

УДК 004.045

ОРГАНИЗАЦИОННЫЕ ФОРМЫ И МЕТОДЫ ОБУЧЕНИЯ ОБЪЕКТНО-ОРИЕНТИРОВАННОМУ ПРОГРАММИРОВАНИЮ

Ардашева Д.Е., магистрант гр. МИ-211, 2 курс
Научный руководитель: Сулейменов Е.Р. – к.ф.-м.н
Казахский гуманитарно-юридический инновационный университет
г.Семей, Казахстан

В настоящее время одним из важных разделов современных курсов информатики, преподаваемых в ВУЗах, являются разделы, связанные с обучением программированию. Методология объектно - ориентированного программирования становится одной из основных и центральных в подготовке будущих специалистов, существует потребность в ее глубоком изучении, но обучение объектно - ориентированного программирования в ВУЗах началось сравнительно недавно. Студенты испытывают немалые трудности в процессе обучения объектно-ориентированного программирования, такие как изменение стиля мышления с алгоритмического на объектно - ориентированный; изучение объектной декомпозиции на практике; формирование представлений об объектно-ориентированном программировании и др.

Основной проблемой в обучении является недостаточная теоретическая разработанность методики обучения объектно-ориентированного программирования, а также трудности изучения объектно - ориентированного программирования связаны со сложностью процесса разработки программного обеспечения.

Обучению объектно-ориентированному программированию и проектированию посвящены работы зарубежных исследователей, таких как Г.Буч, Б.Мэйер, А.Якобсон, Дж.Рамбо, И.Грэхем, М.Фаулер, а также исследования отечественных ученых Г.С.Ивановой, Т.Н.Ничушкиной, Е.К.Пугачевой и др.

Методология объектно-ориентированного программирования многогранна и нестандартна. Отсутствие единой, общепринятой точки зрения на объектно-ориентированный подход является как ее преимуществом, так и недостатком. Преимущество в том, что в процессе обучения можно рассматривать различные интерпретации объектно-ориентированного программирования. Недостаток в том, что отсутствие единой точки зрения на объектно-ориентированного программирования может привести к ее формальному и ограниченному изучению. Рассмотрение принципов объектно-ориентированного программирования без обучения объектной декомпозиции, объектно-ориентированному проектированию и реализации на практике преимуществ объектно-ориентированного программирования, не

способствует формированию у студентов необходимых представлений об объектно-ориентированном программировании.

Также большую часть затруднений студенты испытывают на начальном этапе обучения, то есть в процессе формирования представлений об основах объектно-ориентированного программирования. От формирований этих представлений зависит дальнейшее изучение методологии объектно-ориентированного программирования. У большего числа студентов сформирован алгоритмический стиль мышления. Смена стиля мышления у студентов обычно происходит тогда, когда они начинают понимать преимущества, которые предоставляет методология объектно-ориентированного программирования. Объектно-ориентированный подход позволяет решать проблемы построения сложных систем; улучшать сопровождение программного обеспечения; расширять и масштабировать программный код; создавать повторно используемый программный код. Эти преимущества являются мотивирующим фактором для изучения студентами методологии объектно-ориентированного программирования. Освоение объектной декомпозиции является одним из решающих факторов, который может привести к изменению стиля мышления студентов со структурного на объектно-ориентированный.

Рассмотрение объектной декомпозиции в начале обучения объектно-ориентированного программирования желательно сократить до минимума и продолжить ее изучение тогда, когда у студентов будет сформировано представление об основах объектно-ориентированного программирования, и они смогут реализовывать объектную декомпозицию на практике.

Осуществление объектной декомпозиции позволит студентам абстрагироваться от программного кода в целом и сконцентрироваться на определенных классах, некоторых отношениях между классами. «В основе любого подхода к программированию лежит понятие декомпозиции (разбиения на части) сложных систем с целью последующей реализации в виде отдельных небольших (до 40 - 50 операторов) подпрограмм».

При осуществлении объектной декомпозиции рекомендуется придерживаться следующих правил по определению классов:

- объект должен быть простым и понятным с точки зрения его структуры;
- объект не должен включать в себя несколько абстракций, поэтому имеет смысл разделить этот объект на несколько объектов;
- объект должен быть "самодостаточным".

Объектная декомпозиция осуществляется до тех пор, пока не будут определены объекты, имеющие четкую структуру данных, поведение, соответствие их решаемой задаче и установленным отношениям между другими объектами. Студенты в процессе изучения объектной декомпозиции учатся самостоятельно осуществлять поиск и "отбраковку" классов, обосновывать выбор классов и отношений между ними. Определение объекта начинается с выяснения того, что это за объект и какую роль он играет в данном случае. Роль объекта определяет его атрибуты и операции. В начале

объектно-ориентированного проектирования не следует фиксировать роль какого-либо объекта, так как это может ограничить выбор ролей для объектов, связанных с данным объектом. Поэтому желательно рассмотреть наибольшее количество возможных ролей, которые могут быть применены к объектам. Роль объекта устанавливается в зависимости от выбираемых ролей для объектов, связанных с ним. Таким образом, роли объектов имеют большое значение для реализации объектной декомпозиции.

Студенты должны учиться применять свои знания в реальных ситуациях; выражать свои идеи на языке UML; расширять сферу возможного применения объектно-ориентированного программирования. Для этого рекомендуется решать сюжетные задачи и задачи, имеющие объекты, прототипами которых являются реально существующие объекты.

В ходе обучения объектно-ориентированному программированию необходимо опираться на методические рекомендации по формированию представлений об объектно-ориентированном программировании, изучению объектной декомпозиции и преимуществ объектно-ориентированного программирования на практике. Студенты должны учиться обосновывать принимаемые решения и находить подходящий вариант построения диаграммы классов UML; решать сюжетные задачи и задачи, имеющие объекты, прототипами которых являются реально существующие объекты. Тогда процесс обучения объектно-ориентированного программирования будет соответствовать современным требованиям.

Список литературы:

1. Бертран М. Объектно-ориентированное конструирование программных систем / Пер. с англ. - М.: Издательско-торговый дом «Русская Редакция», 2005. - 1232 с.
2. Буч Г. Объектно-ориентированный анализ и проектирование с примерами приложений на C++, 2-е изд. / Пер. с англ. - М.: «Издательство Бином», СПб: «Невский диалект», 1998. - 560 с.
3. Иванова Г.С., Ничушкина Т.Н., Пугачев Е.К. Объектно-ориентированное программирование: Учебник для вузов. - 3-е изд., стер. /Под ред. Г.С. Ивановой. - М.: Изд-во МГТУ им. Н.Э. Баумана, 2007. - 368 с.