

О ГРАМОТНОЙ АЛГОРИТМИЗАЦИИ (Опыт построения систематического базового курса программирования)

Илья Евгеньевич Ермаков
*Ген. директор ООО «Метасистемы»,
зав. лабораторией инструментального ПО
Технологического института ОрёлГТУ*

Борис Валерьевич Рюмшин
*Директор по науке ООО «Метасистемы»,
преподаватель физико-математического факультета ОГУ*

Задачи коллектива:

- исследования в области системного и инструментального ПО;
- коммерческие промышленные проекты;
- обучение студентов и их вовлечение в решение сложных исследовательских задач.

Технологическая платформа

- На состояние 2005 г. использовались Delphi и C++;
- был начат эксперимент по переводу работ и обучения на Ada.

- Решение было подсказано событиями осени 2005 г.:
- визит Никлауса Вирта; деятельность проекта *Информатика-21*.

Оберон и Информатика-21

- Оберон в виде интегрированной системы программирования Блэкбокс/Компонентный Паскаль — отличный кандидат на единую среду для наукоёмкого программирования, обучения и промышленной разработки.
- Информатика-21 — задача создания единой системы обучения программированию от 5 классов школ до младших курсов вузов, на базе платформы Блэкбокс/КП.
- <http://www.inr.ac.ru/~info21>
- Особое внимание — на проблему обучения программированию непрофессионалов (физиков, инженеров, лингвистов...)

От инструментария к общим вопросам

Почему смешивается вопрос выбора инструмента, платформы (вроде бы тактический) с принципиальными вопросами содержания обучения?

- Инструмент должен уметь «исчезать», становиться гладким продолжением мысли. Выбрать один раз — и забыть.
- На начальном этапе учащийся ещё не отделяет содержания от формы; поэтому форма (инструмент) должна быть безупречной, дефекты недопустимы. (Тезис В.В. Потопахина)

Наше видение базового курса

- Программирование как продолжение математики, «исполняемая математика», замыкание мат. моделей на физическую реализацию.
- Специалист должен уметь свободно, «бесшовно» переходить от деятельности верхних уровней к алгоритмической реализации.

Уровни работы

Концептуальное мышление

Математические методы

Формализация (моделирование)

Продолжение модели до алгоритмической
реализации

Исполнение на машине

Требуется:

- Научить прикладника без усилий доводить идеи до программной реализации.
- Научить программиста проводить большую часть работы на уровне концепций и математических моделей, проецируя решения в код рутинным образом.

Основа для этого:

- Качественный инструментарий («гладкое продолжение прикладной математики»).
- Владение аппаратом доказательного программирования - т. е. уверенного вывода программ из соображений высокого уровня.

Аппарат анализа свойств

Программирование оказывается одновременно и продолжением математики, и инженерной деятельностью (построение технических систем).

- Основа любой инженерии — наличие математизированного аппарата анализа свойств создаваемых конструкций.
- Если этот аппарат не используется, то деятельность не является инженерной, а находится на уровне ремесла-«знахарства».

Аппарат Дейкстры

Эдсгер Дейкстра «Дисциплина программирования»

Миф: «Строгое доказательство правильности громоздко и неприменимо на практике».

Реальность: речь идёт не о доказательствах постфактум, а об уверенном выводе правильных частей программ (в первую очередь - циклов) из общих соображений.

Миф: «Формализация деятельности программиста сковывает мышление и неэффективна».

Реальность: математические методы \neq только формальные методы. Доказать — значит, представить в такой форме, в которой правильность очевидна любому, кто владеет соответствующим аппаратом.

Метод Дейкстры - реальность

- Участники Информатики-21 используют методы Дейкстры и в практической деятельности, и в обучении.
- Общий аппарат вывода циклов был дополнен частным простым методом, упрощающим 80% случаев: базовыми схемами циклов «полный проход» и «линейный поиск». (Автор — координатор И-21 Ф.В. Ткачёв, спецкурс «Введение в современное программирование» на физфаке МГУ).
- Подробнее см. тезисы в сборнике и OberonCore.ru

На пути к общему базовому курсу

Необходим базовый курс программирования,
аналогичный — идеологически и в ряде
конкретных — отношений курсу
математического анализа.

Наш опыт

- 5 лет преподавания программирования в разных форматах
-
- 2004-2006 — школьный спецкурс «Основы программирования и дискретной математики»
- 2006-2008 — спецкурсы на физмате ОГУ «Программное конструирование», «Современные методы системного программирования».
- 2008-2009 — организация полной подготовки будущих сотрудников на базе факультета СПО Технологического института ОрёлГТУ.

Опорные материалы

В построении базового курса алгоритмизации и программирования мы опирались на следующие методические работы:

Э. Дейкстра. Краткое введение в искусство программирования (1971)

А.Г. Кушниренко и др. Основы информатики и ВТ (1991)

А.Г. Кушниренко, Г.В. Лебедев. Программирование для математиков (1988, курс Мехмата МГУ)

Ф.В. Ткачёв. Конспекты лекций, физфак МГУ.

В.Н. Касьянов, В.К. Сабельфельд. Сборник заданий по практикуму на ЭВМ. (1986)

Разделение на этапы

- I этап — обучение проектированию алгоритма как спецификации поведения. Нет работы с данными. Используем учебных исполнителей (расширяем как угодно среду ББ/КП).
- Пошаговая детализация с уточнением.
- Декомпозиция на вспомогательные алгоритмы.
- Понятие обратной связи, планирующих конструкций (IF, WHILE)
- Предусловие, постусловие, инвариант цикла.
- Построение цикла WHILE раскрыткой от постусловия.
- WHILE, WHILE и ещё раз WHILE.

Разделение на этапы

II этап — алгоритмы с переменными.

- Понятие переменной вводится чистым образом (т. е. без опоры на понятие ячейки памяти и т. п.)
- Постепенный ввод понятия: сначала — однократные присваивания с пониманием переменной как обозначения для шага вычисления.
- Затем демонстрируем циклический процесс, порождающий последовательность значений (расчёт нерекуррентных числовых последовательностей).

- Наконец, переходим к циклическим процессам, порождающим рекуррентные последовательности (присваивания включают слева и справа одну и ту же переменную).
- Формируется понимание цикла как описания итерационного процесса, при котором вектор переменных сходится от начальных значений к результату. Результат является решением уравнения в предикатах — постусловия цикла.

Много интересных методически задач может быть решено только на уровне потоковых вычислений, без адресуемых структур данных. Это становится особенно интересно в свете распараллеливания и dataflow-архитектур.

III этап — структуры данных...

Ключевые идеи

- Программирование как гладкое продолжение математики
 - Требуется общий базовый курс программирования, подобный курсам математического анализа.

Ссылки:

- <http://www.inr.ac.ru/~info21>
- <http://OberonCore.ru>
- <http://metasystems.ru>