

ТЕХНОЛОГІЇ РОЗРОБКИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Л.А. Манжос

ОБЪЕКТНЫЕ ТЕХНОЛОГИИ И ОБЪЕКТНО- ОРИЕНТИРОВАННОЕ ПРОГРАММИРОВАНИЕ

Крымский инженерно-
педагогический
университет

кафедра информационно-
компьютерных технологий

Научный руководитель
Сейдаметова З.С.
(д.п.н., доцент)

Проблемы качественной разработки программных продуктов, представляющие собой эффективные, сильные и связанные системы, решаются использованием объектных технологий [1]. Существуют различные мнения, как лучше представить объекты и концепции объектно-ориентированного программирования. Одни предлагают начинать обучение программированию на первых курсах, используя объектно-ориентированный подход; другие считают, что формировать навыки объектно-ориентированного программирования необходимо на старших курсах. Как известно, концепция объектно-ориентированного программирования базируется на понятиях объектов, классов, экземпляров, абстракции данных, инкапсуляции, наследования, полиморфизме [2].

Таксономия Мосли

Для формирования навыков объектно-ориентированного программирования предлагается представленная на рисунке 1, когнитивная модель изучения объектно-ориентированных технологий, опирающаяся на таксономию Мосли [1]. Модель предполагает в начале обучения объектно-ориентированной технологии формирование двух независимых множеств навыков, которые студенты осваивают одновременно и параллельно. Первое мно-

жество способствует формированию у студентов понимания логики объектно-ориентированного программирования. Второе множество – это навыки объектно-ориентированного проектирования. Эта модель предлагает изучение объектного программирования в трехсеместровой последовательности. Эта цепочка из трех учебных семестров заканчивается выполнением курсовой работы, в которой используются все концепции и навыки, приобретенные в рамках трехсеместровой последовательности изучения.

В двух верхних прямоугольниках, представленных на рис. 1, отражены навыки, имеющие прямое отношение к программированию, развитию навыков решения проблем и способности кодирования алгоритмов. В двух прямоугольниках нижнего ряда рис. 1 представлено множество навыков, имеющих отношение к развитию способности объектного мышления в анализе и проектировании.

Навыки, представленные в крайнем правом прямоугольнике рис. 1, невозможно сформировать без двух независимых множеств навыков, описанных выше. Эти навыки базируются на сумме навыков, представленных в прямоугольниках верхнего и нижнего рядов.



Рис. 9. Таксономія Мослі

Проблеми в обученні ООП

В процесі преподавання об'єктно-орієнтованого програмування (ООП) приходиться стикатися з деякими труднощами, перерахованими нижче:

1. Розуміння базових понять.

У студентів викликають труднощі розуміння базових понять ООП, таких як клас, об'єкт, інтерфейс, абстракція, інкапсуляція, успадкування і поліморфізм.

Для розв'язання цієї проблеми в монографії [3], названій «Процес об'єктно-орієнтованої думки», представлені, в зрозумілій для студентів формі, основи об'єктно-орієнтованих концепцій і пояснено, як можна використовувати різні об'єктні технології в практичному програмуванні. Автор монографії [3] М. Вейсфелд знайомить читача з об'єктно-орієнтованими концепціями, абстракціями, класами (public і private), повторними кодами і середовищами розробки. Ураховуючи сучасне становище і розвиток інформаційних технологій в [3] приділено велику увагу питанням, пов'язаним з побудовою об'єктів, що працюють з XML, базами даних і розподіленими системами (включаючи EJBs, .NET, Web-сервіси і т.п.). Для ілюстрацій і побудови відповідних діаграм в [3] використовується UML, стандартний мову моделювання об'єктів, представлені ілюстрації і приклади кожної концепції.

В статті [4] автори вважають, що для формування розуміння, як виконується об'єктно-орієнтована програма, студентам в якості завдань можна запропонувати намалювати діаграму стану програми в окре-

му моменті часу. Преподаватель проводить інструктаж, а також дає студентам мінімальні вказівки щодо того, що повинні містити ці діаграми, як на них повинні бути зображені центральні концепції і зв'язки, що виникають при виконанні програми. Такий підхід дозволяє студентам легко освоювати ООП-поняття і формує розуміння, як виконуються програми.

2. Вибір мови програмування.

Складно вибрати мову програмування, найбільш підходящу для преподавання ООП. Найчастіше використовується одна з наступних мов програмування – C++, C#, Java, Object Pascal, Python, Objective-C. Можливо, в деяких випадках краще використовувати псевдокод. В статті [5] наведено рейтинг ТЮВЕ мов програмування, а також дані рекомендації щодо того, які з них і чому можна використовувати в навчальному процесі підготовки інженерів-програмістів.

При виборі мови програмування для преподавання ООП необхідно враховувати кроссплатформність, наявність великої кількості вільно розповсюджуваних, а також комерційних середовищ розробки, можливість розв'язання широкого кола завдань. Наприклад, Java використовується для створення настільних, серверних, мобільних програм, JavaScript – при розробці програм в веб-дизайні. Мова програмування Java в усіх цих випадках одна, різниця в тому, що використовуються різні бібліотеки класів. Відзначимо, що в кожній мові програмування є свої особливості і можливості, які не є загальною рисою

разработчиков, а являются осознанной необходимостью, позволяющей решать те или иные задачи проще и эффективнее. Например, в C++ при работе с объектами имеется функционал, которого нет в других языках программирования.

«Learn to program»

Многие авторы полагают, что в компьютерных науках и программной инженерии абстракция является обязательной компетентностью [6], [7]. В англоязычных источниках встречается термин «learn to program» («научиться программировать»), имеющий различные интерпретации. Мы полагаем, что наиболее подходящей интерпретацией, описывающей пять перекрывающихся областей и возможные источники проблем, – проблем, которые обучаемый должен для себя уяснить, если хочет научиться хорошо программировать, – является следующая пятикомпонентная структура: (1) Общая ориентация. Какова общая идея программы, для чего она необходима и что с ней можно делать? (2) Нотационная машина. Абстрактная модель машины для выполнения программы. (3) Обозначения. Синтаксис и семантика используемого языка программирования. (4) Структуры. Абстрактные решения стандартных проблем, структурированное множество подходящих знаний. (5) Прагматика. Навыки планирования, разработки, тестирования, отладки и т.п.

Пять обозначенных выше областей являются «дорожной картой» изучения программирования. Кроме обозначенного выше, для ООП важными являются концепции языка (абстракция типов данных, полиморфизм, события, исключения, абстрактные и логические потоки), проектирование (общие критерии проектирования и выбранные паттерны-проектирования), спецификация и реализация (разделение спе-

цификации и реализации, интерфейсы, пре- и пост- условия, простые инвариантные методики, проекции и итераторы, проектирование иерархии простых классов, абстрактные классы, простые рекурсивные структуры данных, приложения стандартных инфраструктур, в частности, графических интерфейсов пользователя).

Литература

1. Mosley P. A taxonomy for learning object technology / Pauline Mosley // *Journal of Computer Science in Colleges*. – 2005. – V. 20, Issue 3. – P. 204–216.
2. Гома Х. UML. Проектирование систем реального времени, параллельных и распределенных приложений / Хассан Гома. – М.: ДМК Пресс, 2002. – 704, [1] с. – (Серия «Объектно-ориентированные технологии в программировании»).
3. Weisfeld M. Object-Oriented Thought Process / Matt Weisfeld. – Sams Publishing, 2003. – 304 p. – ISBN 0-672-32611-6.
4. Sajaniemi J. A Study of the Development of Students' Visualizations of Program State during an Elementary Object-Oriented Programming Course / Jorma Sajaniemi, Marja Kuittinen, Taina Tikansalo // *The Third International Computing Education Research Workshop (ICER'07)*. – September 15–16, 2007. – Atlanta, Georgia, USA – p. 1–15.
5. Манжос Л.О. Мови програмування в навчанні майбутніх програмістів / Л.О. Манжос, З.С. Сейдаметова // *Науковий часопис НПУ ім. М.П. Драгоманова. Серія №2. Комп'ютерно-орієнтовані системи навчання. Зб. наукових праць / Редрада*. – К.: НПУ ім. Драгоманова, 2010. – № 8 (15). – С. 35–41.
6. Alphonse C., Ventura P. Object orientation in CS1-CS2 by design // *In Proceedings of the 7th Annual Conference on iTCSE*. – New York, NY: ACM Press. 2002. – pp. 70–74.
7. Or-Bach R., Lavy I. Cognitive activities of abstraction in object orientation: an empirical study // *SIGCSE*, # 36, 2. 2004. – pp. 82–86.