

■ ВИВЧЕННЯ ОБ'ЄКТНО-ОРІЄНТОВАНОГО ПРОГРАМУВАННЯ В ШКОЛІ

Ярослав Дмитрович Махенько,
студент Житомирського державного
університету імені Івана Франка,
м. Житомир
yarikabv@gmail.com

Яніна Андріївна Стельмашенко,
студентка Житомирського державного
університету імені Івана Франка,
м. Житомир
yanastelmashenko1011@gmail.com

Науковий керівник:
Олександр Миколайович Кривонос,
кандидат педагогічних наук, доцент,
м. Житомир
krypton@zu.edu.ua

Вступ. У зв'язку зі стрімким розвитком новітніх технологій у програмуванні дедалі частіше можна зустріти саме об'єктно-орієнтований підхід до програмування. Тож зрозуміло, чому в шкільному курсі інформатики починають вивчати такі теми, як «Парадигми та технології програмування», після опанування якої учні мають вміння проектувати об'єктно-орієнтовану архітектуру програмних рішень, створювати об'єктно-орієнтовані програмні рішення тощо. Оскільки зазначений курс тільки починають впроваджувати у шкільну програму, то зустрічається він лише в програмах профільних класів, де інформатика вивчається з початкової школи. Тому, хоч і зрозуміло, що навчання, яке передбачає вивчення теми, буде якіснішим, вчителі намагаються оминати окреслений вибіркового блоку через проблемність підбору методів щодо вивчення об'єктно-орієнтованого програмування (ООП) та складність подачі матеріалу. Та враховуючи всі переваги, а саме: розвиток об'єктно-орієнтованого мислення та абстрактного уявлення в учнів, стає зрозуміло, чому ця тема має вивчатись і в класах рівня стандарту.

Виклад основного матеріалу. Термін «об'єктно-орієнтоване програмування» використовується для визначення багатьох речей, ключовий термін тут «об'єкт». Об'єкти — це сутності, які

поєднуються у властивості процедур і даних, оскільки вони виконують обчислення та збереження локального стану. Рівномірне використання предметів контрастує з використанням окремих процедур і даних у звичайному програмуванні. Об'єкт складається з двох елементів: даних і функцій (які називаються методами), що працюють з даними. Наприклад, об'єкти — це рядки, а дані — літери, з яких складається рядок. Це стосується не лише рядків, а й цілих чисел, десяткових чисел та навіть функцій.

Об'єктно-орієнтований підхід до програмування ґрунтується на маніпулюванні об'єктами. Це означає, що розвиток логіки програми досягається шляхом визначення класів різних об'єктів та використання взаємодії об'єктів.

Перші спроби викладання об'єктно-орієнтованого програмування в школі довели, що учні, які до цього вивчали програмування хоча б на базовому рівні, гірше засвоюють ООП через зовсім інший процес мислення, ніж ті, які спочатку зрозуміли основи ООП, а потім вже почали розвивати знання з програмування. Це явище назвали «зміною парадигми», і його зустрічаємо в наукових джерелах дедалі частіше.

Тож стає зрозуміло, що при введенні в шкільну програму, до шкільного курсу інформатики, такої теми, як «Парадигми програмування», потрібно завчасно вводити в курс інформатики вивчення об'єктно-орієнтованого програмування до вивчення основ програмування. Цю думку поділяють певні вчителі та науковці. Так, учені, що досліджували це поняття, під час експериментів довели, що «...взаємодія з об'єктами з самого початку допомогла учням побудувати конкретне розуміння та забезпечила відповідні концептуальні моделі. Ті, хто вивчає об'єктно-орієнтоване програмування, можуть легко зосередитися на поняттях зв'язків класів і об'єктів замість того, щоб зосереджуватися на фактах структурного програмування» [1].

Проте існує й протилежне твердження, що послідовність вивчення не має ніякого значення. Так, у результаті експериментальних досліджень двох учених було оголошено: «Головний результат дослідження говорить, що немає різниці між тим, вивчати спочатку об'єктно-орієнтоване програмування, а потім основи програмування (ООП-first), чи навпаки — спочатку основи програмування, а потім ООП (ООП-later) — це не впливає на результат навчання» [2].

Варто також згадати експериментальні дослідження, що показують відмінності в розумінні програмних текстів об'єктно-орієнтованого стилю. В результаті такого експерименту було виявлено різкий контраст між ментальними репрезентаціями імперативних і об'єктно-орієнтованих програм. У той час як розуміння імперативних програм було загалом кращим, ніж у об'єктно-орієнтованих програм, уявлення імперативних програм були зосереджені на знаннях програмного рівня. З іншого боку, ментальні уявлення об'єктно-орієнтованих програм більше зосереджені на знаннях рівня домену [3].

У результаті стає очевидно, що об'єктно-орієнтоване програмування точно повинно вивчатись в обов'язковому модулі програми, при цьому не має значення, чи передуватиме йому вивчення основ програмування, адже після аналізу проведених досліджень на цю тему стає зрозуміло, що об'єктно-орієнтоване програмування розширює можливості мислення за будь-яких умов його вивчення.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. The Effects of Objects-First and Objects-Late Methods on Achievements of OOP Learners. URL: <https://www.scirp.org/html/23962.html?pagespeed=noscript> (дата звернення: 01.10.2022).

2. Empirical comparison of objects-first and objects-later. URL: https://dl.acm.org/doi/pdf/10.1145/1584322.1584326?casa_token=xL2ji8lM-STUAAAAA:zivCJixx7uP2Q3K3EcyHzMuGBZpjNicBpHBhtvMSrTB4Ys-KOH-0B607TzZbMp1_SyriZWDwYp-D (дата звернення: 01.10.2022).

3. An Empirical Study of Novice Program Comprehension in the Imperative and Object-Oriented Styles. URL: <https://dl.acm.org/doi/pdf/10.1145/266399.266411> (дата звернення: 01.10.2022).