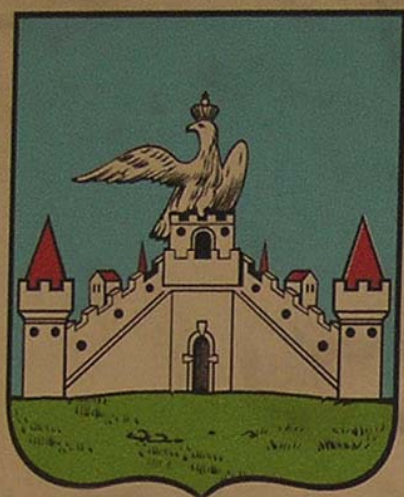


Федеральное агентство по образованию
Российской Федерации



Орел, 2006

СОВРЕМЕННЫЕ МЕТОДЫ ФИЗИКО-МАТЕМАТИЧЕСКИХ НАУК

Труды международной
конференции
9 – 14 октября, Россия

Том 3

- Методика преподавания математики
- Методика преподавания физики
- Методика преподавания информатики

Орловский государственный университет

АДА – ИДЕАЛЬНЫЙ БАЗОВЫЙ ЯЗЫК ОБРАЗОВАНИЯ В ОБЛАСТИ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ

В.О. Мищенко¹, С.И. Рыбин²

¹Харьковский национальный университет им. В.Н. Каразина, Харьков

²Московский государственный университет им. М.В. Ломоносова, Москва

Целью работы являлось – показать адекватность и выявить преимущества использования Ады в качестве базового языка университетского обучения в сфере информационных технологий. Использован опыт авторов и даны методические указания на важные аспекты обучения с применением Ады.

Суть проблемы. Под базовым языком понимаем первый язык обучения систематическому программированию по определённой специальности в данном университете с учётом того, что в последующих курсах этому языку будет отдаваться преимущество во всех случаях, когда использование других языков программирования не диктуется самим содержанием данного учебного курса. Например, в дисциплине «Системное программирование» использование языка Си наряду с языками ассемблеров является естественным (наряду, возможно, с другими) независимо от базового языка обучения. Аналогично в курсе «Параллельное программирование» использование языка Ада (наряду, возможно, с другими) является правильным выбором в любом случае. Но, скажем, в практикуме по вычислительным методам, в дисциплинах, посвящённых информационным системам, методам проектирования, разработки и тестирования программного обеспечения предпочтительно максимально широкое использование именно базового языка образования по данной учебной специальности.

В пост-советском вузовском образовании в области информационных технологий выбор базового языка, с одной стороны, диктуется типовыми программами специальностей, а с другой - определяется множеством разнообразных субъективных причин.

По настоящему важными, однако, являются следующие соображения: насколько данный язык облегчает профессиональное изложение учебного материала и насколько он облегчает усвоение студентами совокупности учебных дисциплин. Среди этих причин – привычки и квалификация преподавателей, доступность литературы, изданной на бумажных носителях и на русском языке, личные предпочтения тех, кто разрабатывает (или утверждает!) учебно-квалификационные программы специальностей и рабочие программы конкретных курсов, представления о том, что в данный момент наиболее востребовано рынком труда. Сплошь и рядом мы имеем совершенно ненормальные ситуации, когда основной (и нередко математизированный) курс по алгоритмам и структурам данных поддерживается ... языком Си, курс по параллельному программированию и системам реального времени основывается на Джаве (которая не для этих областей проектировалась и создавалась), а курс по методам разработки и сопровождения больших программных систем использует в качестве языка сопровождающего практикума, извините, Паскаль-Дельфи. В результате существенная часть времени и сил тратится на попытки разглядеть алгоритм (а еще того сложнее – структуру данных!) через криптографический синтаксис Си или, скажем, на то, чтобы отделить проблемы адресной арифметики от логических и математических проблем организации сортировки и поиска. Или (в курсах по

параллельному программированию) возиться с тем, чтобы правильно переключать семафоры и вовремя посылать сигналы (от чего всеми силами стремится уйти реальная программная индустрия!) вместо того, чтобы выделять процессы и разрабатывать модели их взаимодействия высокоуровневыми средствами, допускающими эффективный статический и динамический контроль. Еще конфузливей «освоение методов» проектирования и разработки больших проектов и систем средствами Дельфи, предназначенной только (и в этом качестве действительно широко и эффективно используемой) для быстрого прототипирования и создания графических интерфейсов конкретно под Виндоуз. При этом такими реальностями больших проектов, как полиплатформенные стандарты, Юникс-ориентированные инструменты, лицензионная чистота, разумеется, даже не пахнет.

В то же время уникальные учебные возможности языка Ада и основанных на нем GNAT-технологий (в ассортименте бесплатно предоставляемых университетам по удобной академической лицензии с фирменной технической поддержкой) остаются в тени былых заблуждений и глубочайшего застоя пост советской программной индустрии. Последнее, однако, не продлится вечно. Уже сейчас рынок эксклюзивного программного обеспечения встроенных систем (который потенциально неограничен и может ориентироваться только на отечественных разработчиков) испытывает значительное оживление. Это означает, что в ближайшие годы Ада возникнет, по крайней мере, в учебных курсах программирования систем РВ и разработки больших программных проектов. Важнейшие аспекты более сложной, но и более перспективной возможности, - избрание Ады в качестве базового языка университетского обучения по некоторым специальностям – обсуждаются в данной статье.

Краткая справка об обсуждаемом языке. Язык Ада был разработан в конце 70-х годов прошлого века по заказу Министерства обороны США. Язык возник в результате проекта, нацеленного на создание единого языка военных встроенных систем. В 1983 году язык приобрел статус американского национального стандарта, а в 1986 – стандарта ISO. В течение ряда лет МО США при помощи специальных административных мер, во-первых, обеспечило применение исключительно Ады для всех новых проектов, разрабатываемых по его заказам, а во-вторых, организовало систему контроля реализаций на соответствие стандарту, в результате которой реализации каких-либо диалектов, расширений или подмножеств языка практически не подучили распространения. Этим была заложена основа уникальнейшей абсолютной устойчивости языка Ада в его развитии на протяжении десятилетий. Именно это необходимо для истинно больших (то есть, разрабатываемых десятилетиями) программных проектов и в той же мере – для нормального учебного применения.

На днях один коллега, похлопывая рукой по стопке рыхлых манускриптов с изображением третьей буквы алфавита и знаков «плюс», изрёк: «У всех этих многочисленных ... достойных и обстоятельных книг имеется ровно одна общая черта. Когда Вы дословно наберёте в используемой Вами системе поддержки программирования какой-нибудь оттуда пример, то он ни за что сразу работать не будет!». Так вот, если Вы получите по Gnat Academic Program последнюю версию Ада-компилятора и “подсунете” ему Ада программу, которая была успешно отлажена в середине 80-х годов прошлого века под сертифицированный компилятор стандарта языка того времени (Ада-83), то, господа преподаватели, поручимся, что эта программа в 2006 г. откомпилируется и заработает!

Несмотря на то, что Ада была создана по заказу МО США для использования в разработках, прежде всего, военных встроенных систем, какие-либо причины, препятствующие использованию языка для невоенных целей, а также вне области встроенных систем, отсутствовали еще тогда.

В ISO принят десятилетний цикл пересмотра информационных стандартов. Стандарт Ады был первый раз пересмотрен в 1995 году. При этом был взят официальный прицел на свободное коммерческое использование Ады любыми субъектами и для раз-

нообразных целей (в частности, предусматривались вычислительные применения, применения в информационных системах, системах распределённых вычислений и РВ, для особо надёжных и безопасных систем, для организации компилируемых программ, использующих модули, разработанные на различных языках программирования). Сейчас завершено второе обновление стандарта. О реальной совместимости прежних программ с новыми стандартами смотри абзацем выше.

С технической точки зрения все современные индустриальные языка предоставляют равноценные наборы выразительных средств, различаясь не в том, что они позволяют делать, а в том, как именно та или иная задача решается средствами того или иного языка. Ада отличается тем, что является на настоящее время единственным языком программирования, предоставляющим высокоуровневые средства программирования асинхронных процессов. Кроме того, модель модульности, предлагаемая Адой, существенно превосходит основанную на текстуальных вставках и препроцессоре «модульность» в С/С++ и их клонах (Джава, С#).

Адекватность и преимущества в базовых курсах программирования. Ада – прямая наследница виртовского Паскаля. Последний в своем развитии заимствовал у Ады в упрощенном виде то, что в нем теперь зовётся «модулем» (юнитом), а в Аде реализовано как «пакет». Код небольших (и не параллельных) программ на Аде без особенных пояснений понимается программирующими на Паскале. В классических университетах, где первым языком обучения был Паскаль, обучить добросовестного студента программированию на Аде занимает от нескольких часов до нескольких дней (судя по нашему личному опыту, который оказался общим и в МГУ, и в ХНУ). В то же время Ада много мощнее Паскаля по предоставляемым средствам программирования, а в рамках подмножества Ады, эквивалентного Паскалю, сняты неестественные по логике ограничения, скажем на тип возвращаемого функцией результата, и вводятся такие удобства, как «расширение» известной конструкции if-then-else до if-then-elseif-then...-else-end if;

Учитывая тот факт, что подавляющее большинство кафедр, ответственных за общие курсы программирования, имеют богатый опыт и полное методическое обеспечение для преподавания с использованием языка Паскаль, переориентация курсов «Основы программирования» на использование языка Ада не составит труда для преподавателей. Что же касается студентов, то в ключевом вопросе обеспечения их учебной литературой на родном языке, то в последние годы произошел перелом. Появился полный электронный учебник по языку Ада, написанный львовским автором на русском языке (и свободно предоставляемый) Автор - А.Е. Гавва из Нацбанка Украины, программист с богатым опытом работы в области встроенных систем. В Киеве издан ряд учебников и пособий на русском и украинском языках. Их автор – доцент Киевской политехники А.В. Корочкин, известный специалист в области параллельного программирования.

Базовые курсы по информационным технологиям (основы программирования, алгоритмы и структуры данных) в значительной степени определяют дальнейшую судьбу студента как профессионала в области информационных технологий. Сверхбыстрая изменчивость инструментальных и операционных сред приводит к тому, что обучать конкретным деталям процесса кодирования становится бессмысленным. Наиболее эффективным является подход, нацеленный на освоение фундаментальных основ профессии программиста, которые неизменны уже много десятков лет – умению правильно спроектировать программу, выбрать адекватные структуры данных, определить на их основе абстракции, удачно отражающие решаемую задачу, выбрать и ясно записать алгоритм решения. И опытные преподаватели (как и потребители молодых специалистов) знают – вот это-то в обучении и сложнее всего! С этой точки зрения лучшим языком программирования, используемым для примеров и учебных заданий, является тот язык, который в наименьшей степени отвлекает на себя внимание, позволяя сконцентрироваться на том, что является целью курса.

В этом контексте обратим внимание на следующие аспекты.

Ада предоставляет максимально обобщенные реализации концепций массива и записи. Неограниченные массивы и атрибуты массивов позволяют с минимальными издержками и затруднениями реализовывать различные алгоритмы обработки регулярных структур данных, а запись с дискриминантами, с одной стороны, намного нагляднее, чем в Паскале, а с другой стороны, дискриминант записей в Аде позволяет не только управлять набором компонент, но и свойствами отдельных полей записи.

Ада предоставляет более строгую, чем Паскаль, реализацию концепции строгой типизации.

Концепция подтипа и механизм исключительных ситуаций позволяют продемонстрировать идею ограничения множества значений объекта данных и динамического контроля ограничений.

Уже упоминавшиеся пакеты, одна из форм программных модулей в языке Ада, является базовым средством структурирования кода, ограничения видимости (часть описаний публичны, другие приватны, а остальные – вовсе закрыты для пользователей пакетов) и организации наследования типов (охватывая и то, что в других языках известно как наследование «объектов» или «классов»)

Наконец, Ада позволяет уже в рамках курса основ программирования рассматривать концепцию типа как параметра настройки модулей-заготовок.

Ада в дисциплинах профессиональной ориентации. Большинство студентов (и не только студентов) считает объектно-ориентированным тот язык, в котором есть синтаксически оформленное понятие класса, а объектно-ориентированной программой – программу, использующую эти классы. На самом же деле такое представление об объектной ориентации является не только однобоким, но и в корне неверным. Ада реализует полноценную модель объектно-ориентированного программирования на основе понятия типа данных, не используя синтаксически оформленного понятия класса или объекта. Кроме того, Аде реализует концепцию динамического полиморфизма без явного определения операции как виртуальной. Таким образом, рассмотрение Ады в курсах объектно-ориентированного программирования наряду с языками, реализующими объектно-ориентированную парадигму на основе конструкции класса, позволяет сконцентрироваться не столько на синтаксисе «классов» (или их аналогов), а на сути самой парадигмы объектно-ориентированного программирования.

Ада отличается от остальных современных индустриальных языков программирования тем, что предоставляет высокоуровневые средства работы с асинхронными процессами, причем не в форме доступа к стандартным или специализированным библиотекам, а в форме языковых конструкций высокого уровня, согласованных с концепцией типа данных. Асинхронно взаимодействующие процессы имеют форму модулей-задач. Программа может запускать и останавливать задачи, а также строить из них структуры данных, скажем, массивы. Входы задач и операторы приема вызовов входов позволяют организовать синхронизацию параллельных процессов с одновременным обменом данными. Но самым интересным являются операторы, позволяющие асинхронному процессу выбирать тот или иной вариант своего поведения в зависимости как от своего состояния, так и от состояния других процессов.

Имеются также программные модули-мониторы (в смысле мониторов Хоара), так называемые, защитные модули, которые предельно упрощают надёжное программирование координации доступа многих процессов (задач) к неразделяемым ресурсам (скажем, памяти).

В результате полное решение для классической задачи асинхронного программирования – взаимодействие процесса-поставщика и процесса-потребителя через буфер – занимает на Аде один экран текста и оказывается чуть ли не одним из самых элементарных упражнений для начинающих

Теперь обратимся к одному знаменательному явлению в информационных технологиях. Понятие большой программы меняется с каждым годом. Те программы, которые считались большими и очень большими пять-десять лет назад, сейчас в лучшем случае могут быть признаны средними по размеру. Поэтому не только руководителям программных проектов, но даже участникам на роли рядового исполнителя необходимо иметь дело с технологиями, которые позволяют (при их умелом использовании) сохранять управляемость всё более растущих в размерах и всё более сложных по логике программными системами. Затронем лишь аспект много-многомодульности таких систем.

Модель распределённости и модульности программ, реализуемая языком Ада, не только решает технические проблемы по принципу «разделяй и властвуй», но предоставляет уникальную по сравнению с другими универсальными языками возможность предварительной разработки и компиляции формальных спецификаций. Таким образом, студент может иметь реальный практикум программирования в спецификациях без томительного изучения какого-либо из специальных языков спецификаций. Это типичный пример существенной экономии учебного времени в случае, если Ада – базовый язык обучения!

Ада – единственный на сегодняшний день язык, предоставляющий стандартную библиотеку для извлечения синтаксической и семантической информации из программ на этом языке (ASIS). Это – открытая дверь в освоение CASE-технологий, программометрический анализ и нестандартные решения высокой эффективности в сфере информационных технологий.

Заключение. Многие специалисты в области образования отдают, как известно, предпочтение языку Паскаль в роли первого языка обучения программированию. Мы здесь показали, что замена в это роли Паскаля на язык Ада не может быть хуже по любым критериям, и явно несёт определённые преимущества. В университетских программах образования в области информационных технологий (то есть по специальностям, которые можно отнести к направлениям подготовки «компьютерные науки», «программная инженерия», «прикладная математика-информатика») будут получены дальнейшие преимущества за счёт использования Ады в фундаментальных курсах профессиональной ориентации, как-то технологии программирования, параллельные вычисления, системы РВ, разработка больших программных проектов.

Испытывая «дефицит места» мы сэкономили на списке литературы. Но необходимые ссылки и даже полные тексты многих источников, пути доступа к компиляторам и технологиям языка Ада имеются здесь:

<http://www.ada-ru.org/index.html> (язык сайта – русский)

Связь с авторами: mischenko@univer.kharkov.ua, rybin@adacore.com

Ниже на фотографиях, сделанных на международной конференции в ОГУ 10 октября 2006 г.:

Фото 1. В.О. Мищенко делает по статье доклад, раскрывающий тезис "Ада и функции базового языка университетского образования"

Фото 2. В рядах секции "Методика преподавания информатики" внимательно слушают доклады ...

Фото 3. С.И. Рыбин делает доклад по другому тезису статьи "Уникальные возможности языка Ада в курсах профессиональной ориентации в области ИТ"

Фото 4. Вопросы, обсуждение, итоги - всё, как полагается, если тема вызвала интерес

