єктно-орієнтованого програмування C++ (C#, Java), а також застосування об'єктно-орієнтованого підходу та методів узагальненого програмування до проектування і реалізації програмного забезпечення; отримання навичок використання класів, механізмів наслідування, інкапсуляції, поліморфізму при самостійній побудові програм

різної складності з використанням структурно-модульного та об'єктно-орієнтованого методу програмування.

Традиційний зміст навчання об'єктно-орієнтованому програмуванню спрямований на те, щоб навчити студентів об'єктно-орієнтованої технології програмування на прикладі обраної мови програмування C++ (Java, C#), та охоплює наступні теми: “Парадигми програмування”, “Класи”, “Організація класів і особливості роботи з об'єктами”, “Особливості механізму переважання операторів”, “Організація механізмів успадкування в класах”, “Поняття про віртуальні функції та поліморфізм”, “Робота з шаблонними функціями та класами” [7, 8].

Відповідно до розробленої нами моделі змісту фундаменталізованої професійної підготовки майбутніх ІТ-фахівців [5] зміст навчання об'єктно-орієнтованому програмуванню має містити поняття про минулі, сучасні та перспективні технології та методи створення програмних продуктів, а також загально-наукові і галузеві фундаментальні закони й категорії. Фундаментальними загально-науковими засадами об'єктно-орієнтованого програмування є філософські та математичні закони і категорії, які лежать в основі проектування, моделювання та реалізації програмних продуктів на основі об'єктно-орієнтованої парадигми. До галузевих фундаментальних законів і категорій належать поняття з інших комп'ютерних дисциплін та раніше вивчені з цієї ж дисципліни. В (табл. 1) нами наведено список базових понять з об'єктно-орієнтованого програмування та фундаментальні загально-наукові і галузеві закони й категорії для їх обґрунтування.

Однією з умов фундаменталізації змісту навчання об'єктно-орієнтованому програмуванню є структурування навчального матеріалу на основі виділення логічної структури, систематизації і впорядкування знань, визначення фундаментальних понять і зв'язків між ними. Для цього кожне базове поняття про об'єктно-орієнтоване програмування необхідно подавати у вигляді моделі: $P = \{R(f), S(f), D(f), H(f)\}$, де $R(f)$ – призначення, $S(f)$ – структура, $D(f)$ – принцип дії, $H(f)$ – характеристики, f – фундаментальні загально-наукові та галузеві закони і категорії.

Нарис. 1 представлено приклад фундаменталізованого поняття навчальної дисципліни “Об'єктно-орієнтоване програмування” у вигляді семантичної моделі поняття “клас”.

Ще однією умовою фундаменталізації освіти є забезпечення продуктивної навчально-пізнавальної діяльності студентів, яка відображає

**МЕТОДИКА ФУНДАМЕНТАЛІЗОВАНОГО НАВЧАННЯ ОБ'ЄКТНО-ОРІЄНТОВАНОМУ
ПРОГРАМУВАННЮ МАЙБУТНІХ ФАХІВЦІВ У ГАЛУЗІ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ**

Таблиця 1.

Список базових понять навчальної дисципліни “Об’єктно-орієнтоване програмування” та фундаментальних загально-наукових і галузевих законів і категорій для їх обґрунтування

ІТ-поняття	Загально-науковий фундамент	Галузевий фундамент
Інкапсуляція	<i>філософські категорії</i> : явище, сутність	абстракція даних
Клас	<i>філософські категорії</i> : система, елемент, структура, загальне, конкретне, особливе; <i>математичні теорії та категорії</i> : множина, теорія автоматів	
Об’єкт (Екземпляр класу)	<i>філософські категорії</i> : система, елемент, структура, загальне, конкретне, особливе	клас
Член даних	<i>філософські категорії</i> : частина, ціле, система, елемент, структура	клас
Функція-член (Метод класу)	<i>філософські категорії</i> : частина, ціле; <i>математичні теорії та категорії</i> : теорія автоматів, функція	клас,
Конструктор	<i>філософський закон</i> : єдності та боротьби протилежностей	клас, функція-член
Деструктор	<i>філософські закони</i> : єдності та боротьби протилежностей, взаємного переходу кількісних і якісних змін	клас, функція-член,
Доступ до членів класу	<i>філософські категорії</i> : явище, сутність	інкапсуляція, клас,
Успадкування	<i>філософські категорії</i> : загальне, конкретне, особливе; <i>математичні категорії</i> : надмножина, підмножина, перетин множин, орієнтовані ациклічні графи	
Базовий клас	<i>математична категорія</i> : підмножина	успадкування
Похідний клас	<i>математична категорія</i> : надмножина	успадкування
Поліморфізм	<i>філософський закон</i> : заперечення заперечення	абстракція даних
Перевантаження		успадкування, поліморфізм
Віртуальна функція		успадкування, поліморфізм, функція-член
Узагальнена функція	<i>філософські категорії</i> : загальне, конкретне, особливе; <i>математичні категорії</i> : функція, функція вищого порядку,	абстракція даних
Узагальнений клас	<i>філософські категорії</i> : загальне, конкретне, особливе; <i>математична категорія</i> : функція вищого порядку	абстракція даних, клас

процес проектування, моделювання та розроблення нового чи удосконалення вже існуючого програмного забезпечення, як на практичних, так і на лекційних заняттях.

Через те, що навчально-пізнавальна діяльність студента у ВНЗ відображає процес набуття суб’єктивно нових знань про предмети, які об’єктивно вже відомі, тому наведена вище модель $P=\{R(f), S(f), D(f), H(f)\}$ може бути використана для представлення понять про існуючі методи та технології об’єктно-орієнтованого програмування під час

продуктивної навчально-пізнавальної діяльності студентів.

Оскільки для створення нових програмних продуктів з покращеними характеристиками на основі вже існуючих достатньо змінити структуру або принцип дії останніх, тому при організації продуктивної діяльності студентів із засвоєння нового матеріалу доцільно базові поняття про методи та технології об’єктно-орієнтованого програмування виводити на основі раніше вивчених.

При цьому, доцільно застосувати ретроспективний

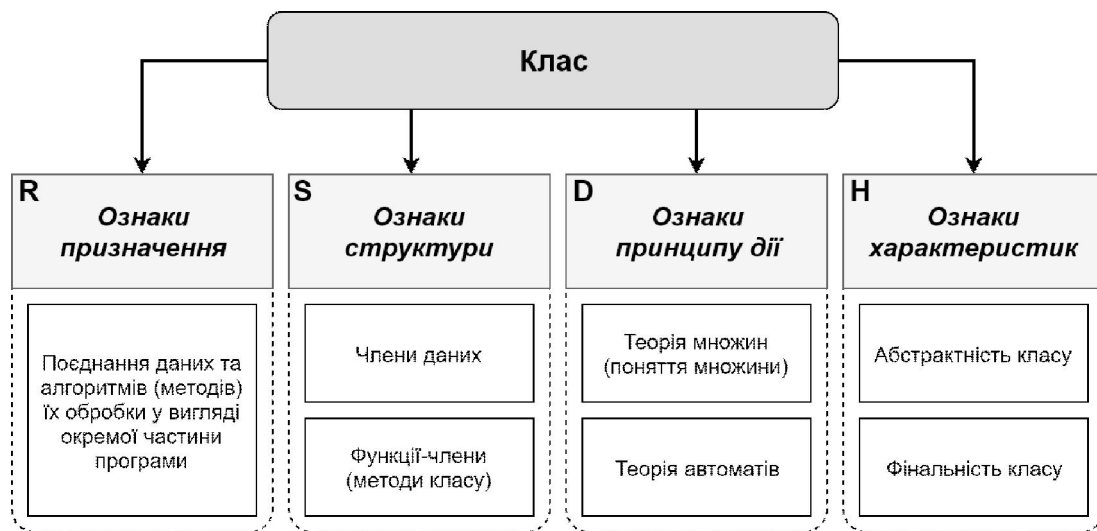


Рис. 1. Семантична модель поняття “клас”

аналіз розвитку методів і технологій програмування, а також відомого програмного забезпечення [2]. Системний ретроспективний аналіз методів і технологій програмування дозволяє визначити основні причини виникнення (нові характеристики) кожного покоління із зазначенням нового призначення, структури, принципу дії, та згодом визначити перспективи їх розвитку.

Наступним кроком, розглянемо методи та дидактичні засоби, які доцільно використовувати для організації продуктивної навчально-пізнавальної діяльності студентів на лекційних та практичних заняттях.

Нами у роботі [3] розроблено метод та дидактичні засоби навчання удосконаленню вже існуючих та створенню нових ІТ-продуктів.

Розглянемо реалізацію методу та відповідних дидактичних засобів удосконалення базового ІТ-продукту при навчанні об'єктно-орієнтованому програмуванню під час лекційних занять на прикладі теми “Організація механізмів успадкування в класах”.

Перше знайомство студентів із поняттям “успадкування в класах” доцільно здійснити на лекції, запропонувавши їм розв’язати наступне завдання: “Два класи з частково ідентичними властивостями та функціональністю, наприклад “Автомобіль” та “Мотоцикл”. Вони мають колеса та двигун, можуть рухатись за їх допомогою по дорозі. Але їх кількісні та якісні показники є різними. Крім того, вони мають велику різницю у загальній компоновці частин корпусу та містять специфічні структури та функціональність (наприклад, автомобіль має кузов та двері, а мотоцикл – підставку для підтримки у нерухомому стані).

Дублювання коду призводить до ускладнення програми, збільшенню її розміру та появленню потенційних помилок у майбутньому при модифікаціях такого коду або додаванні інших класів, що також будуть поділяти ці властивості та функціональність. Треба реорганізувати їх структуру таким чином, щоб уникнути дублювання коду та даних у обох класах і забезпечити надійний механізм модифікації загального для всіх класів коду”.

Викладач за допомогою евристичних питань повинен підвести студентів до наступних висновків.

Кожен клас можна представити як множину властивостей цього класу (значень та функцій). Всі об’єкти (екземпляри) одного класу містять один і той самий набір цих властивостей. Якщо два класи A та M мають частково однакові властивості, їх можна представити у вигляді множин властивостей, що частково перетинаються: $A \cap M = S$, де S – це множина властивостей, що містяться у обох вихідних класах.

Таким чином, $S \subset A$ $A \cap S \subset M$, тобто S є підмножиною множин A та M .

Рішенням задачі буде створення нового класу на основі множини S , що буде містити загальні властивості для обох вихідних класів, та забезпечення зв’язку між класами таким чином, щоб такі зв’язані класи могли розділяти одну й ту ж саму функціональність.

З позицій філософських категорій класи A та M є проявом категорії “особливе” відносно класу S , для яких він, в свою чергу, буде проявом категорії “загальне”. Конкретні об’єкти класів A та M будуть проявом категорії “конкретне”.

Далі викладач робить висновок, що у об'єктно-орієнтованому програмуванні такий зв'язок між класами має назву "успадкування". Клас, що містить загальні властивості, має назву "базовий клас", або "суперклас", а класи, що наслідують ці властивості та додають свої власні, мають назву "похідні класи", або "підкласи".

Класи можуть бути організовані у деревовидній ієрархії, де деякі з них можуть бути як підкласами для одних, так і суперкласами для інших класів. Таку ієрархію можна представити у вигляді спрямованого ациклічного графу з напівстепенем 1 (деревовидного графу).

Всі властивості (значення та функції), що успадковані класом від своїх суперкласів, доступні для зовнішнього користувача класу як власні властивості, тобто для нього не має різниці, де саме (у якому з класів) визначена та чи інша властивість – вона завжди доступна таким чином, ніби міститься безпосередньо у цьому конкретному класі.

Узагальнити та підсумувати теоретичний матеріал доцільно, побудувавши разом із студентами семантичну модель поняття "успадкування класів", наведену на рис. 2.

Наступним кроком розглянемо застосування методу створення нових ІТ-продуктів [3] для організації продуктивної навчально-пізнавальної діяльності студентів на практичних заняттях на прикладі теми "Поняття про віртуальні функції та поліморфізм".

Прикладом такої задачі може бути: "Створити

програму на С++, використовуючи ООП-парадигму, що відображає на екрані довільну кількість геометричних фігур, запитуючи у користувача такі дані:

- 1) тип геометричної фігури (прямокутник, рівносторонній трикутник, коло);
- 2) координати (x, y) центру фігури;
- 3) дані, специфічні для фігур: висота (h) та ширина (w) для прямокутника, довжина сторони (l) для рівностороннього трикутника, радіус (r) для кола;
- 4) колір фігури".

Розв'язати цю задачу відповідно до методу [3] можна за наступним алгоритмом.

Спочатку треба створити модель предметної області, яка буде реалізована в рамках даної задачі. Очевидно, що до неї входять три сутності – прямокутник, рівносторонній трикутник та коло. Кожна з цих сутностей може бути представлена окремим класом – Rectangle, Triangle, Circle.

Аналізуючи склад властивостей кожного з класів, можна дійти висновку, що деякі з них є загальними (філософська категорія "Загальне") для всіх класів, а саме – координати центру фігури та її колір. Решта властивостей є специфічними (філософська категорія "Особливе") для кожного з класів. Загальні для всіх класів властивості треба виділити в окрему сутність – створити загальний клас "Геометрична фігура" (Figure), що буде базовим класом для решти класів.

Для того, щоб вивести на екран фігуру, треба надати кожній з них метод для такої операції. Цей



Рис. 2. Семантична модель поняття "успадкування класів"

метод буде робити одне й те ж саме – малювати свою фігуру на екрані, тому, з одного боку, він має міститися у базовому класі (Figure). З іншого боку, спосіб виведення для кожної фігури повинен бути свій, оскільки кожна фігура має власні особливості. Тому цей метод однозначно має бути реалізований у кожному з специфічних класів геометричних фігур. Для того, щоб застосувати такий підхід, треба використати таку специфічну ООП-конструкцію, характерну для C++, як віртуальний метод. У базовому класі Figure треба створити віртуальний метод `gender()`. Оскільки реалізації цього методу у базовому класі не буде, він повинен бути не просто віртуальним, а суто віртуальним, тобто створеним з особливою декларацією (`=0`), що вказує на відсутність реалізації у даному класі та обов'язковість такої реалізації у всіх похідних класах. У решті класів, що представляють конкретні геометричні фігури, треба задекларувати той самий метод, але вже з повноцінною реалізацією, специфічною для даної фігури.

Останнім кроком треба створити частину загальної програми, що буде циклічно запитувати дані у користувача, створювати екземпляри відповідних класів геометричних фігур та викликати їх метод `gender()` для виведення їх на екран.

Висновки з даного дослідження і перспективи подальших розвідок у даному напрямку. Отже, розроблені на основі фундаменталізованого підходу цілі, зміст, методи та дидактичні засоби навчання об'єктно-орієнтованому програмуванню дозволять сформулювати у студентів професійні компетентності ІТ-фахівця виконувати продуктивну професійну діяльність з проектування і створення нового програмного забезпечення комп'ютерної техніки.

ЛІТЕРАТУРА

1. Бадд Т. Объектно-ориентированное программирование в действии / Т. Бадд. – СПб.: Питер, 1997. – 464 с.

2. Бардус І. О. Визначення закономірностей розвитку програмного забезпечення комп'ютерної техніки як умова фундаменталізації професійної підготовки майбутніх фахівців у галузі інформаційних технологій / І. О. Бардус // Наукові записки Бердянського державного педагогічного університету. Педагогічні науки: зб. наук. пр. – Бердянськ: ФО-П Ткачук О.В., 2017. – Вип. 3. – С. 94–101.

3. Бардус І. О. Методи фундаменталізованого навчання майбутніх фахівців у галузі інформаційних технологій виконувати продуктивну діяльність / І. О. Бардус // Ключові питання наукових досліджень у сфері педагогіки та психології у ХХІ ст.: Збірник тез наукових робіт учасників міжнародної науково-

практичної конференції (м. Львів, 26–27 січня 2018 року). – Львів: ГО “Львівська педагогічна спільнота”, 2018. – Ч.1. – С. 53–56.

4. Бардус І. О. Структура та зміст професійної компетентності фахівців у галузі інформаційних технологій / І. О. Бардус // Проблеми інженерно-педагогічної освіти. Збірник наукових праць. Вип. 54–55. – Харків, Українська інженерно-педагогічна академія (УІПА), 2017. – С. 55–64.

5. Бардус І. О. Фундаменталізація змісту професійної підготовки майбутніх фахівців у галузі інформаційних технологій до продуктивної діяльності / І. О. Бардус // Наукові записки Тернопільського національного педагогічного університету імені Володимира Гнатюка. Серія: педагогіка. – 2017. – №3. – С. 74–81.

6. Бардус І. О. Філософські засади концепції фундаменталізації професійної підготовки майбутніх фахівців у галузі інформаційних технологій / І. О. Бардус // Проблеми інженерно-педагогічної освіти. Збірник наукових праць. – Вип. 52–53. – Харків, Українська інженерно-педагогічна академія (УІПА), 2016. – С. 7–17.

7. Бублик В. В. Об'єктно-орієнтоване програмування: [Підручник] / В. В. Бублик. – К.: ІТ-книга, 2015. – 624 с.

8. Грицюк Ю. І. Об'єктно-орієнтоване програмування мовою C++ : навч. посібн. / Ю. І. Грицюк, Т. Є. Рак. – Львів: Вид-во ЛДУ БЖД, 2011. – 404 с.

REFERENCES

1. Badd, T. (1997). *Obektno-orientirovannoe programmirovaniye v deystvii* [Object-oriented programming in action]. St. Petersburg: Piter, 464 p. [in Russian].

2. Bardus, I. O. (2017). Vyznachennia zakonodirnostei rozvytku prohramnoho zabezpechennia kompiuternoї tekhniki yak umova fundamentalizatsii profesiinoї pidhotovky maibutnikh fakhivtsiv u haluzi informatsiinykh tekhnolohii [Determination of the regularities of the software evolution of computer engineering as the condition of the fundamentalization of professional training of future specialists in the field of information technologies]. *Scientific notes of the Berdyansk State Pedagogical University. Pedagogical sciences*. Berdiansk: FO-P Tkachuk O.V., vol. 3, pp. 94–101. [in Ukrainian].

3. Bardus, I. O. (2018). Metody fundamentalizovanoho navchannia maibutnikh fakhivtsiv u haluzi informatsiinykh tekhnolohii vykonuvaty produktyvnu diialnist [Methods of fundamental training of future IT professionals in productive activities]. *Key issues of scientific research in the field of pedagogy and psychology in the XXI century*. Lviv: HO “Lvivska pedahohichna spilnota”, vol. 1, pp. 53–56. [in Ukrainian].

4. Bardus, I. O. (2017). Struktura ta zmist profesiinoї kompetentnosti fakhivtsiv u haluzi informatsiinykh tekhnolohii [Structure and content of professional competence of specialists in the field of information technologies]. *Problems of engineering and pedagogical education*. Kharkiv: Ukrainska inzhenerno-pedahohichna akademiia (UIPA), vol. 54–55, pp. 55–64. [in Ukrainian].

**ПРОФЕСІЙНА ПІДГОТОВКА МАЙБУТНІХ УЧИТЕЛІВ ПОЧАТКОВОЇ ОСВІТИ
В УНІВЕРСИТЕТАХ ЧЕСЬКОЇ РЕСПУБЛІКИ**

5. Bardus, I. O. (2017). Fundamentalizatsiia zmistu profesiinoi pidhotovky maibutnikh fakhivtsiv u haluzi informatsiinykh tekhnolohii do produktyvnoi diialnosti [Fundamentalization of the content of professional training of future specialists in the field of information technologies to productive activity]. *Scientific Notes of Ternopil National Pedagogical University named after Volodymyr Hnatyuk. Series: pedagogy. Ternopil*, vol. 3, pp. 74–81. [in Ukrainian].

6. Bardus, I. O. (2016). Filosofski zasady kontseptsii fundamentalizatsii profesiinoi pidhotovky maibutnikh fakhivtsiv u haluzi informatsiinykh tekhnolohii

[Philosophical Foundations of concept of fundamentalization training of future specialists in information technology]. *Problems of engineering and pedagogical education*. Kharkiv: Ukrainska inzhenerno-pedahohichna akademiia (UIPA), vol. 52–53, pp. 7–17. [in Ukrainian].

7. Bublyk, V. V. (2015). *Obiektno-orientovane prohratuvannia* [Object-oriented programming]. Kyiv: IT-knyha, 624 p. [in Ukrainian].

8. Hrytsiuk, Yu. I. & Rak, T. Ye. (2011). *Obiektno-orientovane prohratuvannia movoiu S++* [Object-Oriented Programming in C ++]. Lviv: LDU BZhD Publ, 404 p. [in Ukrainian].

Стаття надійшла до редакції 19.03.2018

УДК [371.134+373.3]+378.4(437.1/2)

DOI:

Катерина Біницька, кандидат педагогічних наук, доцент кафедри педагогіки
Хмельницької гуманітарно-педагогічної академії

**ПРОФЕСІЙНА ПІДГОТОВКА МАЙБУТНІХ УЧИТЕЛІВ ПОЧАТКОВОЇ ОСВІТИ
В УНІВЕРСИТЕТАХ ЧЕСЬКОЇ РЕСПУБЛІКИ**

У статті розглянуто окремі аспекти професійної підготовки майбутніх учителів початкової освіти в університетах Чеської Республіки. Визначено, що у 90-х роках ХХ ст. розпочалась реформа освіти. Зазначено, що Болонський процес став головним імпульсом до початку освітніх перетворень для закладів вищої освіти в країні. Проаналізовано нормативно-правову базу Чеської Республіки та акцентовано, що законодавчо закріплена вимога до вчителя початкової освіти мати ступінь вищої освіти “магістр”.

Зазначено, що університети використовують комбіновану магістерську програму з тривалістю навчання 5 років.

Ключові слова: професійна підготовка; учителі початкової освіти; навчальний план; університети; Чеська Республіка.

Лім. 14.

**Kateryna Binytska, Ph.D. (Pedagogy), Associate Professor of the Pedagogy Department
Khmelniiskiy Humanitarian Pedagogical Academy**

**PROFESSIONAL PREPARATION OF THE FUTURE TEACHERS OF PRIMARY
EDUCATION AT THE UNIVERSITIES OF THE CZECH REPUBLIC**

The article examines some aspects of the professional training of the future primary school teachers in the universities of the Czech Republic. It is determined that in the 1990-s the reform of education began, which included modernization of the professional training of the future primary school teachers. It is generalized that from the point of view of professional pedagogical preparation, the reorganization was aimed at eliminating the unified approach to teachers training. It is found out that in the late 1990-s in the Czech Republic, the processes of modernizing of professional training of the future primary school teachers were implemented at 9 pedagogical faculties of universities.

It is mentioned that at the beginning of the XXI century the Bologna process became the major impetus to the beginning of educational transformations for the institutions of higher education in the Czech Republic. In Czechia, due to the requirement of the Berlin Communique on the introduction of the two-staged system of higher education at the pedagogical faculties of universities began to be carried out already since 2003/2004 academic year.

The legal framework of the Czech Republic has been analyzed in the article. It is emphasized that the Law “On Pedagogical Workers” establishes the requirement to the teacher of primary education to have a master’s degree of higher education.

The analysis showed that, in addition to the partial reorganization in the education system, the complete modification of the curricula for the training of future primary school teachers has been conducted. It is found out that the curriculum consists of five components: general educational, subject (academic), subject didactic, pedagogical-psychological and specialization.

It is noted that the process of professional training of the future teacher of primary education is clearly structured. The total proportion of compulsory disciplines, disciplines by choice of the student and optional