

ВИКОРИСТАННЯ СУЧАСНИХ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ПРИ ВИКЛАДАННІ ПРОГРАМУВАННЯ У ВИЩІЙ ШКОЛІ

Гайтан О.М.

УДК 681.3:378.146

Викладання програмування у вищій школі вимагає застосування сучасних та прогресивних інформаційних технологій. Дана стаття присвячена розгляду основних технологій, які використовуються в навчальному процесі як для навчання навичкам програмування, так і для керуванням навчальним процесом та ефективною комунікації між викладачем та студентом.

Інформаційні технології, які застосовуються при викладанні програмування у вищій школі, можна умовно розділити на такі категорії:

1. Системи управління навчанням.
2. Програми для розвитку навичок з програмування.
3. Середовища розробки.

Розглянемо кожен категорію детально.

Сучасні діти знаходяться в світі цифрових технологій з раннього віку, тому значну роль грає забезпечення ефективною комунікації між викладачем і учнем за допомогою інформаційних технологій. Для управління процесом навчання і комунікації можна використовувати інтегровані системи управління навчанням (LMS), наприклад Moodle або WebCT, або платформи власної розробки, за допомогою яких можна видавати завдання, перевіряти їх виконання і спілкуватися з викладачем. Такі системи поєднують в собі можливість надання навчального матеріалу і контролю успішності, забезпечують повний контроль за навчальним процесом та можливість отримання статистики навчання за заданим критерієм.

Проаналізувавши використання інтегрованих систем управління навчанням та платформ власної розробки, можна виділити їх переваги та недоліки.

Переваги використання інтегрованих платформ:

1. Широкий функціонал. Інтегровані системи, як правило, включають систему управління навчальним курсом, підсистему тестування, що підтримує імпорт завдань та різні формати підготовки тестів, форуми для обговорення з широкими можливостями управління; чати; систему обліку дій користувачів; систему обміну повідомленнями, в тому числі систему підписки і повідомлень тощо.

2. Відносна легкість розробки нових курсів. Додавання у систему нових курсів зі стандартними можливостями не вимагає навичок програмування та ґрунтового знання системи.

3. Широка апробація. Інтегровані системи використовуються тисячами викладачів по всьому світу для забезпечення організованого інтерфейсу електронного та дистанційного навчання.

Недоліки використання інтегрованих платформ:

1. Вартість. MOODLE – програмний комплекс, який поширюється з відкритим вихідним кодом і доступний для модифікації, але ліцензія Web СТ коштує близько 3000 доларів.

2. Необхідність фахівця для установки системи, її налаштування і адміністрування.

Власні платформи розробляються або безпосередньо в даному навчальному закладі або на його замовлення, тому інтерфейс і функціонал такої системи відразу заточений під потреби конкретної освітньої організації. Однак розробка таких систем вимагає значних витрат часових і фінансових ресурсів.

Деякі навчальні заклади використовують для видачі та перевірки завдань електронну пошту або загальнодоступні хмарні сховища Google Диск, однак такі сервіси не мають спеціалізованих навчальних функцій таких, як система управління навчальним курсом або наявність вбудованої підсистеми системи тестування, і використовуються в навчальному процесі лише для передачі інформації та спільного доступу до матеріалів. У цьому випадку викладач веде статистику успішності вручну.

У навчальних системах зазвичай використовуються два основні підходи

до організації контролю знань [2]:

1. Оцінка дій студента застосовується в експертно-навчальних системах, тобто навчальних системах, що ґрунтуються на знаннях. Знання про предметну область і правила оцінки дій студента дозволяють системі визначати рівень знань в ході діалогу. За допомогою методу можна на високому рівні моделювати взаємодію викладача зі студентом.

2. Стандартизований контроль знань полягає в тому, що студенту пропонується вибірка спеціальних завдань і з відповідей на неї виноситься судження про його знання. Для вимірювання здібностей даним методом необхідно проводити періодичні перевірки. Стандартизовані методи контролю знань мають наступні позитивні властивості, що визначають доцільність їх застосування:

- короткочасність перевірки;
- стандартність проведення перевірки та аналізу результатів;
- можливість представлення результатів перевірки в числовій формі та їх математичної обробки.

Тестування дозволяє перевірити знання учнів, в той час як при вивченні програмування основною метою є придбання відповідних умінь. Основними методами придбання умінь з програмування за допомогою інформаційних технологій є використання тренажерів та ігрова форма навчання.

Типовий тренажер для перевірки навичок програмування включає:

- редактор завдань;
- редактор коду;
- систему підказок;
- підсистему перевірки і оцінювання.

За обраною темою надається теоретичний матеріал і генерується набір завдань, що охоплює весь обсяг викладеного матеріалу.

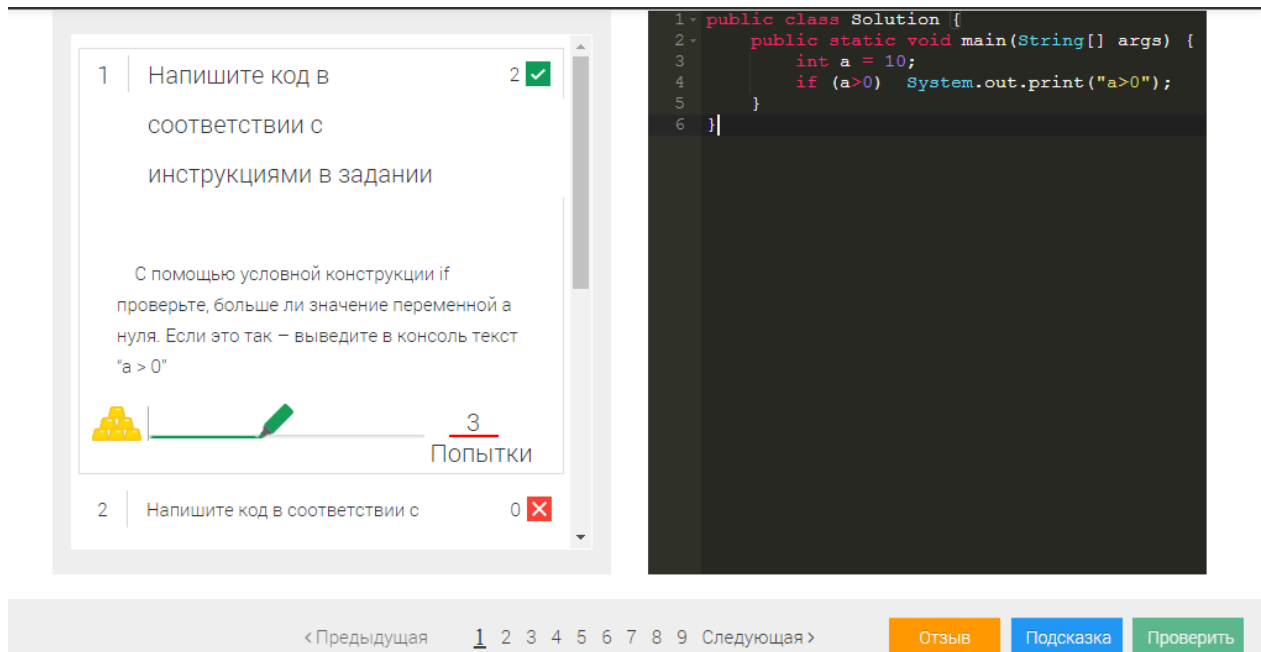


Рисунок 1. Приклад завдання та редактору коду для введення відповіді на курсах програмування ITVDN

Для кожного завдання формується еталонна модель відповіді. Введений в редактор коду текст перетворюється в дерево розбору за допомогою лексичного і синтаксичного аналізу, на підставі чого формується поточна модель відповіді, яка порівнюється з еталонною в підсистемі перевірки і оцінювання. При виявленні помилок система видає відповідну підказку і перенаправляє студента на повторне виконання завдання. Результати заносяться в статистику навчання.

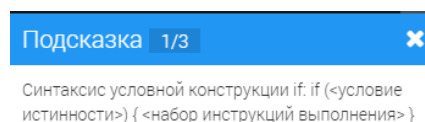


Рисунок 2. Підказка при невірній відповіді на курсах програмування ITVDN

Такий метод дозволяє перевірити практичні навички студентів з програмування, але вимагає створення еталонної моделі для кожної мови програмування окремо та побудову лексичного та синтаксичного трансляторів. Розробка таких механізмів – досить трудомісткий і наукомісткий процес.

Схематично навчальний процес з вивчення окремих розділів курсу можна представити у вигляді DFD-діаграми (рис. 3. – рис. 4).

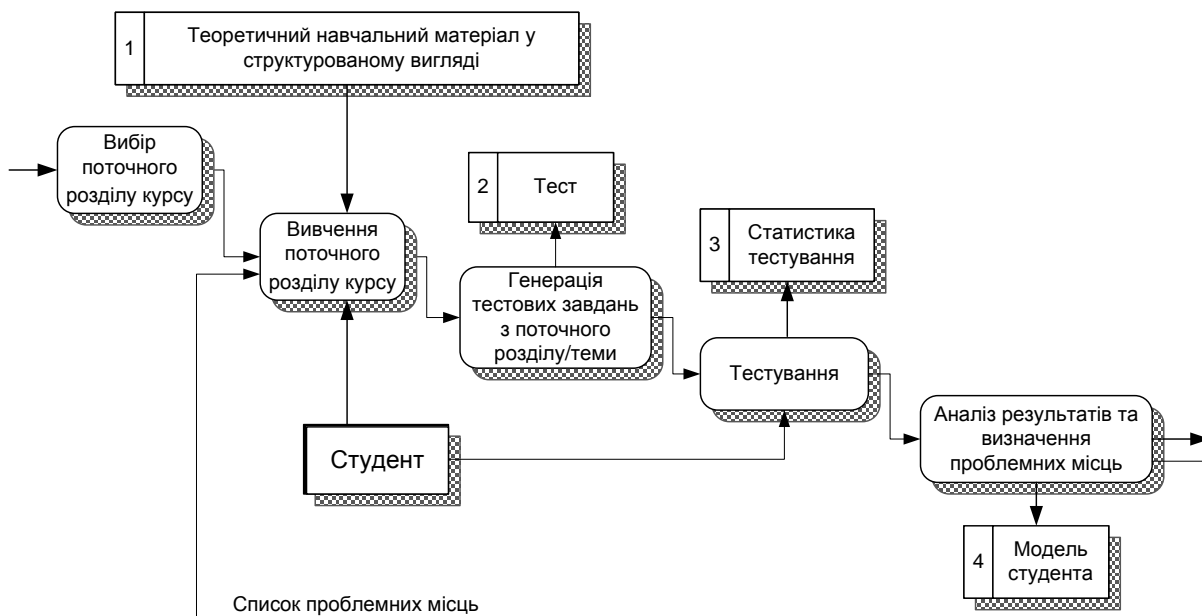


Рисунок 3 Схеми навчального процесу з перевіркою теоретичних знань студентів з програмування за допомогою тестування

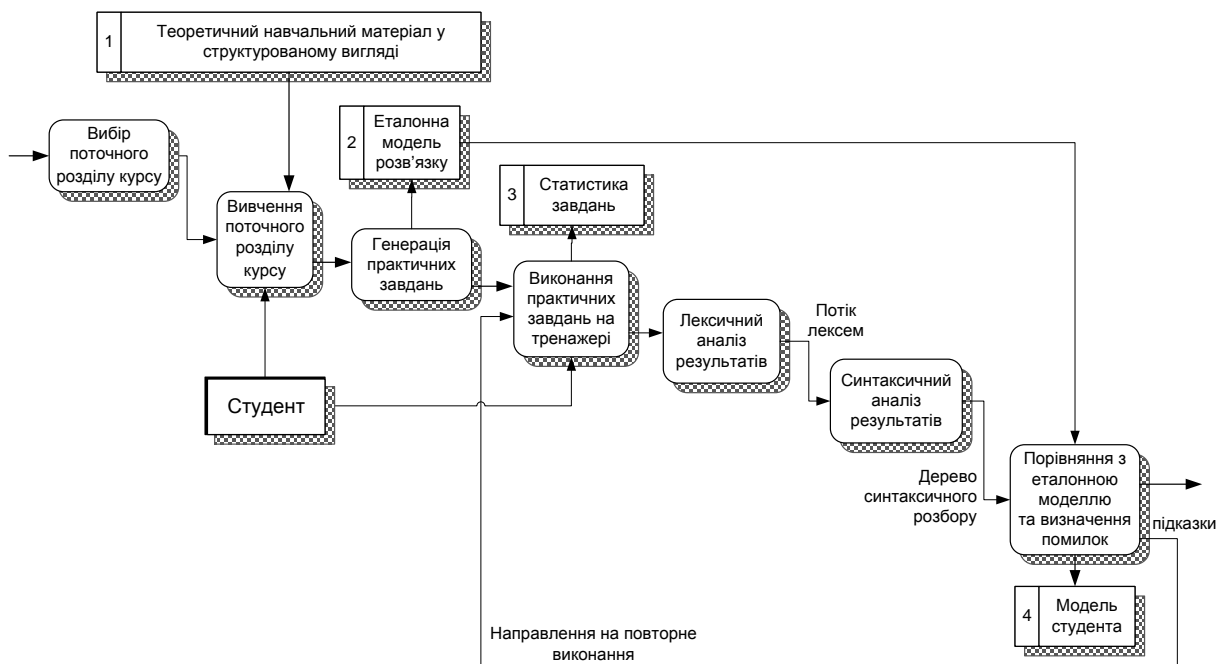


Рисунок 4. Схеми навчального процесу з перевіркою практичних навичок студентів з програмування

Ігрова форма навчання використовує комплексний підхід, в якому поєднується використання теоретичних і практичних навичок, індивідуальності та колективної роботи. При цьому навчання відбувається в більшості на підсвідомому рівні, а засвоєння теоретичного навчального матеріалу, отримання практичних навичок і процес тестування отриманих знань відбуваються практично одночасно в процесі гри. Наприклад, в американських школах було проведено дослідження з вивчення математики за допомогою ігрової комп'ютерної програми ReasoningMind [6]. ReasoningMind – комп'ютерна система, призначена для навчання математики американських школярів. У системі присутній ряд анімованих персонажів, з якими відбуваються різні історії й пригоди. Головним серед цих персонажів є «комп'ютерний джін», якого діти вважають одночасно своїм другом та наставником, що допомагає вивчати математику. Результатом застосування програми в навчальному процесі стало підвищення успішності в експериментальних групах на 16 – 19% за підсумками тестів, розвиток уміння думати в учнів (на думку 100% опитаних вчителів) і суттєве поліпшення ставлення учнів до математики (на думку 94% опитаних вчителів). Крім того, опитування учнів показало, що 75% дітей віддали перевагу використанню ReasoningMind в навчальному процесі у порівнянні зі звичайними уроками.

Використання ігрових сюжетів і анімації в навчальному процесі також застосовується в середовищі Alice (Аліса) [7] – безкоштовного інноваційного програмного середовища, розробленого в університеті Карнегі-Меллона для навчання основам програмування.

Alice – це об'єктно-орієнтована система з відкритим вихідним кодом, розроблена в останнє десятиліття. Вона має інтерфейс перетягування (drag-and-drop), який дозволяє студентам створювати тривимірні середовища і наповнювати їх різноманітними простими в програмуванні об'єктами і персонажами з використанням комп'ютерної анімації та 3D-моделей.

Alice3 включає персонажі, анатомічні рухи та інші художні об'єкти з Sims™ – однієї з найбільш продаваних комп'ютерних відеоігор.

Компоненти анімації Аліси:

Компонент	Визначення	Приклад
Об'єкт	Рухомий чи нерухомий персонаж, що розміщується і / або програмується для переміщення і дії	Тварини, машини, люди, дерева
Сцена	Місце (або «світ» в Алісі), де відбувається історія	Парк, бібліотека, школа
Дії	Вказівки, як повинен діяти кожен об'єкт на сцені	Пройдіть метр, поверніть направо і скажіть «Привіт!»
Взаємодія з користувачем	Способи, якими користувач може маніпулювати об'єктами в анімації	Команди клавіатури або клацання миші для переміщення об'єктів
Специфікації дизайну	Як повинні виглядати об'єкти і декорації в анімації	Розмір, положення, колір

Основні переваги Alice:

1. Високорівнева анімація, що дозволяє користувачам програмувати соціальні взаємодії між персонажами.
2. Підручник на основі розповіді знайомить користувачів з програмуванням через створення сюжету.
3. Галерея 3D-персонажів і декорації з призначеною для користувача анімацією дозволяють «оживляти» ідеї історії.

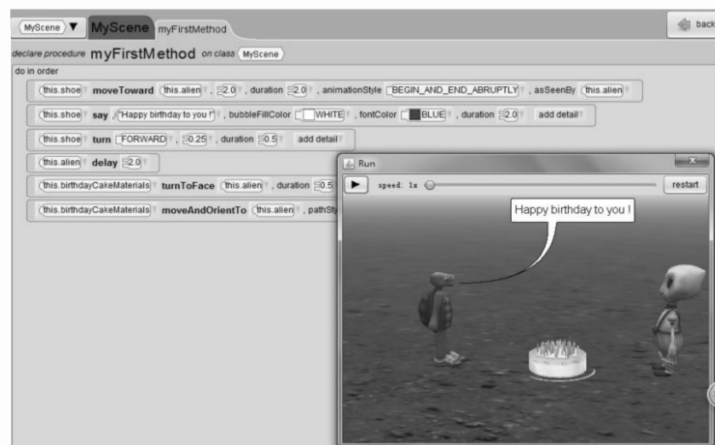


Рисунок 5. Анімація в Alice

Таким чином, за допомогою даного середовища студенти вчаться програмувати, одночасно отримуючи задоволення від створення 3D-анімації, історій і відеоігор.

За результатами досліджень в коледжі Ітаки і університеті Св. Йосипа з

вивчення курсу «Комп'ютерні науки» студентами без досвіду програмування середня успішність покращилася з С до В, засвоєння інформації збільшилася з 47% до 88%. Також підвищився інтерес до програмування без будь-яких змін в основних програмних завданнях. Збільшення часу заняття програмуванням склало 42%.

Близько 15% коледжів та університетів США використовують Алісу для навчання комп'ютерного програмування. Експерти в ІТ-галузі вважають, що його революційний підхід може оживити освіту в області комп'ютерних наук в Сполучених Штатах, починаючи з рівня середньої школи і закінчуючи коледжем.

CodeCombat[8] – одна з найпопулярніших ігор, що навчає програмуванню (Python, основи створення інтерактивних веб-сайтів з використанням HTML, CSS і JavaScript, розробка ігор у стилі CodeCombat). Додаток використовується в освітніх установах для навчання розробці комп'ютерних програм та містить курси з програмування для різних вікових груп.



Рисунок 6.Інтерфейс ігри в CodeCombat

Скорочення аудиторних годин на вивчення навчальних дисциплін, у тому числі програмування, призводить до підвищення ролі самостійної роботи у навчанні. На допомогу студентам при самостійному вивченні програмування приходять сайти для вивчення програмування. На даний момент існує безліч платних та безкоштовних курсів, які присвячені різноманітним розділам програмування, починаючи від програмування і закінчуючи веб-розробкою та

програмуванням ігор. Найкращі безкоштовні ресурси для вивчення програмування, кожний з яких має свої переваги: BitDegree, CodeAcademy, MIT OpenCourseware, FreeCodeCamp, CodeSchool, Codewars, DashGeneralAssembly, CodeConquest, TheOdin Project, Code.org, CodeAvengers, KhanAcademy, edX, WebFundamentalsbyGoogle, Codeeasy.net, Hack.pledge (), aGupieWare, Coursera, Upskill, Udemy.

Для компіляції програми необхідна наявність середовища розробки. Середовище розробки програмного забезпечення – це сукупність програмних засобів, що використовується програмістами для розробки програмного забезпечення. Інтегроване середовище розробки включає в себе редактор тексту, компілятор і / або інтерпретатор, засоби автоматизації збирання і відладчик. До складу комплексу крім перерахованих вище компонент можуть входити засоби управління проектами, система управління версіями, різноманітні інструменти для спрощення розробки інтерфейсу користувача, стандартні заготовки (майстри), що спрощують розробку стандартних завдань, а також браузер класів, інспектор об'єктів і діаграма ієрархії класів.

У навчальному процесі доцільно використовувати середовища розробки, призначені для декількох мов, зокрема Eclipse та Microsoft VisualStudio.

Таким чином, сучасні інформаційні технології в навчальному процесі при викладанні програмування допомагають оптимізувати навчальний процес, переклавши рутинні операції, наприклад по проведенню тестування, з викладача на комп'ютер, та зробити його прозоріше. Використання тренажерів та ігрової форми в процесі навчання дозволяє підвищити мотивацію студентів і зацікавленість у навчальному предметі, оскільки процес переведення складного для учня процесу навчання в ігрове середовище полегшує важку роботу і робить її більш привабливою,.

ЛІТЕРАТУРА

1. Гайтан Е.Н. Функциональные возможности современных систем автоматизации контроля качества обучения. Сравнительный анализ. *Проблеми програмування*. 2015. № 2. С. 101–118.

2. Гайтан О.М. Элементы технологии реализации автоматизованого адаптивного контролю знаний студентов в компьютерных системах навчання. *Радио-електронні і комп'ютерні системи*. 2014. № 4. С. 97–105.

3. Автоматизированные обучающие системы. URL: <http://www.ssl.obninsk.ru/web/002/index.nsf/index/aos> (дата звернення: 15.04.2019).

4. Агеев В. Н. Электронные учебники и автоматизированные обучающие системы. Москва, 2001. 79 с.

5. Проектування і керування процесом контролю знаний на основі онтології предметної області. URL: <http://chito.in.ua/nacionalenij-tehnicnij-universitet-ukrayini-kiyvsekij-polite-v2.html?page=4>(дата звернення: 15.04.2019).

6. ReasoningMind – компьютерная система обучения математике американских школьников. URL: http://mat.1sep.ru/view_article.php?ID=201000609(дата звернення: 15.04.2019).

7. Alice – TellStories. Build Games. Learn to Program. URL: <https://www.alice.org/> (дата звернення: 15.04.2019).

8. Самый захватывающий способ обучения реальному программированию. URL:<https://codecombat.com/>(дата звернення: 15.04.2019).