

УДК 378.147

РАЗВИТИЕ АБСТРАКТНО-ЛОГИЧЕСКОГО МЫШЛЕНИЯ БУДУЩИХ УЧИТЕЛЕЙ ИНФОРМАТИКИ ПРИ ОБУЧЕНИИ ПРОГРАММИРОВАНИЮ НА ОСНОВЕ ИНТЕГРАЦИИ РАЗЛИЧНЫХ ПАРАДИГМ

© З.Р.Халитова

В статье исследуется проблема развития абстрактно-логического мышления будущих учителей информатики в процессе обучения программированию на основе интеграции парадигм программирования.

Ключевые слова: подготовка учителя информатики, развитие мышления, парадигмы программирования.

Программирование как учебная дисциплина занимает одно из центральных мест в системе подготовки учителя информатики. Однако, как показывает опыт, уровень подготовки большинства абитуриентов в этой области является недостаточным. Анализ работ студентов первого курса (профиль информатика) в течение ряда лет показывает, что многие из них не могут самостоятельно реализовать алгоритм решения типовой задачи на языке программирования. Некоторые студенты отмечают, что в школе, изучая основы программирования, они вводили в память компьютера программу, которую составил и объяснил учитель, но самостоятельно программировать не научились. Студенты считают программирование одной из наиболее сложных дисциплин, испытывают серьезные затруднения при его изучении.

Следует отметить, что программирование является специфическим видом человеческой деятельности, для успешной реализации которой необходимо не только применение приобретенных в процессе обучения знаний и умений, но и обладание абстрактно-логическим мышлением.

Мышление формируется и реализуется на определенном содержании [1]. Для развития абстрактно-логического мышления будущих учителей информатики необходимо разработать методическую систему обучения программированию, основанную на интеграции разных парадигм программирования. Парадигма программирования – это комплекс концепций, принципов и абстракций, определяющих фундаментальный стиль программирования. Выделяют следующие парадигмы программирования: процедурную, объектно-ориентированную, функциональную и логическую. В качестве центрального элемента интеграции в настоящей работе предлагается понятие объекта. Объекты могут использоваться для моделирования в различных предметных об-

ластях. В каждой из парадигм программирования введено понятие объекта и реализованы механизмы, позволяющие описать структуру объекта, описать действия с объектами. Объекты могут быть определены на разных уровнях абстракции. В объектно-ориентированном программировании программу можно рассматривать как набор взаимодействующих объектов, в логическом программировании при помощи совокупности утверждений на некотором формальном логическом языке моделируются логические отношения между объектами.

Каждый стиль программирования требует своего подхода к решению задачи. Невозможно признать какой-либо стиль наилучшим во всех областях практического применения. Например, для проектирования баз знаний более пригоден стиль, ориентированный на правила, а для вычислительных задач – процедурно-ориентированный. В настоящее время объектно-ориентированный стиль применяется при разработке широкого круга приложений и включен в другие парадигмы программирования.

Изучение разных парадигм дает возможность проанализировать альтернативные подходы к цели, определить стиль, наиболее соответствующий решаемой задаче и оценить его преимущества.

Основными этапами решения задачи в процедурном программировании являются:

1. Математическое описание, перевод на язык математики ситуаций и задач, построение модели;
2. Выбор метода решения задачи;
3. Построение алгоритма решения задачи;
4. Запись алгоритма на языке программирования;
5. Ввод и синтаксический анализ программы;
6. Тестирование программы и доказательство правильности полученного решения.

Главная задача преподавателя – обучить студента самостоятельно конструировать процесс решения задачи, мыслить творчески.

Выделим три формы организации лабораторных занятий при обучении программированию. При использовании первой формы преподаватель ставит задачу, объясняет метод, алгоритм решения и программу, студенты вводят программу, осуществляют ее синтаксический анализ и тестирование. Они должны научиться исправлять синтаксические ошибки и проверять правильность полученного результата. Преподаватель лишь на первых порах объясняет причины возникших ошибок в программе, далее студенты должны научиться самостоятельно находить и исправлять их, обосновывать результаты тестирования программы. Эта форма позволяет научить студентов находить и исправлять ошибки ввода программы, оценивать полученные результаты, однако при этом студенты не смогут научиться самостоятельно решать задачи, они смогут лишь ввести готовую программу и отладить ее.

Вторая форма предполагает, что преподаватель обсуждает со студентами возможные методы решения задачи, студенты самостоятельно реализуют выбранный метод: составляют программу, отлаживают и тестируют ее. При этом они осуществляют поиск и ликвидацию логических ошибок в программе и – при необходимости – ее модификацию. Достоинством этой формы является более высокий уровень самостоятельности студентов, однако студенты не обучаются самому главному – умению найти решение задачи, они лишь реализуют алгоритм, найденный совместно с преподавателем.

При третьей форме организации лабораторных занятий студенты самостоятельно решают поставленную задачу: строят модель, определяют метод решения, составляют программу и доказывают правильность полученного результата. Преподаватель руководит самостоятельной познавательной деятельностью студента, оказывая ему консультационно-методическую помощь. При успешной реализации именно этой формы удается достичь поставленной цели, однако оценить на практике степень самостоятельности выполненных заданий оказывается затруднительным.

На учебном занятии присутствуют студенты с разным уровнем подготовки по программированию. Для каждой группы студентов следует строить свою траекторию обучения, в процессе которой реализуются: самостоятельный перенос ранее усвоенных знаний и умений в новую ситуацию, видение новой проблемы в знакомой си-

туации, осознание структуры объекта и отношений, поиск способа решения задачи. Такая организация обучения возможна с привлечением всего арсенала средств обучения, в том числе и компьютерных.

При обучении программированию можно условно разбить студентов на три группы. К первой группе отнести студентов, которые испытывают большие затруднения в формализации решения задачи, ко второй – в программировании, к третьей – студентов, справляющихся с предложенными заданиями.

Первая группа студентов требует особого внимания со стороны преподавателя. Преподаватель в процессе беседы со студентом анализирует постановку задачи, уточняет, какие подходы к решению задачи видит сам студент, при необходимости направляет студента.

Вторая группа студентов справляется с формализацией предложенной преподавателем задачи, но при ее программировании затрудняется в использовании конструкций языка, в поиске ошибок, возникающих при ее отладке. Работу с этой группой студентов можно организовать следующим образом:

- преподаватель указывает тему, изучение которой позволит исправить ошибку;
- преподаватель в процессе беседы подводит студента к пониманию причины возникновения ошибки;
- обсуждение материала между студентами, в процессе помощи друг другу они лучше усваивают материал, знания становятся для студента лично значимыми;
- поиск подсказки в справочной системе среды программирования;
- поиск информации в компьютерных средствах обучения.

При работе с третьей группой для преподавателя важно правильно оценить уровень знаний и умений студентов и разработать систему заданий в соответствии с этим уровнем. Студенты при этом самостоятельно выполняют задание, обосновывают правильность полученного результата, преподаватель может обсудить со студентом эффективность его программы и при необходимости рекомендовать реализовать более рациональное решение задачи. Для этой группы студентов можно предложить индивидуальные задания повышенной трудности, включая и творческие задания, работа над которыми продолжится в процессе самостоятельной внеаудиторной работы студента.

Для всех групп студентов преподаватель стремится организовать поисковую деятельность. Практика показывает, что при обучении

объектному программированию число студентов первой группы увеличивается.

В настоящей работе предлагается готовить студента к восприятию понятия объекта уже в процессе обучения процедурному программированию. Объектно-ориентированная программа строится из взаимодействующих объектов, поведение которых определяется методами, представленными в виде процедур. Процедуры являются центральным понятием процедурного программирования. Именно здесь следует рассмотреть разбиение задачи на подзадачи, определение их иерархии, механизмы передачи параметров, структуру и реализацию модулей, в которые в дальнейшем будут помещаться определения классов и объектов при проектировании предметной области, описание и использование типа записи, имеющего поля, аналогичные полям объекта и др.

Объектно-ориентированный стиль в большей степени соответствует обыденному мышлению человека. Моделирование реальных объектов с помощью классов объектно-ориентированного языка программирования нередко более естественно, чем при процедурном программировании, когда обработка данных осуществляется вне самих данных. Системы самой разной природы можно представить с помощью объектов. Объектный подход применяется не только в программировании, но и в проектировании интерфейса пользователя, баз данных и архитектуры компьютеров.

Главными компонентами объектного подхода являются абстрагирование, инкапсуляция, модульность и иерархия [2]. Абстрагирование – это процесс выделения абстракций в предметной области задачи. Инкапсуляция – объединение всех свойств объекта, составляющих его состояние и поведение, в единую абстракцию и ограничение доступа к реализации этих свойств. Модульность – принцип разработки программы, предполагающий реализацию её в виде отдельных частей (модулей). Иерархия – ранжированная или упорядоченная система абстракций.

Можно выделить следующие этапы решения задач по объектно-ориентированной парадигме программирования: постановка задачи, выделение на основе анализа классов объектов (объектно-ориентированный анализ), определение и реализация связей между объектами, эволюция взаимодействий и наследования классов, модификация классов и их распределение по модулям (объектно-ориентированное проектирование), реализация на объектно-ориентированном языке (программирование).

Реализация объединения данных с определёнными видами их обработки делает классы пригодными для описания состояния и поведения моделей реальных объектов. Научить студентов мыслить объектно – сложная задача. Умение формализовать задачу, выделить абстракции данной предметной области, структурировать их и реализовать на языке программирования требует развитого абстрактно-логического мышления. Практика показывает, что лишь небольшая часть студентов справляется с решением задач по объектной методологии. Наибольшие трудности испытывают студенты при выделении абстракций, характеризующих изучаемую предметную область, и их структурировании.

Изучение понятий «класс», «объект», «состояние и поведение объекта», а также объяснение принципов объектно-ориентированного программирования на житейских примерах поможет студенту осуществить самостоятельный перенос ранее усвоенных знаний и применить при объектно-ориентированном анализе и проектировании математических и информационных моделей реальных объектов.

При подборе учебных задач нужно учитывать эволюционное развитие объектно-ориентированной парадигмы программирования, демонстрируя различие стилей мышления при решении одной и той же задачи по процедурной и объектной методологии.

Способность студентов мыслить объектно формируется при разработке визуальных приложений с использованием стандартных объектов (компонент) системы объектно-ориентированного программирования. Даже слабых студентов привлекает возможность создания графического интерфейса разрабатываемого приложения из готовых объектов, входящих в библиотеку визуальных компонентов системы программирования. Студенты создают собственные визуальные приложения, используя стандартные объекты системы программирования, настраивают поведение объектов, изменяют их свойства. Обилие литературы с примерами визуальных приложений, в том числе и электронных источников, позволяет студентам самостоятельно осваивать визуальные возможности среды программирования.

Следует отметить, что обучение объектному программированию только в рамках одного языка программирования (например, в среде Delphi или Lazarus) является недостаточным, нужно рассмотреть объектные возможности других языков программирования (C++, JavaScript). Подобный подход позволит дать более глубокое представление об объектном программировании,

привить навыки его использования и позволит научить мыслить объектно.

Системы обучения, построенные на основе функциональной и логической парадигм, исследованы недостаточно по сравнению с относительно распространенными системами обучения процедурной и объектно-ориентированной парадигмам [3].

Из декларативных языков программирования, на наш взгляд, будущим учителям информатики достаточно изучить язык логического программирования Пролог. Это язык для решения задач искусственного интеллекта, а также любых трудно формализуемых задач, требующих нечислового программирования. Логическое программирование относится к области искусственного интеллекта, оно требует иного стиля мышления, отказа от стереотипов процедурного программирования. Вместо детального описания процесса решения задачи в языке логического программирования описывается сама проблема при помощи совокупности утверждений на некотором формальном логическом языке. Теоретической основой Пролога является раздел символической логики, называемый исчислением предикатов. Логика предикатов была разработана для наиболее простого преобразования принципов логического мышления в записываемую форму. Пролог является языком, основанным на программировании логики, он ориентирован на мышление человека. Пролог особенно хорошо приспособлен для решения задач, в которых фигурируют объекты и отношения между ними [4]. Например, предложение «Саша любит яблоки» в Прологе записывается: `любит («Саша», «яблоки»)`. Объектами являются «Саша» и «яблоки», а «любит» – это отношение, связывающее объекты.

Опыт обучения студентов логическому программированию показывает, что создание программы в терминах решаемой задачи дает возможность лучше понять логику задачи и решить ее. Изучение этой парадигмы позволяет студенту лучше разобраться в своей мыслительной дея-

тельности, установить логические связи между утверждениями с целью получения новых отношений, развить системность мышления.

При обучении будущих учителей информатики программированию разработана и используется система специально подобранных задач, при решении которых студенты постигают особенности проектирования и реализации объектов в различных парадигмах программирования, специфику применения одинаковых методов обработки информации в парадигмах программирования.

Обучение на основе интеграции парадигм программирования позволяет сформировать у студентов умение моделировать изучаемую предметную область разными стилями, проводить анализ и синтез объекта при решении задачи, умение в зависимости от используемой парадигмы программирования переходить с одного уровня обобщения на другой, осуществлять выбор целесообразной парадигмы программирования для решения задачи.

Методическая система обучения программированию, основанная на интеграции разных парадигм программирования, позволяет научить анализу, проектированию и программированию предметной области, развить самостоятельную познавательную деятельность и творческое мышление студентов.

1. *Бим-Бад Б.М.* Умственное воспитание по Лернеру // Историко-педагогический альманах ВЛАДИ. – Владимир, 2010. – № 1(2). – С. 62 – 98.
2. *Буч Г.* Объектно-ориентированный анализ и проектирование с примерами приложений на C++. – М.: Бином; СПб.: Невский диалект, 2000. – 560 с.
3. *Жужжалов В.Е.* Совершенствование содержания обучения программированию на основе интеграции парадигм программирования: дис. ... д-ра педагог. наук. – М., 2004. – 274 с.
4. *Братко И.* Программирование на языке ПРОЛОГ для искусственного интеллекта. – М.: Мир, 1990. – 560 с.

DEVELOPMENT OF ABSTRACT-LOGICAL THINKING IN FUTURE TEACHERS IN THE COURSE OF PROGRAMMING BASED ON THE FUSION OF DIFFERENT PARADIGMS

Z.R.Khalitova

In the article the problem of development of abstract-logical thinking in future teachers in the course of programming on the basis of programming paradigm integration is researched.

Key words: training of information science teacher, development of thinking, programming paradigm.

* * * * *

Халитова Зульфия Равильевна – кандидат педагогических наук, доцент кафедры информатики и вычислительных технологий Института вычислительной математики и информационных технологий Казанского федерального университета.

E-mail: zulaxal@mail.ru

Поступила в редакцию 16.11.2011