

**ОБ'ЄКТНО-ОРІЄНТОВАНА МЕТОДОЛОГІЯ РОЗРОБКИ У
СТРУКТУРІ ПІДГОТОВКИ МАЙБУТНІХ ПРОГРАМІСТІВ***Конюхов С.Л.**Мелітопольський державний педагогічний університет
імені Богдана Хмельницького, м. Мелітополь
e-mail: sergey_konuhov@mail.ru*

Постановка проблеми у загальному вигляді. Інтенсивний розвиток інформаційно-комунікаційних технологій, їхнє використання у всіх сферах життєдіяльності сучасного суспільства, призвели до зростання популярності професій, пов'язаних з комп'ютерними науками. У загальному переліку видів діяльності, що мають відношення до галузі ІКТ, окремо виділяється програмування, оскільки саме програмісти створюють надзвичайне різноманіття користувацьких додатків для інших галузей. Отже, важливим завданням сучасної вищої освіти у сфері підготовки розробників програмного забезпечення є формування середовища для їхнього ефективного навчання і підвищення кваліфікації.

Аналіз останніх досліджень і публікацій. Теоретичні і практичні аспекти розробки складних програмних комплексів, які мають розглядатися у процесі навчання майбутніх програмістів, докладно висвітлені у роботах Т. Бадда, Г. Буча, М. Вайсфельда, Р. Лафоре, Б. Мейера, А. Пола, Б. Страуструпа. Окремі аспекти підготовки фахівців з інформаційних технологій у вищих навчальних закладах досліджували А. Газейкіна (світоглядні аспекти навчання програмуванню); О. Варфоломеева, О. Гладких, Л. Гришко, Ю. Жураковський, К. Осадча, З. Сейдаметова, А. Стрюк (методичні аспекти навчання майбутніх програмістів); П. Алексєєвський, І. Барков, Д. Лавров, А. Петров, Г. Рудакова, О. Широкова (підходи до вивчення об'єктно-орієнтованої методології); Ю. Біляй, В. Бублик, А. Ричкова, В. Талгатова, М. Умрик (дистанційне навчання програмуванню) та інші науковці.

Виділення невіршених раніше частин загальної проблеми, яким присвячується стаття. Важливим елементом підготовки майбутнього програміста є формування здатності до розробки програм з використанням різних парадигм, мов й інструментальних засобів програмування, а також здатності до самостійного освоєння нових технологій на основі наявних знань і умінь. Отже, мова має йти не про вивчення лише мови C++, Java, C#, Python або, навіть, їхнього комплексу, а про оволодіння методологією проектування, розробки, тестування і супроводу програмних комплексів. На сьогодні вже проведено багато досліджень з проблем навчання програмування у вищих навчальних закладах. Разом із тим, залишається актуальним питання формування у студентів готовності до розв'язання складних практичних завдань і вибору оптимальних засобів для цього, оскільки можливості організації роботи над складними проектами під час навчання дуже обмежені. Відсутність такої готовності призводить до того,

що вони обирають найлегші шляхи розробки, які можуть задовольнити лише у процесі навчання, але зовсім не придатні для подальшої професійної діяльності.

Постановка завдання. Проаналізувати роль вивчення об'єктно-орієнтованої методології розробки програмних продуктів у системі підготовки майбутніх програмістів.

Виклад основного матеріалу дослідження.

Сучасні програмні продукти, призначені для підтримки процесів у промисловості, енергетиці, освіті, зв'язку та інших галузях, у переважній своїй більшості характеризуються високим рівнем складності. Г. Буч називає чотири причини складності сучасних програм: складність предметної області, важкість управління розробкою програмного забезпечення, необхідність забезпечити гнучкість програм, складність опису функціонування дискретних систем [1, с. 38]. Разом із тим, він зазначав, що „Основна задача розробників програмного забезпечення – створити ілюзію простоти, щоб захистити користувачів від величезної і часто довільної складності проекту” [1, с. 40].

Програмісти, у свою чергу, також є користувачами складних програмних систем: з'являються нові інструменти програмування, удосконалюються існуючі середовища проектування і розробки. З одного боку, це дозволяє полегшити роботу програмістів і підвищити її ефективність, з іншого – призводить до того, що у нових покоління створюється ілюзія простоти процесів і зникає потреба у розумінні їхніх рушійних механізмів. Ситуація погіршується також постійним прискоренням бізнес-процесів і прагненням збільшити швидкість випуску нових програмних продуктів або їхніх версій.

Отже, необхідно формувати у студентів розуміння того, що простих програмних систем не існує, існують лише методи, які дозволяють вирішувати проблеми, пов'язані із розробкою складних програм. Розуміння цих методів і готовність їх використовувати дозволить подолати бар'єри входу до професійної діяльності. Саме з цією метою була створена об'єктно-орієнтована технологія, яка на сьогодні є промисловим стандартом розробки програмного забезпечення. Її використання дозволяє з високим ступенем адекватності змодельовати предметну область та її поведінку. Б. Жураковський, О. Варфоломєєва, О. Гладких і О. Хахлюк вказують, що „Об'єктно-орієнтований підхід найбільш природно відповідає реальному процесу розробки систем і не тільки програмних, який є ітеративним і може зажадати внести зміни до вже розроблені і налагоджені компоненти системи.” [3, с. 50].

Наголосимо, що мова йде саме про технологію, а не про конкретні мови і середовища проектування і розробки. Погодимось з М. Вайсфельдом, який зазначає, що освоєння об'єктно-орієнтованих концепцій не обмежується вивченням конкретного метода розробки, мови програмування або

інструментів проектування. Робота в об'єктно-орієнтованому стилі, на його думку, є образом мислення [2, с. 15].

Отже, вивчення об'єктно-орієнтованої методології (ООМ) і формування відповідного стилю мислення є обов'язковим компонентом підготовки майбутніх програмістів у вишах. У процесі проектування методичної системи навчання необхідно визначити обсяг і порядок вивчення ООМ, складовими якої є:

1. Об'єктно-орієнтований аналіз – метод аналізу, у якому вимоги розглядаються з позицій класів і об'єктів, які входять до складу словника предметної області [1, с. 635].

2. Об'єктно-орієнтоване проектування – методологія проектування, яка об'єднує процес об'єктно-орієнтованої декомпозиції і систему позначень для представлення логічної і фізичної, статичної і динамічної моделей систем, які проектуються. Система позначень складається з діаграм класів, об'єктів, компонентів і процесів [1, с. 634].

3. Об'єктно-орієнтоване програмування – методологія програмування, в якій програми представляють собою сукупності взаємодіючих об'єктів, кожний з яких є екземпляром певного класу, який входить до ієрархії класів, пов'язаних відносинами наслідування [1, с. 634].

Навчальний план підготовки бакалаврів за напрямом підготовки 6.040302 „Інформатика” у Мелітопольському державному педагогічному університеті ім. Б. Хмельницького включає дисципліну „Об'єктно-орієнтоване програмування”. Цей курс вивчається на 4 курсі в обсязі 5,25 кредиту ЄКТС. В якості основної мови програмування використовується С++, яку студенти вивчають у курсі „Програмування”.

Наведемо зміст навчальної дисципліни „Об'єктно-орієнтоване проектування”:

1. Змістовий модуль 1. „Класи в об'єктно-орієнтованому програмуванні”.

Основна ідея об'єктно-орієнтованого програмування. Приклади використання ООП. Поля та методи класів. Екземпляри класів. Области видимості класу. Оператор дозволу контексту. Вкладені класи. Static-члени. Показчик this. Функції-члени static і const. Створення об'єкта. Класи з конструкторами. Конструктор за замовчуванням. Конструктор копіювання. Ініціатор конструктора. Класи з деструкторами. Клас string. Члени, що мають тип класу. Використання стандартної бібліотеки класів: математичні та контейнерні класи.

2. Змістовий модуль 2. „Поліморфізм і перезавантаження”.

Поліморфізм. Перетворення, обумовлені класом. Конструктори як перетворення. Перевантаження і вибір функції. Програма overloading. Дружні функції. Дружні класи. Перезавантаження операторів. Перевантаження унарного оператора. Перевантаження бінарного оператора. Перевантаження оператора присвоювання й індексованих операторів. Функція vect::operator=(const vect& v).

3. Змістовий модуль 3. „Спадкування і віртуальні класи”.

Спадкування. Породжені класи. Видимість у похідному класі. Повторне використання коду: границі динамічного масиву. Сумісність типів предка і нащадка. Конструктори у нащадках. Деструктори у нащадках. Віртуальні класи. Віртуальні методи. Поліморфізм. Абстрактні класи. Множинні базові класи. Віртуальні базові класи.

4. Змістовий модуль 4. „Виключення і шаблони”.

Поняття про виключення. Класи виключень. Генерація виключень. Перехоплення виключень. Захищені блоки. Макроси. Заміна ідентифікаторів. Включення файлів. Умовна компіляція. Керування рядком. Поняття про шаблони. Шаблони функцій. Шаблони функцій членів. Шаблони класів. Параметризовані поля. Шаблони і дружні функції. Шаблон масиву. Шаблон Хеш-таблиці. Шаблон черги. Шаблон стеку. Шаблон деку.

Формування практичних умінь здійснюється у процесі виконання лабораторних робіт: Найпростіші класи, Доступність елементів класу, Класи з конструкторами та деструкторами, Поліморфізм, Спадкування, Виключення, Шаблони.

Висновки з даного дослідження і перспективи подальших кроків у даному напрямі. Вивчення об’єктно-орієнтованої методології повинно займати провідне місце у структурі підготовки майбутніх програмістів, оскільки на сьогодні вона є стандартом у галузі розробки програмних продуктів. Необхідно забезпечити достатньо глибоке вивчення всіх компонентів ООМ. З цією метою доцільно ввести до навчального плану напряму підготовки „Інформатика” окремий курс, присвячений об’єктно-орієнтованому аналізу і проектуванню. Окрім цього, посилити увагу до засобів ООП у всіх дисциплінах, пов’язаних із вивченням програмування. Для реалізації практичної підготовки доцільно організувати колективну розробку програмних проєктів під час проходження студентами виробничої практики.

Література

1. Буч Г. Объектно-ориентированный анализ и проектирование с примерами приложений, 3-е изд.: Пер. с англ. / Г. Буч, Роберт А. Максимчук и др. – М.: ООО «И.Д. Вильямс», 2008. – 720 с.
2. Вайсфельд М. Объектно-ориентированное мышление / М. Вайсфельд. – СПб.: Питер, 2014. – 304 с.
3. Жураковський Б.Ю. Об’єктно-орієнтована технологія проектування систем управління / Б.Ю. Жураковський, О.Г. Варфоломеєва, О.В. Гладких, О.А. Хахлюк // Вісник ДУІКТ. – 2013. – №1. – С. 49–53.

Анотація: У статті розглядається значення об’єктно-орієнтованої методології у системі навчання майбутніх програмістів. Наводиться зміст дисципліни „Об’єктно-орієнтоване програмування”. Сформульовані пропозиції щодо удосконалення системи вивчення ООМ в університеті.

Ключові слова: програмування, програміст, складність програмного забезпечення, об’єктно-орієнтований аналіз, об’єктно-орієнтоване проектування, об’єктно-орієнтоване програмування.

Аннотация: В статье рассматривается значение объектно-ориентированной методологии в системе обучения будущих программистов. Приводится содержание дисциплины «Объектно-ориентированное программирование». Сформулированы предложения по совершенствованию системы изучения ООМ в университете.

Ключевые слова: программирование, программист, сложность программного обеспечения, объектно-ориентированный анализ, объектно-ориентированное проектирование, объектно-ориентированное программирование.

Abstract: The article discusses the significance of object-oriented methodology in the system of future programmers education. It is presented the contents of the subject «Object-oriented programming.» It is formulated proposals to improve the system of OOM studying at the University.

Keywords: programming, programmer, software complexity, object-oriented analysis, object-oriented design, object-oriented programming.