ОСНОВЫ ПРОГРАММИРОВАНИЯ

Программирование все в большей степени становится занятием лишь для профессионалов. Объявленный в середине 1980-х гг. лозунг «Программирование — вторая грамотность» остался в прошлом. В понятие «компьютерная грамотность» сегодня входит прежде всего навык использования многообразных средств информационных технологий. Решая ту или иную информационную задачу, необходимо выбрать адекватное программное средство. Это могут быть электронные таблицы, системы управления базами данных, математические пакеты и т.п. И только в том случае, когда подобные средства не дают возможности решить задачу, следует прибегать к универсальным языкам программирования.

Принято различать программистов двух категорий: прикладных и системных. Системные программисты — это разработчики базовых программных средств ЭВМ (операционных систем, трансляторов, сервисных средств и т.п.). Они являются профессионалами высочайшего уровня в программировании. Прикладные программисты разрабатывают средства прикладного программного обеспечения ЭВМ, предназначенные для решения задач из различных областей (наука, техника, производство, сфера обслуживания, обучение и т.п.). Требования к качеству как прикладных программ, так и системных сегодня очень высоки. Программа должна не только правильно решать задачу, но и иметь современный интерфейс, быть высоконадежной, дружественной по отношению к пользователю и т.д. Только такие программы могут выдерживать конкуренцию на мировом рынке программных продуктов. Программирование на любительском уровне сегодня никому не нужно.

По мере развития компьютерной техники развивались также и методика, и технология программирования. Сначала возникает командное и операторное программирование, в 1960-х гг. бурно развивается структурное программирование, появляются линии логического и функционального программирования, а в последнее время — объектно-ориентированное и визуальное программирование.

Задача, которую следует ставить при первоначальном изучении программирования, — освоение основ структурной методики программирования. Для указанной цели наиболее подходящим средством является язык программирования Паскаль. Автор языка Паскаль — швейцарский профессор Никлаус Вирт — создавал его именно для этого. Структурная методика остается основой программистской культуры. Не освоив ее, человек, взявшийся изучать программирование, не имеет никаких шансов стать профессионалом.

Реализации Паскаля в версиях фирмы Borland для IBM, известных под названием Турбо Паскаль, значительно расширили язык по сравнению с вариантом Вирта. Начиная с версии 5.5 Турбо Паскаль становится также и языком объектного программирования.

Содержание третьей главы настоящего учебника ориентировано на глубокое освоение студентами базовых понятий языков программирования высокого уровня в их реализации на Паскале. Такая подготовка облегчает изучение других языков программирования. Изложение основ языков Си/Си++ в четвертой главе опирается на знания и навыки, полученные учащимися при изучении Паскаля.

При изучении данного курса студентам понадобятся знания основ алгоритмизации в рамках школьного базового курса информатики. Обычно в школе алгоритмизация изучается с использованием учебных исполнителей, с помощью которых можно успешно освоить основы структурной методики, а именно:

- построение алгоритмов из базовых структур;

- применение метода последовательной детализации.

Желательным является знакомство с архитектурой ЭВМ на уровне машинных команд (достаточно на модельных примерах учебных компьютеров, изучаемых в школьной информатике; совсем' не обязательно изучение реальных языков команд или ассемблера). Эти знания позволяют освоить основные понятия программирования, такие как переменная, присваивание; «входить в положение транслятора» и благодаря этому не делать ошибок, даже не помня каких-то деталей синтаксиса языка; предвидеть те «подводные камни», на которые может «напороться» ваша программа в процессе выполнения. По существу, все эти качества и отличают профессионального программиста от дилетанта.

Еще одно качество профессионала — способность воспринимать красоту программы, получать эстетическое удовольствие оттого, что она хорошо написана. Нередко это чувство помогает интуитивно отличить неправильную программу от правильной. Однако основным критерием правильности является, безусловно, не интуиция, а грамотно организованное тестирование.

Процесс изучения и практического освоения программирования делится на три части:

- изучение методов построения алгоритмов;

- изучение языка программирования;

- изучение и практическое освоение определенной системы программирования .

Решению первой задачи посвящены вторая и пятая главы учебника. Во второй главе даются основные, базовые понятия и принципы построения алгоритмов работы с величинами. В пятой главе излагаются некоторые известные методики полного построения алгоритмов, обсуждаются проблемы тестирования программ, оценки сложности алгоритмов.

Языки программирования Турбо Паскаль и Си/Си++ излагаются в третьей и четвертой главах учебника соответственно. Подчеркнем, что данная книга — это прежде всего учебник по программированию, а не по языкам Паскаль и Си. Поэтому исчерпывающего описания данных языков вы здесь не найдете. Языки излагаются в том объеме, который необходим для начального курса программирования. Более подробное описание языков можно найти в книгах, приведенных в списке литературы.

В учебнике нет инструкций по работе с конкретными системами программирования для изучаемых языков. С ними студенты должны познакомиться в процессе практики на ЭВМ, используя другие источники.

Шестая глава представляет собой достаточно большой задачник по программированию. Этот задачник можно использовать для организации практических и лабораторных занятий по любому из изучаемых языков.