

А. И. Газейкина

Екатеринбург

ОБУЧЕНИЕ ПРОГРАММИРОВАНИЮ БУДУЩЕГО УЧИТЕЛЯ ИНФОРМАТИКИ

КЛЮЧЕВЫЕ СЛОВА: объектно-ориентированное программирование; параллельное изучение языков; методы и приемы обучения.

АННОТАЦИЯ. Анализируется опыт и описывается подход к реализации курса программирования в педагогическом вузе. Обосновывается целесообразность параллельного изучения нескольких объектно-ориентированных языков, предлагаются методы и приемы обучения.

A. I. Gazykina

Ekaterinburg

TRAINING FUTURE TEACHER OF COMPUTER SCIENCE IN PROGRAMMING

KEY WORDS: object-oriented programming; parallel study of languages; methods and training techniques.

ABSTRACT. The article analysis the experience in pedagogical high school and describes the approach to the implementation of the course of programming. The author justifies the feasibility of the parallel study of several object-oriented languages, methods and training techniques are offered.

Курс программирования является одним из основных курсов предметной подготовки будущего учителя информатики и информационно-коммуникационных технологий (ИКТ). Поэтому его содержание, формы и методы обучения должны соответствовать современному состоянию языков, методов и технологий программирования и перспективам их развития. Однако перед преподавателем, читающим данный курс, встает ряд проблем, решение которых необходимо для эффективной подготовки будущего учителя информатики.

Специфика педагогического вуза такова, что в число первокурсников, как правило, не попадают вчерашние школьники, имеющие высокий уровень знаний и умений в области программирования. Анализ анкет и результатов входных тестов по программированию студентов 1-го курса Института информатики и информационных технологий Уральского государственного педагогического университета в течение четырех последних лет (это время, когда итоговая аттестация в школе проводилась в форме Единого государственного экзамена и при зачислении абитуриентов учитывался балл, полученный за ЕГЭ по информатике и ИКТ) позволил заключить, что около 40% студентов 1-го курса при обучении в школе не изучали программирование ни в каком его виде.

В качестве входного теста тем первокурсникам, которые занимались программированием в школе, было предложено составить три программы на любом языке

программирования (либо алгоритм в виде словесного описания или блок-схемы): осуществить поиск большего из двух чисел; вычислить сумму входящей числовой последовательности; упорядочить элементы массива по возрастанию/убыванию. В результате выяснилось, что десятая часть выпускников помнит только отдельные ключевые слова и команды языка и не в состоянии составить даже простейшей программы (алгоритма); около 50% могут выполнить лишь первое задание на составление разветвляющейся программы; примерно 30% знают, что такое цикл и справились со вторым заданием, и лишь единицы правильно обрабатывают массив и могут реализовать алгоритм его сортировки.

Примерно четвертая часть первокурсников, изучавших в школе программирование, осваивала язык Basic, остальные обучались программированию на языке Pascal. По результатам опроса, около 80% первокурсников определили свой уровень подготовки по программированию как нулевой или низкий, остальные — как средний и никто не оценил его как высокий.

Таким образом, уровень подготовки первокурсников в области программирования является крайне низким. Это то, что мы имеем «на входе». «На выходе», т. е. по прошествии положенного срока обучения в вузе, мы должны получить выпускника, который в соответствии с требованиями образовательного стандарта [1; 5] овладеет знаниями, умениями и практическим опытом в области объектно-ориентированного, логического и web-программирования. Не рас-

смаатривая двух последних пунктов в этом перечне, останавимся более подробно на обучении будущих учителей информатики «классическому» программированию.

При этом можно обозначить ряд проблем:

- выбор начального языка программирования и языков для последующего обучения;
- отбор содержания обучения;
- выбор методов, форм и средств обучения, в том числе основанных на применении информационно-коммуникационных технологий;
- реализация межпредметных и внутрипредметных связей и т. п.

Традиционно обучение программированию будущего учителя информатики включает в себя освоение студентами сначала структурного программирования (Pascal), а затем объектно-ориентированного (Object Pascal, Java, C++). Как показывает анализ статей и публикаций по этой проблеме, большая часть педагогических вузов придерживается такого же подхода при обучении своих студентов.

В реализуемом подходе не вполне понятным остается назначение первого раздела, т. е. обучение студентов структурному программированию. Основными доводами, приводимыми в качестве обоснования существования этого раздела в вузовском курсе программирования, являются невозможность изучения объектно-ориентированного программирования без опоры на знания и умения в области структурного, а также тот факт, что осуществляется процесс подготовки будущего учителя информатики, который, придя затем в школу, должен будет обучать школьников структурному программированию (и, как правило, на языке Pascal).

Однако еще Э. Дейкстра в своей книге «Дисциплина программирования» отмечал, что «наиболее незаметным свойством любого инструмента является его влияние на формирование привычек людей... Когда этот инструмент — язык программирования, его влияние, независимо от нашего желания, сказывается на нашем способе мышления» [2. С. 4]. Поэтому проблема выбора языков программирования, отбора содержания и выбора адекватных методов и средств обучения в процессе профессиональной подготовки будущего учителя информатики в настоящее время весьма актуальна.

Сегодняшние студенты начнут свою профессиональную деятельность через 3–4 года после изучения курса программирования в вузе. За это время индустрия программирования продвинется вперед в раз-

витии языков, методов и технологий. Поэтому нецелесообразно обучать тому, что уже сегодня является вчерашним днем в области информационных технологий. Структурный подход к разработке программ должен быть заменен объектно-ориентированным [4], являющимся в настоящее время основной парадигмой, используемой в индустрии программирования. Он является основным и при программировании для Интернета (разработка веб-сервисов, создание апплетов, динамического контента).

Таким образом, первым языком программирования для студентов педагогического вуза должен быть объектно-ориентированный язык. В качестве него может выступать Object Pascal (с его реализацией в визуальной среде программирования Delphi) — это облегчит усвоение материала теми студентами, которые изучали структурное программирование на Pascal в школьном курсе информатики. Полновесной заменой Delphi может служить свободная среда разработки Lazarus.

Другой вариант выбора первого языка программирования в вузе — это Java, который является полностью объектно-ориентированным языком, позволяющим разрабатывать платформу-независимые приложения, а также апплеты и приложения для мобильных устройств. Для разработки Java-программ можно использовать интегрированную среду разработки Eclipse (это среда с открытым исходным кодом, для ее использования не требуется покупка лицензии) либо интегрированную визуальную среду NetBeans, которая поставляется вместе с Java 2 SDK Standart Edition последних версий. Третий вариант выбора — язык C++ как наиболее распространенный профессиональный язык программирования для разработки приложений самого широкого назначения.

Представляет интерес подход к обучению программированию студентов, основанный на одновременном (параллельном) изучении и использовании сразу двух языков программирования. В 2010–11 уч. г. такой эксперимент был начат со студентами 1-го курса Института информатики и информационных технологий УрГПУ, он продолжается до настоящего времени. Для обучения программированию студентов 1-го курса были выбраны Object Pascal и Java, на 2-м курсе к этим двум языкам добавился сначала язык C, а затем C++.

- Достоинствами такого подхода являются:
- возможность использования разного инструментария для решения одной и той же задачи;
 - возможность сравнительного анализа

разных языков программирования и формирование способности выбирать оптимальный инструментарий для решения поставленной задачи;

- развитие алгоритмического и логического мышления студентов.

Еще одной проблемой для преподавателя является обоснованный выбор способов обучения (т. е. методов, организационных форм и средств обучения). В настоящее время неприемлемой (и это отмечено в Федеральном государственном образовательном стандарте [5]) является традиционная форма организации обучения «лекции — лабораторные занятия», при которой в ходе лекции рассматриваются возможности языка программирования, а на лабораторных занятиях студенты выполняют задания на составление программ с использованием рассмотренных на лекции возможностей языка программирования. Опыт показал, что при организации изучения теоретического материала в ходе лекционных занятий целесообразно использовать следующие формы и приемы обучения:

- предварительное самостоятельное знакомство студентов с изучаемым материалом;
- коллективный разбор большого количества практических заданий с использованием компьютера и мультимедиа-проектора;
- систематический блиц-опрос (тесты, «диктанты») в начале каждой лекции;
- использование мультимедийных презентаций, поясняющих в наглядной форме основные понятия объектно-ориентированного программирования;
- предоставление студентам после лекции всех ее материалов в электронном виде (основное содержание лекции, презентации, решенные задачи и составленные программы, дополнительные справочные материалы, электронные учебные пособия).

В ходе лабораторных занятий эффективными являются следующие методические приемы:

- выполнение учебных заданий разного типа на:
 - составление программы;
 - поиск и исправление синтаксических и семантических ошибок в программе;
 - определение результатов работы программы (выполнение трассировки);
 - формулировку исходной задачи, которую решает представленная задача;
 - оптимизацию алгоритма и программы;

- составление учебных заданий самими студентами;
- применение не только индивидуальных, но и групповых форм организации учебной деятельности (например, коллективная разработка приложения, когда каждый студент реализует часть алгоритма в виде отдельного модуля (подпрограммы, класса); затем каждый из студентов пишет головную программу: осуществляет «сборку» приложения из разработанных модулей);
- «коллективно-конвейерный» способ решения задач: один студент начинает решение, затем преподаватель прерывает его и приглашает другого продолжить решение и так до тех пор, пока задача не будет полностью решена (способ удобен тем, что повышает концентрацию внимания и обеспечивает эффективную работу со студентами, имеющими затруднения в освоении программирования).

В процессе подготовки лабораторного занятия преподаватель должен построить дидактически полную систему учебных заданий, обеспечивающих формирование необходимых знаний и умений и, если потребуется, их индивидуальную коррекцию. Заданий должно быть достаточно для того, чтобы в случае необходимости предложить студентам разные их наборы. Можно выделить четыре вида учебных заданий по объектно-ориентированному программированию [3]:

- индивидуальные задания, выполнение которых позволит закрепить или проверить знание синтаксиса языка программирования (разработка одного-двух классов, реализация метода, алгоритма или его фрагмента и т. п.), — время выполнения 5—7 минут;
- фронтальные лабораторные работы, позволяющие на практике освоить объектно-ориентированное программирование, — время выполнения 1,5—2 часа;
- индивидуальные и групповые проекты, выполнение которых дает студентам возможность освоить на практике объектную декомпозицию, что практически невозможно сделать в ходе лабораторных работ, — длительность исполнения проекта составляет от одного месяца до целого семестра, примером такого проекта может служить курсовая работа по дисциплине «Программирование»;
- долгосрочные коллективные проекты, выполнение которых организуется не отдельным преподавателем, а несколькими кафедрами университета (по прикладной тематике возможно участие кафедр педагогики, психологии и др.),

основывается на тесном сотрудничестве участников; в ходе выполнения такого проекта студенты получают знания и опыт коллективной разработки программного обеспечения, учатся создавать повторно используемый программный код — длительность работы над таким проектом может составлять от года до двух-трех лет.

При этом важным элементом обучения студентов программированию остается изучение алгоритмизации, основных типовых алгоритмов начиная от простейших и заканчивая, возможно, разбором алгоритмов, использующихся в выполнении заданий олимпиадного уровня (например, динамическое программирование, алгоритмы на графах и т. п.). Это позволяет студентам выполнять задания с использованием любого языка программирования на качественно новом уровне, в том числе успешно решать олимпиадные задачи.

При проведении экспериментальной работы по обучению студентов объектно-ориентированному программированию с одновременным использованием нескольких языков программирования были выявлены следующие проблемные области:

- методология объектно-ориентированного программирования представляется преподавателем (и в учебной литературе, особенно отечественной) весьма поверхностно и, как следствие, у студентов не формируется глубокого понимания принципов и методологии объектно-ориентированного подхода;
- вследствие того что на занятиях преобладают небольшие по объему учебные

задания, студенты воспринимают объектную и алгоритмическую декомпозиции как абсолютно противоположные и взаимоисключающие, формируется представление, что в процессе анализа предметной области задачи нужно выбрать какую-то одну из них;

- недостаточная разработанность содержания учебных заданий и проектов, сложности с подбором большого числа однотипных заданий, а также заданий межпредметного характера, направленных на моделирование различных систем и процессов, в том числе на улучшение сопровождения программного обеспечения и создание повторно используемого программного кода.

Наблюдение за деятельностью студентов при проведении эксперимента, анализ выполненных индивидуальных заданий, курсовых работ, результатов участия в олимпиадах и конкурсах различных уровней привели нас к выводу о возможности и целесообразности организации обучения программированию будущего учителя информатики на основе объектно-ориентированного подхода с использованием нескольких языков программирования. При этом формируется предметная составляющая профессиональной подготовки будущего учителя, развивается абстрактное, логическое и алгоритмическое мышление, опосредованно оказывается влияние и на его методическую компетентность, так как содержание и методы обучения в вузе переносятся затем начинающим учителем в школу, в практику его профессиональной деятельности.

ЛИТЕРАТУРА

1. ГОСУДАРСТВЕННЫЙ стандарт высшего профессионального образования. Специальность 050202 Информатика. URL: <http://www.edu.ru>.
2. ДЕЙКСТРА Э. Дисциплина программирования. М. : Мир, 1978.
3. ПЕТРОВ А. Н. Основные подходы к обучению студентов объектно-ориентированному программированию и проектированию // *Фундаментальные исследования*. 2008. № 4. URL: www.rae.ru/fs/?section=content&op=show_article&article_id=7780826 (дата обращения: 26. 09. 2012).
4. СТАРИЧЕНКО Б. Е., ШЕМЕТОВА А. Д. Совершенствование информационно-технологической подготовки студентов на основе системно-объектного подхода // *Образование и наука. Известия УрО РАО*. 2009. № 4 (61).
5. ФЕДЕРАЛЬНЫЙ государственный стандарт высшего профессионального образования по направлению подготовки 050100 Педагогическое образование. URL: <http://минобрнауки.рф/документы/924>.

Статью рекомендует д-р пед. наук, проф. Б. Е. Стариченко