

ПОБУДОВА КУРСУ ПРОГРАМУВАННЯ НА ОСНОВІ НАСКРІЗНОГО ПРОЄКТУВАННЯ

Онiщенко Сергiй Миколайович,
здобувач наукового ступеня
спеціальності 014.09 «Середня освіта (Інформатика)»,
Український державний університет імені Михайла Драгоманова, м. Київ, Україна
s.m.onishchenko@npu.edu.ua

Твердохліб Ігор Анатолійович,
доцент кафедри інформаційних технологій і програмування,
кандидат педагогічних наук, доцент
Український державний університет імені Михайла Драгоманова, м. Київ, Україна
i.a.tverdokhlib@gmail.com

При вивченні програмування традиційно використовується діяльнісний підхід до процесу навчання, що базується на концепції “навчання через задачі”. Основними методами навчання вважаються метод доцільно дібраних задач, метод демонстраційних прикладів, метод відкритих програм, метод проєктів [1, с. 24]. Кожен з них має як позитивні сторони так і певні недоліки, що достатньо проаналізовано і описано у працях вітчизняних і зарубіжних науковців, вчених, методистів. Існує багато рекомендацій щодо використання цих методів, і кожний вартий уваги у застосуванні в навчальному процесі.

Структура курсу з програмування за останні роки набула усталеної форми. Зміст і структура навчального матеріалу розробляється і узгоджується з чинними стандартами, навчальними планами. Зазвичай, аудиторні заняття передбачають проведення лекцій і лабораторних робіт.

На лекції подається теоретичний матеріал, що стосується процесу програмування взагалі, опису синтаксичних конструкцій і різноманітних засобів окремої мови програмування, деяких питань технології програмування, тощо. За формою проведення лекції можуть супроводжуватися демонстраціями роботи у обраному середовищі при застосуванні певних засобів мови у розв’язуванні навчальних задач, презентаціями з переліком довідкового матеріалу, демонстраціями використання онлайн-ресурсів.

Основним видом занять для отримання практичних навичок програмування є лабораторні роботи. Структура лабораторної роботи зазвичай передбачає наявність таких компонентів: тема, мета, перелік потрібних основних інструментальних засобів, перелік основних теоретичних положень, визначень, термінів для засвоєння, перелік основних умінь і навичок, що мають бути сформовані у процесі роботи, стислий виклад потрібного теоретичного, довідкового матеріалу з даної теми, приклади розв’язаних задач, список навчальних завдань для виконання, тестові завдання, список інформаційних джерел. Приклади розв’язаних задач можуть бути наведені для різних цілей: як для демонстрації застосування теоретичного матеріалу, так і для зразка оформлення результату виконання лабораторних робіт.

У деяких закладах вищої освіти також передбачаються і практичні заняття, що є важливим допоміжним компонентом навчального процесу, на яких уточнюються і відпрацьовуються теоретичні складові, їхнє практичне використання.

При проведенні лекцій активно використовуються метод демонстраційних прикладів, метод відкритих програм. У процесі обговорення теоретичного

матеріалу важливо одразу вказати на його можливі практичні застосування. Для цього розглядають процес розв'язування ряду задач, на основі яких акцентують увагу на механізмах роботи окремих засобів мови програмування. Можливі варіанти розгляду процесу розв'язування задачі можуть використовуватися для різних навчальних цілей. Аналіз заздалегідь підготовленого розв'язання задачі передбачає виявлення набору доцільно використаних засобів мови, а також і ситуацій їхнього застосування. Самостійне розв'язування задачі викладачем має на меті показати всі етапи створення програмного засобу у середовищі розробки – постановка завдання, аналіз вхідних даних і отриманого результату, абстрагування, побудова моделі розв'язання, знаходження та визначення оптимального алгоритму, добір потрібних структур даних, запис алгоритму засобами мови, виявлення помилок та налагодження програмного засобу, тестування. Демонстрація такої деталізації процесу розв'язання має певні недоліки – значні витрати часу, і можливе відволікання на несуттєві деталі. Адже створення навіть нескладного програмного засобу, навіть якщо програма і не має явної функціональної структури – це досить трудомісткий процес, що передбачає принаймні три компоненти – підготовку середовища, набір тексту і налагодження програми.

Для підвищення ефективності навчання програмування пропонується побудова курсу і його проведення на основі методу наскрізного проектування. За певними ознаками метод наскрізного проектування нагадує метод проєктів, але є суттєві відмінності і у процесі побудови курсу, і у навчальному процесі [2, с. 72].

Основна ідея полягає у навчанні на основі поетапного проектування та створення програмного засобу протягом усього курсу. Такий підхід було застосовано при написанні відомих книг-бестселерів з програмування. Так, у посібнику «Мова асемблера для IBM PC» (1993 р.) автори П. Нортон і Д. Соухе упродовж усієї книги створюють програмний проєкт – оболонку операційної системи Norton Commander, яка була дуже популярною в 90-х роках минулого сторіччя. Аналогом її сьогодні є Total Commander, якою частина користувачів і сьогодні активно користується. Г. Шилдт у посібнику «Java: посібник для початківців» (2019 р.) упродовж всієї книги створює програмний проєкт – довідкову систему Java.

Реалізація методу наскрізного проектування при побудові курсу з програмування передбачає такі дії учасників навчального процесу:

- Підготовчий етап здійснюється викладачем:
 - на основі чинних державних стандартів і навчальних програм формується відповідний зміст навчального матеріалу;
 - проєктуються і описуються навчальні програмні засоби, для створення яких мають бути задіяні всі засоби мови програмування та етапи створення програмного засобу, що відображені у змісті навчального матеріалу;
 - кількість таких навчальних програмних засобів має бути достатньою для того, щоб кожен студент міг працювати самостійно;
 - кожен навчальний програмний засіб має бути змістовним (не формалізованим), практично значимим і цікавим;
 - у відповідності до змісту навчального матеріалу курсу готується цикл лекцій, який передбачає групування теоретичного матеріалу і

демонстрацію його використання при розробці викладачем навчального програмного засобу;

- готуються лабораторні роботи, зміст яких відповідає циклу лекцій; у кожній роботі мають бути достатньо детально сформульовані навчально-методичні вказівки до виконання, на основі яких студентом буде розроблено працюючий компонент майбутнього навчального програмного засобу.
- Кожне лекційне заняття або група таких занять має складатися з двох основних частин:
 - 1) викладач пояснює запланований теоретичний матеріал, що стосується засобів мови програмування, самого процесу програмування, технології та методів програмування, або окреслює межі теоретичного матеріалу і вказує на відповідні доступні інформаційні джерела;
 - 2) викладач демонструє застосування описаного теоретичного матеріалу на прикладі побудови окремого компонента навчального програмного засобу, який створюється протягом усього курсу.

Студенти на таких лекційних заняттях активно пізнають нові засоби мови, а також стають учасниками процесу програмування та створення діючого програмного засобу. Таким чином від лекції до лекції створюваний програмний засіб постійно удосконалюється і набуває нового, потрібного функціоналу. А студенти на власні очі спостерігають за всіма етапами реального процесу створення програмного засобу і доцільність застосування певних засобів мови, методів програмування та дібраних класичних алгоритмів.

- На лабораторних роботах студенти відповідно до розроблених навчально-методичних вказівок застосовують поданий на лекційних заняттях теоретичний і довідковий матеріал для побудови власного проєкту. Результатом виконання кожної такої роботи має бути засвоєний та опрацьований теоретичний матеріал, що може бути перевірено за допомогою підготовлених тестових завдань. Має бути створено працюючий проєкт на даній стадії розробки завдяки доповненому функціоналу на основі отриманих відомостей про нові засоби мови. Також має бути оформлено звіт про доцільне застосування певних засобів мови, за необхідністю визначених методів програмування, класичних алгоритмів, тощо.
- На практичних заняттях (якщо такі заплановано) планується детальне опрацювання лекційного теоретичного і довідкового матеріалу без конкретного втілення у програмні проєкти – всебічний аналіз синтаксичних і семантичних особливостей засобів мови, алгоритмів виконання нових конструкцій, особливостей застосування структур даних, тощо. Можливі поради та консультації викладача щодо планування та реалізації власних проєктів при виконанні студентами конкретної лабораторної роботи.
- По закінченні навчання за розробленим курсом з програмування і викладач і студенти завершують створення навчальних проєктів – працюючих програмних засобів. Отже, можна провести підсумкову зустріч для демонстрації власних розроблених програм і загальне оцінювання роботи кожного розробника за вказаними викладачем критеріями, в якому будуть

брати участь всі розробники проєктів, тобто студенти цього навчального курсу.

Запропонована побудова курсу і його проведення на основі методу наскрізного проєктування має свої переваги та недоліки, які можна коротко характеризувати так.

Позитивним у такій побудові і навчанні курсу є його практична спрямованість, адже відомості про засоби певної мови програмування а також різноманітні приклади програм сьогодні можна знайти і у перекладених або оригінальних зарубіжних посібниках, і на спеціальних онлайн-ресурсах розробників, і на різних онлайн-форумах. Саме присутність студента при створенні програмного проєкту на лекціях, а також розробка власного проєкту під керівництвом викладача дають не лише уявлення про процес програмування і довідковий матеріал, а й можливість отримати знання мови програмування і важливий досвід процесу програмування.

Як недолік такої побудови курсу можна зазначити відносну складність його підготовки. При доборі програмних проєктів на стадії планування і опису технічного завдання потрібне вміння бачити використання запланованого теоретичного матеріалу – відповідних засобів мови, методів програмування – у кінцевому варіанті тих програмних засобів, що плануються як навчальні проєкти для лекційних занять і лабораторних робіт. Також важливим є не лише теоретичні знання з мови програмування, а й практичний досвід викладача у проєктуванні і створенні програмних засобів.

Список використаних джерел:

1. Морзе Н. В. Система методичної підготовки майбутніх вчителів інформатики в педагогічних університетах: автореф. дис. на здобуття наук. ступеня доктора пед. наук: спец. 13.00.02 «Теорія та методика навчання інформатики». Національний педагогічний ун-т ім. М.П. Драгоманова. К., 2003. 40 с.
2. Пузікова А.В., Лупан І.В. Застосування навчальних проєктів у навчанні програмування. Наукові записки. Серія: Проблеми природничо-математичної, технологічної та професійної освіти. №1 (2024). DOI: <https://doi.org/10.32782/cusu-pmtp-2024-1-9>

ДЕЯКІ АСПЕКТИ НАВЧАННЯ ІМІТАЦІЙНОГО МОДЕЛЮВАННЯ

Підгорна Тетяна Володимирівна,

професор кафедри комп'ютерних наук та інформаційних систем,

доктор педагогічних наук, доцент

Державний торговельно-економічний університет, м. Київ, Україна

[*t.pidhorna@knu.edu.ua*](mailto:t.pidhorna@knu.edu.ua)

Самусенко Петро Федорович,

професор кафедри математичного аналізу та теорії ймовірностей,

доктор фізико-математичних наук, доцент

Національний технічний університет України «КПІ імені Ігоря Сікорського», м. Київ, Україна,

[*psamusenko@ukr.net*](mailto:psamusenko@ukr.net)

Розвиток комп'ютерної техніки та її функціоналу значно спростив процес моделювання в різних галузях діяльності людства, в тому числі і складних процесів і систем. Останнім часом досить поширеним стає імітаційне моделювання. Цьому сприяє не тільки розвиток теоретичних основ такого моделювання, а також поява