

УДК 004:371.333+371.026

## **СИСТЕМА ДЕМОНСТРАЦІЇ ПРОГРАМ ТА КОНТРОЛЮ ЗНАНЬ В ІНТЕГРОВАНОМУ СЕРЕДОВИЩІ ВИВЧЕННЯ КУРСУ “ОСНОВИ АЛГОРИТМІЗАЦІЇ ТА ПРОГРАМУВАННЯ”**

**Колеснікова Н.В., Надєєва А.В.**  
**Херсонський державний університет**

*Описано призначення середовища демонстрації програм, використання якого дозволяє приділити більше уваги саме аналізу алгоритмів, та система поточного та підсумкового контролю знань, яка включає як традиційну систему тестів, так і модуль тестування програмного коду, що є специфічним для даної предметної області.*

*Setting of environment of demonstration of the programs is described, the use of which allows to spare more of attention of exactly analysis of algorithms and current and final checking of knowledges system, which includes both the traditional system of tests and module of testing of programmatic code, that is specific for this subject domain.*

За останні роки на Україні активізувалися процеси інформатизації освіти. У зв'язку з цим набули актуальності загальні наукові, методологічні та технологічні проблеми, пов'язані з процесами створення, супроводження та ефективного використання педагогічних програмних засобів (ППЗ) протягом життєвого циклу.

Для створення ППЗ (педагогічного програмного засобу), який би задовольняв усі вимоги до сучасного навчального програмного продукту необхідно не виключати традиційне навчання, а поєднувати його з новими технологічними підходами. Таким чином в сучасному навчанні приділяється важливе значення наступним функціям:

- **Технологічність** передбачає планування навчального процесу з деталізацією всього процесу. Це, по-перше, чітке визначення мети навчання; планування задач, оптимальних методів навчання та навчальних матеріалів (структурно-логічний аналіз матеріалу, виділення ключових понять для засвоєння), підготовка контрольньо-оціночних матеріалів.
- **Організація процесу навчання на основі плану.** Визначення оптимального методу навчання, створення позитивної мотивації студентів, організації і активізації навчально-пізнавальної діяльності на основі самостійної роботи.
- **Організація навчального процесу націлюється на досягнення мети.** Обов'язково в навчальному процесі повинен бути моніторинг знань, який передбачає збір, накопичення, обробку результатів успішності студентів.

Сьогодні змінюється не тільки підхід до навчання, а й технології навчання. Однак зміна технологій, пов'язаних з використанням методів інформатики, з удосконаленням засобів обчислювальної техніки (як технічної, так і програмної компоненти), зробивши революційний вплив на прикладні дисципліни інформаційного циклу, практично не призвела до змін в змістовній частині курсу “Основи алгоритмізації і програмування” (ОАП).

При вивченні основ алгоритмізації у ВНЗ основна увага у першу чергу повинна приділятися:

- виявленню загальних закономірностей і принципів алгоритмізації;
- основним етапам розв'язування задач за допомогою сучасних інформаційних технологій;
- аналізу поставленої задачі, методам формалізації та моделювання реальних процесів та явищ;

- вибору виконавця поставленої задачі, виходячи з того, що він є об'єктом з певними властивостями і набором дій, які потребують аналізу для правильного і ефективного їх використання;
- методам і засобам формалізованого опису дій виконавця, сучасним засобам їх конструювання і реалізації за допомогою комп'ютера;
- аналізу ефективності алгоритмів, що використовуються для розв'язування задач.

Однією з проблем, що постають перед викладачами, методистами, розробниками педагогічного програмного забезпечення курсу “Основи алгоритмізації і програмування” є проблема поєднання доволі консервативної алгоритмічної лінії курсу з більш динамічними та прогресивними лініями виконавця, формалізації та моделювання, інформаційних технологій.

Дійсно, у багатьох ВНЗ, як споконвічно вивчалися питання, пов'язані з даними і способами їхньої обробки, з типізацією даних та використовувалися процедурні мови програмування, так і сьогодні ситуація не змінилася. Ядром цього курсу, як і раніше, є вивчення найпростіших конструкцій, пов'язаних із слідуванням, розгалуженням, циклами, подальшим переходом до вивчення статичних структур даних (масиви, записи, файли, множини), динамічних структур даних, (списки, стеки, черги, дерева і т.п.) та засобів структурування програм (процедури і функції). Для опису алгоритмів “на змістовному рівні” при цьому часто використовується мова блок-схем. Таким чином, протягом останніх 20 років не спостерігалися істотні зміни у змістовній частині курсу ОАП.

Основу нової методики навчання основам алгоритмізації та програмування, що максимально розкриває можливості комп'ютера в організації та управлінні дидактичним процесом, автоматизації етапів процесу навчання, повинні складати навчально-методичні комплекси. Це має бути Web-орієнтоване середовище алгоритмізації, що підтримує структурну методологію алгоритмізації у поєднанні з об'єктно-орієнтованими засобами побудови програмних систем. Інтегроване середовище повинно:

- спиратися на вже існуючу і загальноприйнятту нотацію в записі алгоритмів (наприклад, нотацію Pascal);
- підтримувати процес розробки алгоритмів для проблемно- та предметно-орієнтованих виконавців (абстрактних типів даних);
- надавати в розпорядження програміста всі засоби однієї з розповсюджених реалізацій об'єктно-орієнтованої системи програмування (наприклад, Borland Pascal);
- бути простим у використанні та наочним у роботі;
- бути придатним до використання викладачем – для викладання навчального матеріалу, демонстрацій процесу розробки і виконання алгоритмів, перевірки властивостей алгоритмів та аналізу їх ефективності;
- бути придатним до використання студентами – для практичних занять та лабораторних робіт;
- містити у своєму складі необхідне дидактичне і методичне забезпечення.

Програмний засіб “Інтегроване середовище вивчення курсу “Основи алгоритмізації та програмування” для вищих навчальних закладів” розроблено за об'єктно-орієнтованою технологією проектування і складається з окремих програмних продуктів та модулів.

Одним з таких програмних модулів є середовище демонстрації програм призначене для використання на лекціях, під час проведення практичних завдань і лабораторних робіт для наочної демонстрації роботи алгоритмів. Використання модуля Середовище демонстрації програм дозволяє більше уваги приділити саме аналізу алгоритмів: на різних масивах даних в результаті виконання демонстрації визначаються основні характеристики – кількість порівнянь та кількість перестановок.

Таким чином, завдяки можливостям середовища викладач має змогу урізноманітнити види практичних завдань з алгоритмізації:

- виконати алгоритм з колекції системи або колекції користувача для певних даних;

- скласти алгоритм розв'язування задачі;
- визначити ефективність алгоритму;
- порівняти ефективність алгоритмів для певного набору даних;
- дослідити та змодельовати дані для певного алгоритму (випадковим чином, найкращий та найгірший випадки та ін.);
- узагальнити результати аналізу алгоритмів під час порівняння різних методів розв'язування задачі;
- запропонувати більш ефективний алгоритм розв'язування задачі.

*Функція завантажити алгоритм* – завантажує алгоритм у середовище демонстрації. Завантажений у середовище демонстрації алгоритм можна зберегти у колекцію користувача та виконати його візуалізацію.

*Функція колекція системи* – відкриває колекцію системи, що містить основні алгоритми з курсу ОАП. Кожен алгоритм з колекції системи можна завантажити у середовище демонстрації і виконати його візуалізацію.

*Функція колекція користувача* – відкриває колекцію користувача, що містить алгоритми, створені користувачем інтегрованого середовища. Кожен алгоритм з колекції системи можна завантажити у середовище демонстрації і виконати його візуалізацію.

*Функція сформувати дані* – формує дані для візуалізації виконання алгоритму у середовищі демонстрації одним із способів:

- за зростанням;
- за спаданням;
- випадковим чином;
- введенням даних;
- завантаженням даних.

*Функція виконати алгоритм* – виконує алгоритм у середовищі демонстрації.

*Функція перемикання режиму виконання алгоритму* – перемикає покроковий / безперервний режим виконання та дозволяє відображати візуально обміни та порівняння при виконанні алгоритму в середовищі демонстрації.

*Функція аналізу алгоритму* – виконує підрахунок кількості перестановок та кількості порівнянь при виконанні алгоритму у середовищі демонстрації на певному масиві даних.

Іншим програмним модулем є система поточного та підсумкового контролю знань, що включає як традиційну систему тестів, так і модуль тестування програмного коду, що є специфічним для даної предметної області.

Тести згруповані за темами курсу ОАП. Викладач може редагувати існуючі тести та створювати власні. Кожне питання тесту може бути одного з наступних видів:

- Множинний вибір
- Альтернатива
- Коротка відповідь
- Числове питання
- Співставлення
- Опис
- Випадкове питання
- Випадкове питання на відповідність, що потребує короткої відповіді
- Заповнення прогалін у тексті (Закрита відповідь)

Модуль тестування програмного коду працює наступним чином. Студент обирає задачу з бази та розв'язує її. Потім отриманий алгоритм відправляє на сервер, де відбувається його компіляція. Результати роботи програми порівнюються з результатами зразкового розв'язку. Програма виконується для всіх тестів, які розташовані на сервері для даної задачі. У разі виявлення помилки студент отримує одне з наступних повідомлень:

- Помилка компіляції;
- Run-time error;
- Error time execute.

База задач (БЗ) містить задачі з курсу ОАП, які розв'язані викладачем з метою включення їх до складу алгоритмічних тестів. БЗ структурована за *Розділами* курсу ОАП та *Темами* курсу. Кожна з тем містить перелік задач. Кожна задача має *Номер*, який формується з номера розділу та номера задачі у розділі.

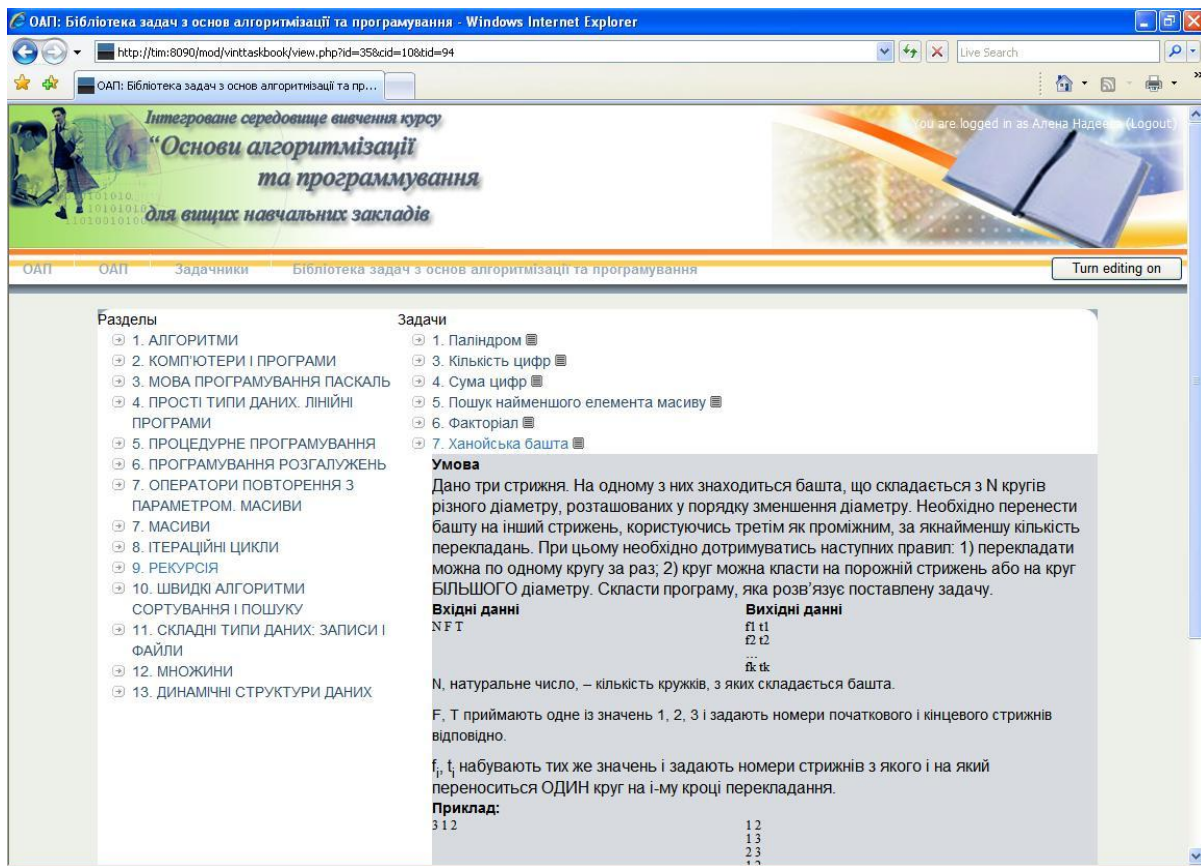


Рис. 1. База задач

Задача з даним номером містить (див. Рис. 1.):

- умову;
- формат введення та виведення даних;
- приклад вхідних та вихідних даних;
- розв'язання;
- тести для перевірки правильності розв'язання задачі студентом.

БЗ може редагуватися за допомогою функцій створення та редагування БЗ. Це функції *Створити*, *Перейменувати*, *Додати*, *Видалити*, *Перемістити*.

Задачі з бібліотеки задач можуть бути розв'язані студентами на практичних, лабораторних заняттях та під час самостійної роботи.

Алгоритм розв'язування задачі з БЗ можна виконати у середовищі демонстрації програм.

Викладач встановлює для кожної задачі опцію *для самостійного розв'язання / для перевірки*.

Головна особливість Web-орієнтованого інтегрованого середовища вивчення курсу "Основи алгоритмізації та програмування" для вищих навчальних закладів полягає у врахуванні специфіки предметної області та у реалізації за єдиною методологією взаємодії всіх електронних засобів навчання: електронного посібника, задачника, середовища демонстрації програм, системи поточного та підсумкового контролю знань, що містить алгоритмічні тести. Основною ж перевагою середовища є можливість організації за його

допомогою самостійної роботи студентів та поточного і підсумкового контролю знань студентів у вищих навчальних закладах.

***ЛІТЕРАТУРА***

1. Співаковський О.В., Львов М.С. Шляхи удосконалення курсу “Основи алгоритмізації та програмування” у педагогічному вузі // Комп’ютер у школі та сім’ї. – 2001. – №4. – С.22-24.
2. Співаковський О.В. Про вплив інформаційних технологій на технології освіти//Комп’ютерно – орієнтовані системи навчання: Зб. наук. робіт. Випуск 4. – К.: НПУ ім. М.П. Драгоманова. – 2001. – С.3-11.
3. Співаковський О.В., Львов М.С., Кравцов Г.М., Крекнін В.А., Гуржій Т.А., Зайцева Т.В., Кушнір Н.О., Кот С.М. Педагогічні технології та педагогічно орієнтовані програмні системи: предметно-орієнтований підхід // Комп’ютер у школі та сім’ї. –2002. – №4(22). – С. 24-28.
4. Співаковський О.В. Концепція викладання дисциплін інформатики в школі і педагогічному вузі // Комп’ютер у школі та сім’ї. – 2003. – №3. – С. 21-25.