

АНАЛІЗ ЗМІСТУ НАВЧАННЯ ПРОГРАМУВАННЯ ІГОР МАЙБУТНІХ ІНЖЕНЕРІВ-ПРОГРАМІСТІВ

Володимир Ракович

Мелітопольський державний педагогічний університет імені Богдана Хмельницького

Анотація:

Одним з напрямів підготовки майбутніх інженерів програмістів є навчання програмування ігор, що пояснюється швидкими темпами розвитку ІТ-індустрії і попитом на фахівців у галузі розробки ігор. Питання формування змісту навчання програмування ігор майбутніх інженерів-програмістів докладно не розглядалось у наукових працях, тому вимагає дослідження. На основі аналізу змісту навчання програмування ігор, що міститься в підручниках і посібниках різних авторів (А. Хонич, Т. Міллер, М. Моррісон, В. Касихін, С. Маниш, М. Цехнер, М. Доусон), і його відповідності наведеним принципам відбору сформульовано висновок про те, що ці матеріали лише частково можуть бути використані при доборі змісту навчання програмування ігор майбутніх інженерів-програмістів, а тому доцільно формувати власний зміст.

Аннотация:

Ракович Владимир. Анализ содержания обучения программированию игр будущих инженеров-программистов.

Одним из направлений подготовки будущих инженеров программистов является обучение программированию игр, что объясняется быстрыми темпами развития ИТ-индустрии и спросом на специалистов в области разработки игр. Вопросы формирования содержания обучения программированию игр будущих инженеров-программистов подробно не рассматривались в научных трудах, поэтому требуют исследования. На основе анализа содержания обучения программированию игр, представленного в учебниках и пособиях различных авторов (А. Хонич, Т. Миллер, М. Моррисон, В. Касихин, С. Маниш, М. Цехнер, М. Доусон), и его соответствия приведенным принципам отбора сделан вывод о том, что эти материалы лишь частично могут быть использованы при отборе содержания обучения программированию игр будущих инженеров-программистов, а потому целесообразно формировать собственное содержание.

Resume:

Rakovych Volodymyr. Analysis of the content of training programming games for future software engineers.

One of the training areas for future program engineers is training in game programming, which could be explained by the rapid pace of the IT industry development and the demand for specialists in the field of game development. The formation of the content of teaching the programming of games for future software engineers was not considered in detail in scientific papers, and therefore they require research. Based on the analysis of the content of teaching programming games, presented in textbooks and papers of various authors (A. Khonich, T. Miller, M. Morrison, V. Kasikhin, S. Manish, M. Tsekhner, M. Dawson), and its compliance with the principles selection the author concluded that these materials can only partially be used in selecting the content of training programming games for future software engineers, and therefore it is appropriate to create one's own content.

Ключові слова:

зміст навчання програмування; програмування ігор; майбутні інженери-програмісти.

Ключевые слова:

содержание обучения программирования; программирование игр; будущие инженеры-программисты.

Key words:

content of teaching programming; game programming; future software engineers.

Постановка проблеми. У листі МОНмолодьспорту України «Щодо покращення якості підготовки фахівців для ІТ-галузі» (2012 р.) було наголошено на необхідності перегляду змісту нормативних дисциплін, які викладаються в процесі підготовки фахівців, відповідно до сучасних досягнень розвитку інформаційних технологій [6]. Отже, формування змісту дисциплін професійного спрямування в процесі підготовки майбутніх інженерів-програмістів має важливе значення. Одним з напрямів такої підготовки є навчання програмування ігор, що пояснюється швидкими темпами розвитку ІТ-індустрії та попитом на фахівців у галузі розробки ігор.

Аналіз наукових і методичних праць свідчить про те, що змісту вищої професійної освіти присвятили свої праці О. Андреев, Н. Брюханова, В. Гладуш, Л. Гризун, Г. Лисенко, А. Сбруева та ін. Зокрема О. Андреев визначив фактори, що впливають на зміст освіти, а саме: потреби суспільства, суб'єктивні чинники (політика, методологічна позиція вчених), науково-технічний прогрес (розвиток мікро- комп'ютерних

і телекомунікаційних засобів і систем), сучасні потреби власне системи освіти, інтереси бізнесу, інвестиції та ін. [1, с. 170]. Слушною є думка В. Гладуш і Г. Лисенко про те, що «коли йдеться про зміст освіти, який лежить в основі професійної підготовки, то мається на увазі, що він орієнтований на формування професійної і загальної культури фахівців у широкому розумінні» [2, с. 80].

Окремі питання навчання програмування ігор майбутніх інженерів-програмістів висвітлено в наукових працях Р. Агравала (R. Agrawal), Т. Байбака (Т. Baibak), П. Воробкалова, В. Касихіна, А. Катаєва, О. Шабаліної та ін. Зміст професійної підготовки майбутніх інженерів-програмістів розглядався в дослідженнях І. Крашеніннік, В. Круглика, В. Осадчого, С. Семерикова, З. Сейдаметової, О. Спіріна. Зокрема В. Круглик [5, с. 140–141] запропонував зміст професійної підготовки інженерів-програмістів в умовах ВНЗ, який складався з 16 дисциплін загальної та професійної підготовки як ядра професійної підготовки та загальних і фахових компетентностей, що формуються в процесі

вивчення цих дисциплін. В. Осадчий та І. Крашеніннік [11] порушили проблему формування змісту освітніх програм підготовки майбутніх інженерів-програмістів на засадах скороченого терміну навчання, результатом дослідження якої стала пропозиція проведення аудиту змісту спецкурсів і вибіркових дисциплін для посилення їх спрямування на ознайомлення студентів з сучасними технологіями та засобами програмної розробки. Проте питання формування змісту навчання програмування ігор майбутніх інженерів-програмістів докладно не було розглянуто.

Формулювання цілей статті. Мета статті – на основі аналізу науково-методичних матеріалів (підручників і посібників) узагальнити зміст навчання програмування ігор майбутніх інженерів-програмістів.

У ході проведення дослідження використано різні методи наукової роботи: аналіз науково-методичних праць з проблемних питань формування змісту навчання та програмування ігор, синтез навчально-методичних ідей; узагальнення власного педагогічного досвіду професійної підготовки майбутніх інженерів-програмістів; інтерпретація результатів дослідницької роботи.

Виклад основного матеріалу дослідження. У Мелітопольському державному педагогічному університеті імені Богдана Хмельницького на кафедрі інформатики і кібернетики навчання майбутніх інженерів-програмістів програмування ігор базується на змісті відповідних дисциплін, зокрема таких, як «Теорія ігор», «Основи ігрового програмування», «Розробка мобільних додатків».

Постає питання формування їх змісту як системи знань і вмінь, відібраних для вивчення в певному типі навчального закладу (І. Підласий) [12, с. 223].

Складність відбору змісту дисциплін зумовлена суперечностями, що пов'язані з великим обсягом інформації і традиційними підходами до визначення змісту навчання, а також зі швидким розвитком інформаційних технологій та повільними темпами надходження інформації про досягнення ІТ-індустрії до викладачів і студентів. Для усунення цих суперечностей у процесі відбору змісту вважаємо за доцільне дотримуватись принципів, визначених К. Осадчою [10, с. 100–106], а саме: фундаменталізації, орієнтації на компетентнісний підхід, уніфікації, варіативності й альтернативності, науковості, систематичності, доступності, практикоорієнтованості, цілісності, модульності, наочності, свідомості й самосвідомості.

Проаналізуємо на основі цих принципів зміст навчання програмування ігор, наведений у підручниках і посібниках різних авторів.

Зміст навчання програмування ігор, запропонований А. Хоничем [13] базується на вивченні мови C і влаштуванні тривимірної гри типу DOOM, на її структурі, усіх процедурах та їх взаємозв'язках, алгоритмах і проблемних місцях. Передбачене ознайомлення з такими теоретичними питаннями, як вибір інструментарію для розробки гри; методологія створення відеогри; ініціалізація зони пам'яті, файлових ресурсів, графічних даних, текстур, спрайтів, анімації, маніпулятора миші, джойстика, клавіатури; багаторежимність і багатофункціональність; синхронізація швидкісного таймера; завантаження даних у грі (ресурсів, об'єктів); основні віртуальні процеси в 3D іграх; обробка об'єктів і класів; основні принципи роботи з графікою та звуком.

Для розуміння матеріалу цього посібника студенти вже мають володіти мовою програмування C й орієнтуватись у 386-му Асемблері, щоб розуміти подані в ньому підпрограми на асемблері та макроси. В основному акцент у змісті цього посібника зроблений на формуванні вміння аналізувати поданий матеріал з метою розроблення власної програми. Зміст посібника ґрунтується на принципах фундаменталізації, доступності, систематичності та цілісності, оскільки автор розкриває методологію створення відеогри за типом DOOM, просто й докладно здійснює опис техніки написання ігор, а також будує систему книги, спрямовану на послідовне й логічне написання гри, починаючи з вибору інструментарію й завершуючи створенням власної графіки. Зміст посібника не розкриває багатьох виділених нами принципів відбору змісту навчання програмування майбутніх інженерів-програмістів, тому може бути застосований лише частково, зокрема в частині опису особливостей розробки відеоігор.

Т. Міллер [8] акцентує увагу на вивченні можливостей використання керованого DirectX під час розробки різноманітних графічних і мультимедійних додатків. У посібнику розглядаються як основи програмування 3D графіки, так і більш складні розділи, наприклад, керування рівнями деталізації mesh-об'єктів, використання високорівневої мови рейдерів і символічної анімації. Автор приділив достатньо уваги таким важливим питанням, як керування звуком і пристроями введення й додавання звукових ефектів і джойстика зі зворотним зв'язком. Матеріал посібника спрямований значною мірою на безпосереднє створення повнофункціональних мультимедійних додатків, ніж на вивчення синтаксису мови програмування

C# або Visual Basic.Net. Вивчення матеріалів посібника Т. Міллера дасть змогу студентам набути таких умінь: здійснювати вибір пристрою для розробки гри за допомогою Direct3D, створювати тривимірне зображення, використовувати технології рендерингу, застосовувати mesh-об'єкти для рендерингу складних моделей, використовувати керований DirectX для програмування ігор, керувати ресурсами гри, володіти методами побудови графіки (мова тендерів, шейдерів і рендеринг скелетної анімації), додавати звук у гру, керувати пристроями введення в грі, додавати мережні можливості (вузли, сесії, події та обробники), використовувати методи досягнення максимальної швидкодії гри. У змісті посібника Т. Міллера можна виокремити такі принципи добору змісту, як принцип фундаменталізації, систематичності, доступності, практикоорієнтованості, цілісності й наочності. Проте в ньому бракує принципів орієнтації на компетентнісний підхід, уніфікації, варіативності й альтернативності, науковості, модульності, свідомості й самосвідомості, тому у формуванні змісту навчання програмування майбутніх інженерів-програмістів цей посібник може бути використаний лише частково.

Практичне керівництво М. Моррісона [9] спрямовано на допомогу в розробленні й реалізації гри для мобільного телефону. Автор пропонує для розгляду такі теми: «Основи програмування ігор для мобільних телефонів», «Мобільна графіка 101», «Віртуальні ігри та штучний інтелект у мобільному телефоні», «Використання переваг роботи в мережі», «Оптимізація ігор». Їх вивчення дасть змогу студентам набути вмінь для використання спрайтової анімації, обробки введення користувача, відтворення цифрового звуку й музики, розроблення мобільних ігор на Java, створення ігрових шарів і керування ними, застосування переваг роботи в мережі, відладки та установки мобільних ігор, оптимізації Java-ігор. Книга написана простою мовою й не містить складної термінології програмування, крок за кроком ознайомлює з методикою створення технології «plug-and-play», яка застосовується для створення великої кількості ігор. Отже, у ній ураховано принципи доцільності, систематичності й цілісності. Через те, що автор уводить у текст докладні описи, приклади кодів для ігор та інформацію, що є необхідною для створення власної гри, вважаємо, що в посібнику реалізовано принцип практикоорієнтованості. Використання в книзі ілюстрацій дає змогу твердити про унаочнення навчального матеріалу, що розкриває принцип наочності в її змісті. Проте багато принципів, на нашу думку, не відображено в змісті

аналізованого керівництва, тому його можна використовувати лише частково в змісті дисципліни «Розробка мобільних додатків».

У керівництві В. Касихіна [4] містяться відповіді на запитання, якими знаннями має володіти розробник комп'ютерних ігор, які ігрові жанри є і які з них користуються популярністю, які складники має гра і які формати файлів використовуються в ній, як розробити маркетингову стратегію для рекламування гри на ринку. Як зауважує автор, ознайомлення з книгою допоможе навчитись не лише створювати гру самостійно, а й використовувати ігрові рушії, розробляти події й адрони на основі улюблених ігор. Аналіз змісту керівництва дає змогу констатувати, що при його віборі не дотримано принципів фундаменталізації, орієнтації на компетентнісний підхід, уніфікації, варіативності й альтернативності, науковості, модульності, свідомості й самосвідомості. Це дає змогу частково використовувати матеріал книги в процесі навчання програмування ігор майбутніх інженерів-програмістів.

У посібнику С. Маніша [7] зміст навчання програмування ігор базується на вивченні мови програмування BASIC і містить огляд таких питань, як ознайомлення з особливостями цієї мови та кодуванням з її допомогою циклів, функцій і масивів; поняття про змінних, умовних і логічних операторів; вивчення графічних засобів мови; засоби обробки звуків і музики в BASIC. Вивчаючи матеріал цього посібника, студенти набудуть таких умінь, як робота з пікселями в BASIC, додавання звуків і відео в гру, використання елементів штучного інтелекту в грі, дотримання етики програмування (написання коментарів, використання доцільних назв змінних), створення графічних зображень, робота з кольором у процесі програмування ігор, застосування буферів у процесі малювання в BASIC, програмування зображень (перетворення, зсув, масштабування, зміщення фонів), використання растрових зображень в анімації (створення растрових зображень, відображення руху), виявлення конфліктів під час взаємодії об'єктів у грі. Проаналізувавши зміст посібника, ми дійшли висновку, що в ньому враховані такі принципи: принцип фундаменталізації, адже вивчають класичні мовні конструкції; принцип доступності, оскільки викладається матеріал на зрозумілому рівні; принцип цілісності, що полягає в забезпеченні єдності окремих частин посібника; принцип наочності, бо посібник містить графічний матеріал; принцип практикоорієнтованості, тому що автор наводить практичні приклади, а також до посібника

додається компакт-диск з необхідними для створення власної гри програмами, вихідні дані для виконання практичних вправ тощо. Попри те, що в змісті посібника С. Маніша враховано багато принципів, усе-таки частина їх залишилась поза увагою автора, тому ми вважаємо, що книгу можна використовувати для відбору змісту навчання програмування ігор майбутніх інженерів-програмістів лише частково.

Зміст навчання програмування ігор, запропонований М. Цехнером [14], зосереджений навколо питань розробки ігор винятково під операційну систему Android. Автор пропонує вивчення фундаментальних питань проектування ігор, основ програмування, способів створення ігрового рушія й цікавих ігор. У посібнику вміщено докладний опис усього процесу створення ігор для платформи Android: налаштування й використання інструментів для розробки в Android, класичне програмування двомірних ігор і створення власних екшенів та ігорплатформерів, графіка й аудіо для Android, ігрова механіка (виявлення зіткнень, фізика й спрайтова анімація), додавання в ігри тривимірних ефектів, опублікування гри, організація технічної підтримки для користувачів. На практикоорієнтованість змісту посібника вказує те, що автор з метою навчання студентів створив проект у Google Code, який містить такі елементи: увесь програмний код і активи під ліцензією GPL 3 доступні з репозиторію Subversion проекту; невелике текстове керівництво, яке містить опис процесу імпорту проектів у Eclipse, а також відеоролик, що демонструє те саме; трекер проблем, що виникають, за допомогою якого можна повідомляти про всі помилки в самій книзі або в програмному коді; група обговорення книги, членом якої може стати кожен. Крім того, аналіз змісту посібника М. Цехнера дає змогу твердити, що в ньому враховані такі принципи: принцип фундаменталізації, принцип орієнтації на компетентнісний підхід, принцип науковості, систематичності, доступності, цілісності, наочності, проте не дотримані принципи уніфікації, варіативності й альтернативності, модульності, свідомості й самосвідомості. Саме тому матеріали посібника можуть бути використані частково в змісті дисципліни «Розробка мобільних додатків».

Інший підхід до змісту навчання програмування ігор пропонує М. Доусон [3]. Він наголошує на необхідності вивчити мову C++ для того, щоб уміти програмувати першокласні ігри. Кожен розділ розробленого М. Доусоном посібника містить опис самостійного ігрового проекту. Перший розділ «Типи, змінні,

стандартне введення-виведення» ознайомлює з основами C++ як найважливішою мовою програмування, що використовується в ігровій індустрії. У другому розділі «Істина, розгалуження та ігровий цикл» розглядається гра, яка виконує, пропускає або повторно виконує певні блоки коду, залежно від тих чи інших умов. Третій розділ «Цикли for, рядки й масиви» ознайомлює з ігровими послідовностями, рядками, програмними об'єктами. Четвертий розділ «Бібліотека стандартних шаблонів» присвячений ознайомленню з потужною бібліотекою, що дає змогу об'єднувати елементи в колекції й зберігати їх у такому вигляді, наприклад, як спорядження в рюкзаку персонажа. У п'ятому розділі «Функції. Гра "Божевільні бібліотекарі"» наводиться опис використання функцій як елементарних логічних одиниць, застосовуваних у програмах. Шостий розділ під назвою «Посилання» присвячено ознайомленню з посиланнями в грі та з елементами штучного інтелекту. Сьомий розділ «Покажчики» містить опис деяких найбільш низькорівневих і потужних можливостей мови C++. У наступних двох розділах наводиться концепція парадигми об'єктно-орієнтованого програмування. У десятому розділі «Успадкування й поліморфізм» розглядається процес створення одних об'єктів за допомогою інших.

Вивчаючи матеріал цього посібника, студенти навчаться відображати виведення у вікні консолі; виконувати арифметичні обчислення; працювати зі змінними й отримувати призначене для користувача введення, набране на клавіатурі; генерувати випадкові числа, що дасть змогу додавати в ігри елемент непередбачуваності; використовувати ігровий цикл, цикли, масиви, програмні об'єкти та функції; розділяти ігрові програми на компактні блоки, з якими зручно працювати; використовувати в грі елементи штучного інтелекту; звертатись безпосередньо до комп'ютерної пам'яті й маніпулювати нею; створювати ігри з використанням парадигми об'єктно-орієнтованого програмування. Аналіз змісту посібника М. Доусона дає змогу твердити, що в ньому враховані принципи фундаменталізації, науковості, систематичності, доступності, практикоорієнтованості, цілісності, наочності, проте не розкриті принципи орієнтації на компетентнісний підхід, уніфікації, варіативності й альтернативності, модульності, свідомості й самосвідомості. З огляду на це, проаналізований посібник також можна використовувати в змісті навчання програмування ігор майбутніх інженерів-програмістів лише частково.

Висновки. Отже, зміст навчання програмування ігор, презентований у проаналізованих нами підручниках і посібниках різних авторів, дає змогу дійти висновку, що жоден з них не може бути вичерпним для викладання ігрового програмування студентам ІТ-спеціальностей через не дотримання у відборі змісту всіх

наведених нами принципів. Проте їх матеріали частково можуть бути використані в процесі добору змісту навчання програмування ігор майбутніх інженерів-програмістів.

Перспективу подальших досліджень вбачаємо у формуванні власного змісту навчання програмування ігор майбутніх інженерів-програмістів.

Список використаних джерел

1. Андреев А. А. Педагогика высшей школы. Новый курс. Москва: Московский международный институт эконометрики, информатики, финансов и права, 2002. 264 с.
2. Гладуш В. А., Лисенко Г. І. Педагогіка вищої школи: теорія, практика, історія: навч. посібник. Дніпропетровськ: Акцент ПП, 2014. 416 с.
3. Доусон М. Изучаем C++ через программирование игр. Санкт-Петербург: Питер, 2016. 352 с.
4. Касихин В. В. Как стать создателем компьютерных игр. Краткое руководство. Москва: Издательский дом «Вильямс», 2006. 288 с.
5. Круглик В. С. Система підготовки майбутніх інженерів-програмістів до професійної діяльності у вищих навчальних закладах: монографія. Мелітополь: МДПУ ім. Б. Хмельницького, 2017. 384 с.
6. Лист МОНмолодьспорт «Щодо покращення якості підготовки фахівців для ІТ-галузі» від 16. 02. 2012 р. № 1/9-119. URL: http://osvita.ua/legislation/Vishya_osvita/27674 (дата звернення: 28. 04. 2018).
7. Маниш С. Программирование игр. Москва: 100 книг, 2007. 384 с.
8. Миллер Т. DirectX 9 с управляемым кодом. Программирование игр и графика. Москва: Издательский дом «КомБук», 2005. 386 с.
9. Моррисон М. Создание игр для мобильных телефонов. Москва: Издательский дом «ДМК-пресс», 2006. 496 с.
10. Осадча К. П. Формування професійної компетентності майбутніх учителів інформатики у процесі вивчення фахових дисциплін: дис. ... канд. пед. наук: 13.00.04 / Вінницький держ. пед. ун-т ім. М. Коцюбинського; Вінниця, 2010. 420 с.
11. Осадчий В. В., Крашеніннік І. В. Формування змісту освітніх програм підготовки майбутніх інженерів-програмістів за скороченим терміном навчання на основі аналізу ринку праці. *Інформаційні технології і засоби навчання*. 2017. Том 58. № 2. С. 11–25.
12. Подласый И. П. Педагогика в 2 т. Т. 1: Теоретическая педагогика. В 2 кн. Кн. 1: учебник и практикум для академического бакалавриата. Москва: Юрайт, 2016. 404 с.
13. Хонич А. Как самому создать трехмерную игру. Москва: Фирма «МИКРОАРТ», 1996. 290 с.
14. Цехнер М. Программирование игр под Android. Санкт-Петербург: Питер, 2013. 688 с.

Рецензент: д.пед.н., доцент Прийма С.М.

Відомості про автора:

Ракович Володимир Анатолійович
 rackovychv@gmail.com
 Мелітопольський державний педагогічний
 університет імені Богдана Хмельницького
 вул. Гетьманська, 20, м. Мелітополь,
 Запорізька обл., 72312, Україна

doi: 10.7905/nvmdpu.v0i20.2481

References

1. Andreyev, A.A. (2002). *Pedagogy of higher education. New course*. Moscow: Moskovskiy mezhdunarodnyy institut ekonometriki, informatiki, finansov i prava. [in Russian]
2. Hladush, V.A., Lysenko, H. I. (2014). *Higher School Pedagogy: Theory, Practice, History. Teaching guide*. Dnipropetrovsk: Aktsent PP. [in Ukrainian]
3. Dawson, M. (2016). *Beginning C++ Through Game Programming*. SPb.: Piter. [in Russian]
4. Kasikhin, V.V. (2006). *How to become a creator of computer games. Quick Start Guide*. Moscow: Izdatelskiy dom "Viliams". [in Russian]
5. Kruhlyk, V. S. (2017). *The system of training of future software engineers for the professional activity at higher educational institutions: monograph*. Melitopol: MDPU im. B. Khmelnytskoho. [in Ukrainian]
6. Letter of the Ministry of Education, Science, Youth and Sports «On improving the quality of training specialists for the IT industry» dated February 16, 2012 No 1 / 9-119. Retrieved from: http://osvita.ua/legislation/Vishya_osvita/27674/. [in Ukrainian]
7. Manish, S. (2007). *Game Programming*. Moscow: 100 knig. [in Russian]
8. Miller, T. (2005). *DirectX 9 with managed code. Programming games and graphics*. Moscow: Izdatelskiy Dom "KomBuk". [in Russian]
9. Morrison, M. (2006). *Creating games for mobile phones*. Moscow: Izdatelskiy dom DMK-press. [in Russian]
10. Osadcha, K.P. (2010). *Professional competence of computer science teachers' formation in the process of special disciplines' study: thesis*. Vinnytsya. [in Ukrainian]
11. Osadchyi, V.V. (2017). *Formation of short-cycle curricula content for software engineer training on the basis of the labour market analysis*. In: Information Technologies and Learning Tools. 58 (2), 11–25. [in Ukrainian]
12. Podlasyi, I.P. (2016). *Pedagogy in 2 vol. Vol.1: Theoretical pedagogy. In 2 books. Book. 1: a textbook and a workshop for academic baccalaureate*. Moscow: Izdatelstvo Yurayt. [in Russian]
13. Khonich, A. (1996). *How to create a three-dimensional game*. Moscow: Firma MIKROART. [in Russian]
14. Tsekhner, M. (2013). *Programming games for Android*. SPb.: Piter. [in Russian]

Information about the author:

Rakovych Volodymyr Anatoliiovych
 rackovychv@gmail.com
 Bohdan Khmelnytsky Melitopol
 State Pedagogical University
 20 Hetmans'ka St., Melitopol,
 Zaporizhia region, 72312, Ukraine

doi: 10.7905/nvmdpu.v0i20.2481