

до самоосвіти, самоактуалізації, саморозвитку, оволодінню знаннями, уміннями та навичками, розвитку пізнавальної діяльності, навчання впродовж усього життя.

Науково-методична система підготовки інженерів з використанням сучасних ІКТ повинна ґрунтуватись на теоретико-методичних засадах адхократичної ідеї динамічності, адаптивності, гнучкості, творчого підходу, умінні вирішувати проблеми, передбачати результат власних дій, відповідальності за прийняті рішення, умінні працювати у колективі тощо.

Список використаних джерел

1. Ключко В.І., Формування знань майбутніх інженерів з інформаційних технологій розв'язування диференціальних рівнянь: монографія / І. В. Ключко, З. В. Бондаренко. – Вінниця: ВНТУ, 2010. – 216 с.
2. Klochko O. Adaptation of education system of Ukraine in global informatization / O. Klochko // Information Technologies in Education: Scientific Journal. Issue № 1(34). – Kherson: KSU, 2018. – P. 65–78.
3. Ключко О.В. Професійна підготовка майбутніх менеджерів аграрного виробництва засобами сучасних інформаційно-комунікаційних технологій: монографія / О. В. Ключко. – Вінниця : Т. П. Барановська, 2018. – 350 с.

Використання онлайн-курсів у навчанні майбутніх інженерів-програмістів за освітніми програмами скороченого терміну

Крашеніннік Ірина Володимирівна

Мелітопольський державний педагогічний університет імені Богдана Хмельницького

Анотація. Розглядаються окремі аспекти використання сторонніх масових відкритих онлайн-курсів у процесі професійної підготовки майбутніх інженерів-програмістів за освітніми програмами скороченого терміну навчання. Обґрунтовується доцільність їх застосування в освітньому процесі як джерела нової інформації, середовища професійного спілкування з іншими слухачами, засобу організації самостійної роботи.

Ключові слова: майбутній інженер-програміст; МВОК; освітня програма; скорочений термін підготовки.

В Україні університети нині мають право здійснювати підготовку фахівців освітнього ступеня “бакалавр” за освітніми програмами скороченого терміну, тобто студенти можуть здобути вищу освіту і отримати диплом бакалавра за два або три роки, замість нормативних чотирьох. Ця можливість надається вступникам у випадку відповідності критеріям, які сформульовані у правилах прийому на навчання до закладів вищої освіти (ЗВО). Головною вимогою є наявність попередньої освіти за освітнім ступенем “молодшого бакалавра” або за освітньо-кваліфікаційним рівнем “молодшого спеціаліста” здебільшого за спорідненими спеціальностями.

Професійна підготовка в університетах майбутніх інженерів-програмістів за освітніми програмами скороченого терміну дозволяє колишнім випускникам коледжів підвищити рівень власної кваліфікації, отримати новий досвід, задовольнити освітні потреби. Заклади вищої освіти, здійснюючи навчання студентів за такими програмами, виконують замовлення суспільства і ринку праці щодо підготовки працівників високої кваліфікації для галузі інформаційних технологій.

Підготовка майбутніх інженерів-програмістів за освітніми програмами скороченого терміну спирається на вже сформовані у них раніше компетентності. Разом із тим, зміст і ступінь сформованості цих компетентностей може суттєво відрізнятись, що призводить до виникнення проблем, наприклад у процесі колективного виконання навчальних проєктів. Таким чином, перед викладачами фахових дисциплін постає завдання мінімізувати наявну різницю і створити умови для ефективного формування у студентів нових компетентностей.

Одним із шляхів вирішення цього завдання є організація самостійної роботи здобувачів вищої освіти у процесі вивчення дисциплін, передбачених навчальним планом, а також факультативів і сертифікатних курсів. З цією метою викладачі можуть використовувати масові

відкриті онлайн-курси (МВОК), доступні у мережі Інтернет на платформах www.coursera.org, www.edx.org, prometheus.org.ua й ін.

Абревіатура МВОК розшифровується так: Масовий (Massive) – для проведення курсу потрібна велика кількість учасників; Відкритий (Open) – курс є безоплатним, приєднатися до нього можуть усі бажаючі у будь-який момент; Онлайн (Online) – матеріали курсу і результати спільної роботи знаходяться в Інтернеті у відкритому для учасників доступі; Курс (Course) – ресурс має назву, відповідну структуру, правила роботи і загальні цілі, які для кожного учасника можуть трансформуватися [1].

Проблеми створення МВОК, їх використання для надання доступу до освіти широким верствам населення, забезпечення якості такої освіти досліджують вітчизняні (К.Л. Бугайчук, А.С. Кравченко, Г.М. Кузьменко, В.М. Кухаренко, Л.В. Ноздріна, Л.О. Сав'юк, О.П. Войченко й ін.) й іноземні (С.А. Золотухін, D. Cormier, K.G. Dunn, S. Porter, G. Siemens, L.J. Waks й ін.) науковці.

Узагальнюючи досвід досліджень з різних аспектів МВОК, С.А. Золотухін наводить переваги і недоліки їх використання. Деякі переваги: доступність, широке залучення слухачів, реалізація принципів безперервної освіти. Проблеми використання МВОК: відсутність індивідуального навчання, високий відсоток дострокового завершення участі, висока вартість створення, проблема оплати, xMOOC є формою просування корпоративних інтересів й ін. [2, с. 99-100].

Деякі “міфи” про МВОК викриває Т. Bates, зокрема: МВОК збільшують доступ до вищої освіти в країнах, що розвиваються; МВОК використовують нові прогресивні педагогічні підходи; використання великого обсягу статистичних даних (про слухачів) покращує процес навчання; застосування комп'ютерів сприяє індивідуалізації навчання [3].

У процесі професійної підготовки майбутніх інженерів-програмістів за освітніми програмами скороченого терміну навчання МВОК, наявні в Інтернеті, можна використовувати різними способами. Наприклад, якщо частина студентів групи не має початкових знань і умінь з окремих мов програмування, необхідних для виконання більш складних проектів у курсі “Програмування”, їм доцільно запропонувати МВОК для самостійного вивчення. У такому випадку, МВОК застосовується як засіб організації самостійної роботи, але викладач може надавати студентам допомогу і контролювати виконання практичних завдань. Іноді зміст МВОК можна ввести до змісту дисципліни (здебільшого вступних тем) і оцінювати його виконання разом з іншими розділами курсу.

Прикладами МВОК, які можна використовувати у процесі професійної підготовки майбутніх інженерів-програмістів, є розміщені на вітчизняній платформі PROMETHEUS курси “Основи програмування на C#” (https://courses.prometheus.org.ua/courses/Microsoft/CS201/2016_T1/course/) й “Основи програмування на Java” (https://courses.prometheus.org.ua/courses/EPAM/JAVA101/2016_T2/course/). З їх використанням студенти, які вже мають досвід програмування, можуть самостійно ліквідувати недоліки у підготовці. У процесі вивчення дисципліни “Основи ігрового програмування” з метою занурення студентів у світ розробки комп'ютерних ігор доцільно використовувати матеріали серії англomовних МВОК “Video Game Design” (<https://www.edx.org/xseries/video-game-design>), розроблених Rochester Institute of Technology і розміщених на платформі edX.

Основним недоліком використання сторонніх МВОК є те, що їхній зміст не може (з об'єктивних причин) повністю відповідати вимогам вивчення конкретної дисципліни навчального плану університету. Окрім того, їх неможливо повністю інтегрувати в освітній процес ЗВО. З іншого боку, такі МВОК є цінним джерелом інформації, середовищем професійного спілкування з іншими слухачами, засобом організації самостійної роботи.

Отже, МВОК нині не розглядаються як панацея і засіб подолання усіх проблем в освіті, але вони можуть ефективно використовуватися задля підвищення ефективності професійної підготовки.

Список використаних джерел

1. Бугайчук К.Л. Масовий відкритий дистанційний курс: поняття, особливості проведення та перспективи використання в навчальному процесі системи МВС: [Електронний ресурс] //

Інформаційні технології і засоби навчання. – 2011. – № 6 (26). – Режим доступу: <https://journal.iitta.gov.ua/index.php/itlt/article/view/551>.

2. Золотухин С.А. Преимущества и недостатки массовых открытых онлайн-курсов // Дискуссия. – 2015. – № 4 (56). – С. 97-102.

3. Bates T. What's right and what's wrong about Coursera-style MOOCs: [online] // Online Learning and Distance Education Resources. – 05.08.2012 – URL: <https://www.tonybates.ca/2012/08/05/whats-right-and-whats-wrong-about-coursera-style-moocs/>.

Деякі методичні аспекти навчання генетичних алгоритмів

Кузьміна Наталія Миколаївна

кандидат фіз.-мат. наук, професор кафедри теоретичних основ інформатики
Національний педагогічний університет імені М.П. Драгоманова

Анотація. Розглядаються основні поняття генетичних алгоритмів і пропонуються методичні аспекти їх навчання і реалізації у педагогічних університетах.

Ключові слова. Генетичні алгоритми, функція пристосованості, відбір, схрещування, мутація.

Метою і завданнями навчання нормативної дисципліни «Основи теорії і методів оптимізації» студентів-магістрів інформатичних спеціальностей є ознайомлення і оволодіння основними теоретичними принципами та класичними методами оптимізації, набуття практичних навичок доцільного, ефективного і педагогічно виваженого використання сучасних інформаційних технологій при розв'язуванні задач оптимізації в різних галузях науки і техніки. До предмету навчання відносять основні відомості про класичні методи оптимізації функцій однієї та багатьох змінних, огляд основних постановок, методів дослідження і розв'язування задач безумовної та умовної оптимізації, прикладних задач оптимізації; а також сучасні інформаційні системи і технології, що використовують під час розв'язування оптимізаційних задач [5].

Але доцільним також є розгляд у даному курсі таких оптимізаційних задач, для яких класичні методи і алгоритми оптимізації непридатні.

Одним з універсальних методів оптимізації є генетичні алгоритми, за допомогою яких розв'язують широкий спектр оптимізаційних задач [2-4]. *Генетичний алгоритм* – це алгоритм, за допомогою якого знаходять придатні (за певними критеріями, вимогами) розв'язки аналітично нерозв'язних або складно розв'язних проблем, застосовуючи послідовний добір і комбінування невідомих параметрів (величин) з використанням механізмів, що нагадують біологічну еволюцію.

Термін «генетичний алгоритм» і його класичну схему вперше було запропоновано у 1975 году Дж. Холландом у роботі «Adaptation in Natural and Artificial Systems» [1].

Для застосування генетичних алгоритмів задачу оптимізації розглядають як задачу знаходження максимуму деякої невід'ємної на замкненій області визначення функції $f(x_1, x_2, \dots, x_n)$, яку називають *функцією пристосованості* (fitness function), для якої не вимагаються неперервність і диференційованість. Кожний параметр функції пристосованості кодується рядком бітів. $(x_1, x_2, \dots, x_n) \Rightarrow (101, 11001, \dots, 1110)$. Універсальність генетичних алгоритмів полягає в тому, що від постановки конкретної задачі залежать тільки такі параметри, як функція пристосованості і кодування розв'язків. Всі інші кроки алгоритмів для всіх задач виконуються однаково.

Генетичні алгоритми оперують із сукупністю особин (популяцій), які є рядками – кодами можливих розв'язків задачі, тоді як більшість інших алгоритмів оптимізації оперують лише з одним розв'язком, покращуючи його.

За допомогою функції пристосованості серед всіх особин популяції (розв'язків) виокремлюють: найбільш пристосовані, які можуть схрещуватись і давати потомство, та найгірші, які вилучаються з популяції і не дають потомства. Таким чином, пристосованість нового покоління в середньому вище за попереднє.