

ОПЕРАЦИИ С ОБЪЕКТАМИ И КОММУНИКАЦИЯ МЕЖДУ ОБЪЕКТАМИ В ОБУЧЕНИИ ОБЪЕКТНО ОРИЕНТИРОВАННОМУ ПРОГРАММИРОВАНИЮ

Ивайло Дончев¹, Эмилия Тодорова²

Великотырновский университет имени Св.Св. Кирилла и Мефодия
Болгария, 5000 Велико Тырново, ул. "арх. Г. Козарев" 3, тел. +359 62 649831,
E-mail: ¹i.donchev@abv.bg, ²emilia_todorova@yahoo.co.uk

Abstract

In order to apply successfully the object-oriented approach in programming, it is especially important for students to understand the dynamics of processes within a program and to acquire skills for efficient modeling of the communication and interaction between objects. Forming such skills is an objective both of the Object Oriented Analysis and Design (OOAD) curriculum and of the Object-Oriented Programming (OOP) curriculum.

This paper comments on the weak points of the educational literature on the subject of programming related to the study of operations with objects and their communication; it proposes a system of basic concepts related to this issue and necessary for the OOP course; we suggest a classification of operations with objects consistent with the objectives of the course and we provide some recommendations to the teachers.

Вступление

В процессе исследования учебной литературы замечается неблагоприятная тенденция концентрировать обучение ООП на изучение синтаксических конструкций на конкретно объектно ориентированном языке. Еще в вводных темах подробно комментируются языково зависимые особенности, которые часто не касаются принципов объектно ориентированного подхода, а скорее эффективности программной реализации. Эти особенности безусловно являются важными, но если целью курса является формирование у студентов объектно ориентированного мышления, в первую очередь надо вывести общие для парадигмы концепции и механизмы.

Другой такой тенденцией является акцентирование на данные, описанные в классе и их структуру, а в меньшей степени уделяется внимание на коммуникацию между объектами и операции с ними. Классы и объекты рассматриваются скорее как дополнительная языковая конструкция, подсобляющая процедурное программирование. Для объектно ориентированного подхода важным является декомпозирование системы классов и дефинирование ее динамики в терминах передачи сообщений между совместно работающими объектами этих классов. "Способом фокусирования на передачу сообщений еще в первый день обучения студенты могут достичь до программной модели, в которой отношения и коммуникация между объектами занимают центральное место в проектировании программы, а не алгоритмы" [1].

Основные понятия, связанные с коммуникацией между объектами

В ООП программу надо рассматривать как систему взаимодействующих между собой объектов, которые осуществляют коммуникацию способом сообщений. Объекты реагируют на полученные сообщения, изменяя свое состояние благодаря вызванным методам, что со своей стороны может привести к новым сообщениям, направленные к другим объектам. Именно эта коммуникация по мнению Alan Kay является вторым из шести принципов первого успешного языка для ООП - Smalltalk:

„Объекты коммуницируют между собой способом отправления и получения сообщений” [9, с. 534].

Таким образом, графически, работа одной объектно ориентированной программы выглядит так, как показано на рис. 1. Стрелки указывают направление передачи сообщений:

”Самая главная характеристика класса в том, какие сообщения ему можно отправлять” [4, с. 31]. Вот почему особенно существенной является проблема выяснения смысла понятий метод и сообщение, связь между ними и механизм передачи сообщений.

В книгах по программированию, рассматривающих как процедурные, так и объектно ориентированные аспекты, например [10][5][6][8][16] чаще всего передача сообщения не комментируется. Речь идет только о роли методов (чаще называемые члены-функции) манипулировать данные в объекте. Очень часто в примерах содержится один единственный класс и коммуникация между объектами не обсуждается. Наверное, причина в том, что такие учебники пользуются гибридным языком (чаще всего C++) и особое внимание обращается на алгоритмические и процедурные возможности, причем большое внимание уделяется и на структуры данных, и их эффективную программную реализацию. Кроме того, сам язык приводит к полному использованию преимуществ обеих парадигм.

В книгах, изданных только в целях ООП, но требующих познания процедурных особенностей языка, также не уделяется внимания на передачу сообщений, но идет речь, что ” коммуникация с представителями класса осуществляется только через члены-функции” [15, с. 35] и ”активирование метода на

объект можно сравнить с отправлением сообщением к объекту какие действия он должен исполнить” [15, с. 45].

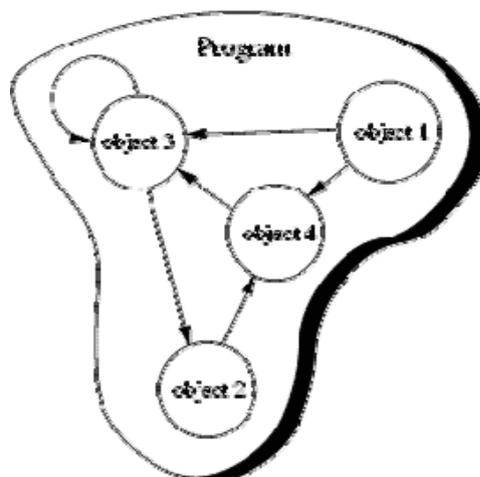


Рис. 1. Программа, состоящая из 4 объектов

В чистых объектно ориентированных языках используется термин сообщение – говорим, что данный объект отправляет сообщение другому, а между тем в гибридных языках как C++ чаще встречаем объяснения того же самого механизма как вызов члена-функции объекта.

Самое большое внимание на механизм передачи сообщений, на связи и коммуникацию между объектами уделено в книгах по ООАП ([12][7][2]). Там ” на самом деле сообщение является операцией которую один объект выполняет над другим, несмотря на разные управляющие механизмы. В наших целях термины операция и сообщение одни и те же” [2, с. 84].

Учитывая особенности учебного курса ООП предлагаем следующие дефиниции основных понятий, связанных с взаимодействием между объектами:

Деф. 1. Метод: член-функция класса. Определяет что могут делать объекты этого класса (дефинирует операцию над объектами класса). Методы каждого класса определяют способ манипулирования данных этого класса, а позже и его функциональность.

Деф. 2. Сообщение: подача заявки к объекту для исполнения метода. Содержит имя метода и стоимости его аргументов. Вызов метода можно рассматривать как реакция на полученное сообщение.

Деф. 3. Операция: „услуга, которую класс может предоставлять своим клиентам” [2, с. 86].

В объектно ориентированных языках операции, исполняемые с объектами, как правило, являются часть дефиниции класса, который является типом объектов (методы). Реализация операций с внешними для класса функциями (обычно дружескими) допустима для некоторых языков (например Object Pascal, C++, CLOS и Ada), но это нарушает требование для капсуляции данных и не рекомендуется, особенно на начальных этапах обучения ООП. В [13] механизм дружеских функций назван „убийцей капсуляций”. Конечно, есть случаи, когда реализация операций с внешней функцией на самом деле необходима для эффективности и элегантности программы. Такими являются операции над объектами разных классов или непростые операции над объектами одного и того же класса.

Таким образом приходим к выводу, что все методы – операции, но не все операции являются методами.

Нужно отметить, что операцию можно осуществить и еще другим способом – статичным методом (так называемым методом класса). Статичные методы ассоциированы в классе и у них нет никакого отношения к его объектам.

Деф. 4. Состояние объекта: текущие значения его членов-данных.

Деф. 5. Поведение объекта: как реагирует на полученные сообщения, как меняется его состояние, какие сообщения он передает.

Деф. 6. Идентичность объекта: это то, что отличает объект от остальных объектов.

Деф. 7. Интерфейс класса: часть класса, которая является ”зримой” для внешней среды. Именно благодаря ей объекты осуществляют коммуникацию с внешней для них средой..

Понятие интерфейс не является в достаточной степени широким, чтобы охватить все взаимодействие объекта с внешней средой. Поэтому Woosh использует понятие ”протокол”, включающее кроме методов и все операции, связанные с объектом.

Деф. 8 Протокол объекта: ”Совокупность всех методов и свободных процедур, относящихся к конкретному объекту, образует протокол этого объекта. Протокол, таким образом, определяет поведение объекта, охватывающее все его статические и динамические аспекты.” [2, с. 88].

Классификация операций

В более старых учебниках по программированию и структурам данных нет классификации основных операций с абстрактными типами данных. В целях обучения ООП такая классификация очень полезна. Главная цель - акцентирование на общем в операциях для разных классов.

Делая анализ предложенных в [11][14][3][16][2] классификаций в большой степени сталкиваемся с часто встречаемой в компьютерной литературе проблемой с терминологией. С одной стороны разные авторы вкладывают разный смысл в терминах (это в большой степени зависит от учебника – процедурного программирования, структур данных, ООП или ООАП), а с другой – их перевод на разные языки тоже не однозначен.

Учитывая цели курса ООП, предлагаем следующую классификацию операций с объектами:

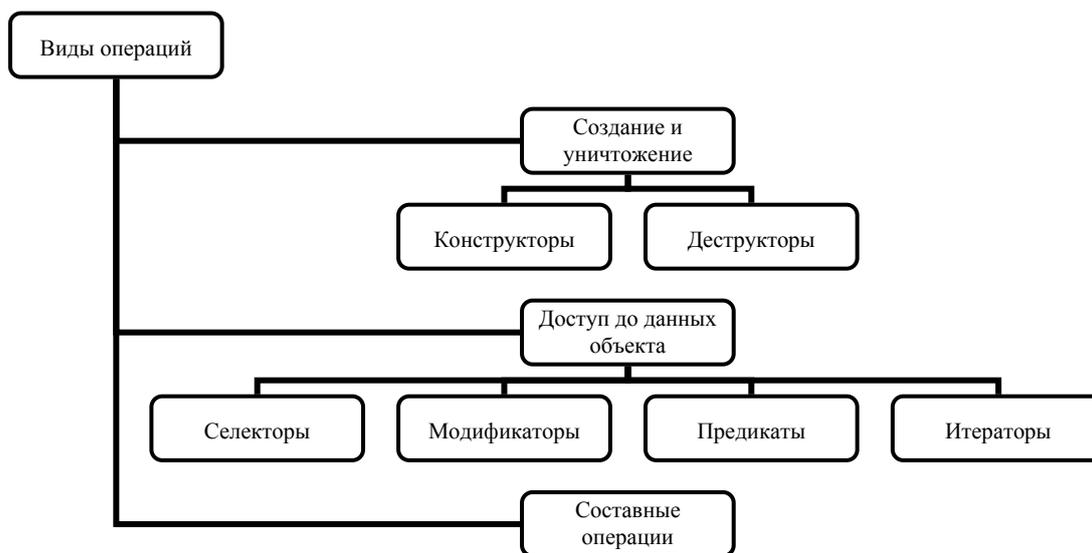


Рис. 2. Классификация видов операций

Конструкторы это операции, которые исполняют для создания объектов, а **деструкторы** - для их уничтожения.

Селекторы служат для получения текущих стоимостей членов-данных объекта, причем этих данных изменить невозможно.

Модификаторы (называемые еще мутаторы) служат для изменения значений членов-данных объекта.

Предикаты можно рассматривать как специальный вид селекторов, обособленных в отдельной категории, потому что, прочитав значения членов-данных, они единственно возвращают булевой результат (true или false).

Использование **итераторов** единственно касается объектов контейнерных классов (реализации списка, очередь, граф, стек и т.д.). Они обеспечивают доступ к данным в четко определенной последовательности.

Составные операции – это комбинация нескольких операций, например, сортировка контейнерного класса или обработка данных класса, причем необходимо использование селекторов, модификаторов и т. д.

Выводы и рекомендации

1. Для правильного усвоения подхода рекомендуем в вводной части учебного курса ООП ограничить реализацию операций способом статических методов и дружеских функций, тем более, что в чистых объектно ориентированных языках (например Smalltalk) такая реализация недопустима.
2. При изучении динамических аспектов объектов надо выяснить механизм, благодаря которому методы получают доступ к объекту, т.е. дополнительный скрытый параметр – указатель к объекту, для которого они вызваны. В разных языках этот указатель обозначен this, self или Me.
3. Для полного усвоения объектно ориентированного подхода необходимо рассматривать объекты с точки зрения их состояния, поведения и идентичности.
4. В учебных курсах ООП трудно находят место примеры, использующие сложные объекты с нетривиальным поведением. В индустрии, однако, часто надобно использование кстади таких объектов, над которыми работает множество программистов. Из-за ограниченного количества

уроков лабораторной работы студентам надо предлагать более сложные примерные проекты в полуготовом виде, к которым они должны добавить функциональность.

5. Для формирования умений работы в команде студентам надо возлагать разработку более объемных курсовых проектов.

Литература

- [1] Bergin J., Wallingford E., Caspersen M., Goldweber M., Kolling M., "Teaching polymorphism early", Proceedings of the 10th annual SIGCSE conference on Innovation and technology in computer science education, Caparica, Portugal, ACM, 2005, Pages: 342 – 343
- [2] Booch Grady, "Object-Oriented Analysis And Design With Applications", Second Edition, Addison-Wesley, 1998, 543 pages
- [3] Deitel H.M., Deitel P.J., "C How to Program", Prentice Hall, 3 edition (August 16, 2000), 1253 pages
- [4] Eckel B., "Thinking in Java", Prentice Hall, 4 edition, 2006, 1150 pages
- [5] Gosling J., Bill Joy, Guy Steele and Gilad Bracha, "The Java™ Language Specification", Prentice Hall PTR, 3 edition (June 24, 2005), 688 pages
- [6] Hekmat Sh., "C++ Essentials", PragSoft, 2005, 311 pages
- [7] Hortsman, C. "Object-Oriented Design and Patterns", Wiley, 2004, 472 pages
- [8] Kaley D., "ANSI/ISO C++ Professional Programmer's Handbook", Que, 1999, 356 pages
- [9] Kay Alan C., "The Early History of Smalltalk", in Bergin T. J, & Gibson R. G., History of Programming Languages - II, ACM Press, 1996, Pages 511-598
- [10] Liberty J., Jones B., "Sams Teach Yourself C++ in 21 Days", (5th Edition), Sams, 2004, 936 pages
- [11] Lippman St., Lajoie J., "C++ Primer", Addison Wesley, 1998, 1264 pages
- [12] Rumbaugh J., Jacobson I., Booch G., "The Unified Modeling Language Reference Manual", Second Edition, Addison Wesley, 2004, 752 pages
- [13] Zhu H., Zhou M., "Methodology First and Language Second: A Way to Teach Object-Oriented Programming", Companion of the 18th annual ACM SIGPLAN conference on OOPSLA, Anaheim, CA, USA, 2003, Pages: 140 – 147
- [14] Азълов П., Златарова Ф., Тодорова М., "Информатика за 10-ти клас Профилирана подготовка", Просвета, София, 2003, 352 стр.
- [15] Богданов Д., "Обектно-ориентирано програмиране със C++", Техника, 1994, 239 стр.
- [16] Тодорова М., "Програмиране на C++, Първа и Втора част", СИЕЛА, София, 2002, 482 стр.