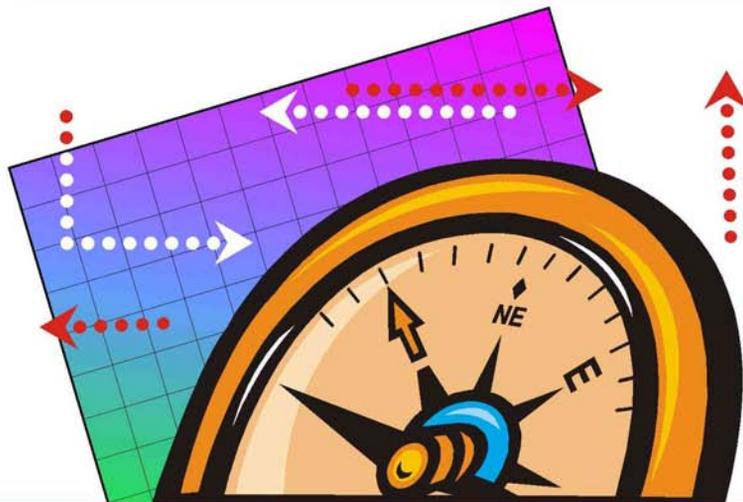


# Профессиональное программирование Системный подход

## 2 - е издание

- Методологии и технологии программирования
- Языки и системы программирования
- Аппаратная и операционная платформы
- Управление программным проектом
- Социальные вопросы и профессиональная карьера



**МАГИЕР**

**Игорь Одинцов**

**Профессиональное  
программирование  
Системный подход  
2 – е издание**

Санкт-Петербург  
«БХВ-Петербург»  
2004

УДК 681.3.068  
ББК 32.973-018  
О-42

**Одинцов И. О.**

О-42 Профессиональное программирование. Системный подход.— 2-е изд. перераб. и доп. — СПб.: БХВ-Петербург, 2004. — 624 с.: ил.

ISBN 5-94157-457-6

Подробно рассматриваются методологии, технологии, языки и системы программирования, аппаратная и операционная платформы, как давно известные, так и новейшие. Представлены различные их классификации. Описывается управление программным проектом и его особенности. Обсуждаются планирование, мотивация и реализация профессиональной карьеры в программировании и другие важные социальные вопросы информатики. Предлагается большой список литературы, включающий основные работы по каждой из исследуемых областей. Во втором, существенно переработанном и дополненном издании, отражены изменения последних лет в области информационных систем и дана оценка текущему состоянию дел в анализируемых областях.

*Для начинающих и профессиональных программистов, менеджеров проектов*

УДК 681.3.068  
ББК 32.973-018

### **Группа подготовки издания:**

Главный редактор	<i>Екатерина Кондукова</i>
Зам. главного редактора	<i>Евгений Рыбаков</i>
Зав. редакцией	<i>Григорий Добин</i>
Редактор	<i>Нина Седых</i>
Компьютерная верстка	<i>Татьяны Олоновой</i>
Корректор	<i>Татьяна Звертановская</i>
Дизайн обложки	<i>Игоря Цырульникова</i>
Зав. производством	<i>Николай Тверских</i>

Лицензия ИД № 02429 от 24.07.00. Подписано в печать 05.05.04.

Формат 70×100<sup>1</sup>/<sub>16</sub>. Печать офсетная. Усл. печ. л. 50,31.

Тираж 3000 экз. Заказ №

"БХВ-Петербург", 190005, Санкт-Петербург, Измайловский пр., 29.

Гигиеническое заключение на продукцию, товар № 77.99.02.953.Д.001537.03.02 от 13.03.2002 г. выдано Департаментом ГСЭН Минздрава России.

Отпечатано с готовых диапозитивов  
в Академической типографии "Наука" РАН  
199034, Санкт-Петербург, 9 линия, 12.

ISBN 5-94157-457-6

Одинцов И. О., 2004  
Оформление, издательство "БХВ-Петербург", 2004

# Содержание

<b>Предисловие</b> .....	<b>11</b>
Цель книги .....	11
Для кого предназначена эта книга .....	13
Организация книги .....	13
Профессиональные рамки .....	14
Основные направления профессионального программирования .....	15
Рекомендации по порядку чтения глав .....	18
Принятые в книге обозначения .....	18
Благодарности .....	19
<b>Предисловие ко 2-му изданию</b> .....	<b>21</b>
Основные изменения в книге и в мире .....	21
Книга как отправная точка .....	22
Благодарности .....	23
<b>Глава 1. Профессиональное программирование</b> .....	<b>25</b>
1.1. Профессионализм в программировании .....	25
1.1.1. Функции, черты и особенности мышления профессиональных программистов .....	27
1.2. Программирование. Что это такое? .....	31
1.3. Программирование. Чему учиться? .....	34
1.3.1. Подход на основе рекомендаций ACM/IEEE .....	35
1.3.2. Российские программы по информатике .....	37
1.3.3. Точка зрения опытных программистов .....	39
1.3.4. Подход, предлагаемый данной книгой .....	41
1.4. Программирование. Как учиться? .....	43
1.4.1. Подход на основе рекомендаций ACM/IEEE .....	43
1.4.2. Рекомендации российских специалистов .....	45
1.4.3. Авторские подходы .....	46
1.4.4. Отступление "О роли наставника в процессе обучения" .....	48
1.5. Программирование. Искусство, наука, ремесло или технология? .....	49

1.6. Кризис программирования .....	53
1.7. Основные понятия профессионального программирования .....	55
1.7.1. Отступление "о понятиях, определениях и классификациях" .....	55
1.7.2. Программа и программный продукт .....	57
1.7.3. Алгоритмы.....	58
1.7.4. Модели и моделирование .....	60
1.7.5. Жизненный цикл программного обеспечения .....	61
1.7.6. Проект и команда .....	62
1.7.7. Платформы .....	67
1.8. Основная задача профессионального программирования .....	69
Литература.....	70
<b>Глава 2. Методологии программирования.....</b>	<b>73</b>
2.1. Введение в методологии программирования .....	73
2.1.1. Основные понятия и определения.....	73
2.1.2. История и эволюция.....	75
2.1.3. Классификации методологий .....	77
2.2. Ядра методологий.....	79
2.2.1. Методология императивного программирования .....	79
2.2.2. Методология объектно-ориентированного программирования .....	83
2.2.3. Методология функционального программирования .....	87
2.2.4. Методология логического программирования .....	91
2.2.5. Методология программирования в ограничениях .....	94
2.3. Топологическая специфика методологий.....	97
2.3.1. Методология структурного императивного программирования .....	97
2.4. Реализационная специфика методологий .....	99
2.4.1. Методология императивного параллельного программирования....	99
2.4.2. Методология логического параллельного программирования .....	103
2.5. Заключение .....	105
2.5.1. Выводы и рекомендации .....	105
2.5.2. Проблемы и перспективы развития.....	105
2.5.3. Рекомендации по литературе.....	107
2.5.4. Вопросы и задания .....	108
Литература.....	108
<b>Глава 3. Технологии программирования .....</b>	<b>111</b>
3.1. Введение в технологии программирования.....	111
3.1.1. Основные понятия и определения.....	111
3.1.2. История и эволюция.....	112
3.1.3. Классификации .....	113
3.2. Классические технологические процессы.....	117
3.2.1. Возникновение и исследование идеи .....	117

3.2.2. Управление .....	122
3.2.3. Анализ требований и проектирование .....	136
3.2.4. Программирование (реализация) .....	153
3.2.5. Тестирование и отладка .....	161
3.2.6. Ввод программы в действие .....	166
3.2.7. Эксплуатация и сопровождение .....	167
3.2.8. Завершение эксплуатации .....	170
3.3. Стандартные технологические процессы .....	171
3.3.1. Основные процессы .....	171
3.3.2. Вспомогательные процессы .....	173
3.3.3. Организационные процессы .....	175
3.3.4. Взаимосвязь между процессами .....	176
3.4. Основные стадии технологических подходов .....	177
3.4.1. Фазы как крупные временные рамки .....	177
3.4.2. Стадии как отражение классических процессов .....	177
3.4.3. Вариант подробного разбиения на стадии .....	178
3.4.4. Контрольные точки .....	180
3.5. Основные технологические подходы .....	181
3.5.1. Ранние технологические подходы .....	181
3.5.2. Каскадные технологические подходы .....	181
3.5.3. Каркасные технологические подходы .....	185
3.5.4. Генетические технологические подходы .....	190
3.5.5. Подходы на основе формальных преобразований .....	192
3.5.6. Группа ранних подходов быстрой разработки .....	196
3.5.7. Адаптивные технологические подходы .....	198
3.5.8. Подходы исследовательского программирования .....	203
3.6. Технологии коллективной разработки .....	205
3.6.1. Авторская разработка .....	205
3.6.2. Коллективная разработка .....	206
3.6.3. Общинная модель разработки .....	213
3.6.4. Отступление "об оффшорном программировании" .....	214
3.7. Качество программного обеспечения .....	216
3.7.1. Подходы к качеству программного обеспечения .....	217
3.7.2. Характеристики качества программного обеспечения .....	218
3.7.3. Оценка качества процесса разработки .....	220
3.7.4. "Достаточно хорошее" программное обеспечение .....	223
3.7.5. Стандартизация информационных технологий .....	224
3.8. Заключение .....	226
3.8.1. Выводы и рекомендации .....	226
3.8.2. Проблемы и перспективы развития .....	229
3.8.3. Рекомендации по литературе .....	230
3.8.4. Вопросы и задания .....	231
Литература .....	233

<b>Глава 4. Языки программирования.....</b>	<b>237</b>
4.1. Введение в языки программирования .....	237
4.1.1. Основные понятия и определения.....	238
4.1.2. История и эволюция.....	245
4.1.3. Классификация языков.....	247
4.2. Языки, поддерживающие классические технологические процессы..	249
4.2.1. Возникновение и исследование идеи .....	249
4.2.2. Управление .....	249
4.2.3. Анализ требований и проектирование .....	251
4.2.4. Программирование (реализация).....	251
4.2.5. Тестирование и отладка .....	251
4.2.6. Ввод в действие .....	251
4.2.7. Эксплуатация и сопровождение.....	251
4.2.8. Завершение эксплуатации.....	252
4.3. Языковые абстракции .....	252
4.3.1. Отступление "об абстрагировании" .....	252
4.3.2. Абстракция данных.....	253
4.3.3. Абстракция управления.....	257
4.3.4. Абстракция модульности.....	264
4.4. Языки моделирования .....	269
4.4.1. Моделирование на основе структурной методологии .....	269
4.4.2. Моделирование на основе объектно-ориентированной методологии .....	271
4.4.3. Языки моделирования данных .....	277
4.4.4. Языки моделирования знаний.....	279
4.5. Языки программирования высокого уровня.....	282
4.5.1. Обзор языков, принадлежащих к семействам .....	282
4.5.2. Обзор языков, ориентированных на предметную область.....	288
4.6. Формальные языки .....	291
4.6.1. Иерархия грамматик.....	292
4.6.2. Техники распознавания (разбора).....	293
4.6.3. Образное сравнение типов грамматик .....	294
4.6.4. Метатрансляция .....	294
4.6.5. Язык спецификации грамматик.....	297
4.7. Естественные языки.....	298
4.7.1. Особенности естественных языков и культурных сред.....	298
4.7.2. Семантический анализ естественных языков.....	300
4.7.3. Интернационализация и локализация программных продуктов ...	302
4.7.4. Отступление "о придании тексту эмоций" .....	304
4.8. Заключение .....	306
4.8.1. Выводы и рекомендации .....	306
4.8.2. Проблемы и перспективы развития.....	307
4.8.3. Рекомендации по литературе.....	308
4.8.4. Вопросы и задания .....	309
Литература.....	311

<b>Глава 5. Системы программирования .....</b>	<b>315</b>
5.1. Введение в системы программирования.....	315
5.1.1. Основные понятия и определения.....	315
5.1.2. История и эволюция.....	317
5.1.3. Классификация .....	318
5.2. Процесс-ориентированный инструментарий.....	320
5.2.1. Возникновение и исследование идеи .....	320
5.2.2. Управление .....	321
5.2.3. Анализ требований и проектирование .....	324
5.2.4. Программирование (реализация) .....	326
5.2.5. Тестирование и отладка .....	338
5.2.6. Ввод в действие .....	343
5.2.7. Сопровождение .....	343
5.2.8. Завершение эксплуатации.....	345
5.3. Универсальный инструментарий.....	345
5.3.1. Инструменты работы с текстом .....	345
5.3.2. Системы документирования .....	351
5.3.3. Системы разработки интерфейсов.....	353
5.3.4. Системы управления базами данных.....	353
5.3.5. Системы управления базами знаний и экспертные системы.....	354
5.3.6. Электронные библиотеки и инструментарий Интернета .....	358
5.4. Инструментарий поддержки процессов некоторых технологических подходов .....	368
5.4.1. Системы верификации программ .....	368
5.4.2. Системы формального преобразования программ.....	368
5.4.3. Средства сборочного программирования.....	368
5.5. Инструментальные системы.....	369
5.5.1. Инструментальные среды программирования.....	369
5.5.2. Средства автоматизации разработки программ (CASE-средства) .....	370
5.5.3. Интегрированные среды .....	371
5.5.4. Репозитории проекта.....	372
5.6. Средства поддержки коллективной разработки .....	376
5.6.1. Системы разделения файлов .....	376
5.6.2. Системы поддержки работы виртуальных групп .....	380
5.7. Естественно-языковой интерфейс.....	382
5.7.1. Диалоговые системы.....	382
5.7.2. Вопросно-ответные системы .....	383
5.7.3. Автоматизированные обучающие системы и системы контроля знаний .....	384
5.7.4. Системы искусственного интеллекта .....	385
5.8. Заключение .....	386

5.8.1. Выводы и рекомендации .....	386
5.8.2. Проблемы и перспективы развития.....	389
5.8.3. Рекомендации по литературе.....	391
5.8.4. Вопросы и задания .....	392
Литература.....	393
<b>Глава 6. Архитектурная платформа.....</b>	<b>397</b>
6.1. Основы архитектуры ЭВМ.....	397
6.1.1. Основные понятия и определения.....	397
6.1.2. История и эволюция компьютерных архитектур.....	398
6.1.3. Классификация вычислительных систем.....	402
6.2. Основные архитектуры набора команд.....	406
6.2.1. Классические архитектуры .....	406
6.2.2. Архитектуры CISC, RISC и VLIW .....	409
6.2.3. Вычисления .....	411
6.2.4. Архитектура ЭВМ и языки программирования .....	414
6.2.5. Оценка производительности вычислительных систем.....	414
6.3. Организация вычислительной системы.....	416
6.3.1. Процессор .....	416
6.3.2. Память.....	416
6.3.3. Шина .....	417
6.3.4. Устройства ввода и вывода данных .....	418
6.3.5. Функциональное описание.....	418
6.4. Особенности SPARC-архитектуры.....	419
6.4.1. История создания SPARC-архитектуры.....	419
6.4.2. Основные понятия SPARC-архитектуры .....	420
6.4.3. Основной набор команд языка Assembler.....	423
6.4.4. Примеры программ на языке Assembler.....	429
6.4.5. Организация SPARC-архитектуры.....	432
6.5. Параллельные и распределенные архитектуры .....	434
6.5.1. Основные классы параллельных архитектур .....	434
6.5.2. Коммутаторы вычислительных систем.....	435
6.5.3. Некоторые классификации параллельных архитектур.....	437
6.6. Архитектура компьютерных сетей.....	439
6.6.1. Классификация сетей и сетевые топологии .....	439
6.6.2. Стандарты в области сетей IEEE 802 .....	441
6.6.3. Аппаратная поддержка локальных сетей .....	442
6.6.4. Глобальная сеть Интернет .....	444
6.7. Заключение .....	447
6.7.1. Выводы и рекомендации.....	447
6.7.2. Проблемы и перспективы развития.....	449
6.7.3. Рекомендации по литературе.....	451
6.7.4. Вопросы и задания .....	452
Литература.....	453

<b>Глава 7. Операционная платформа .....</b>	<b>455</b>
7.1. Введение в операционные системы .....	455
7.1.1. Основные понятия и определения.....	455
7.1.2. История и эволюция операционных систем .....	457
7.1.3. Классификация операционных систем .....	460
7.2. Процессы.....	465
7.2.1. Процессы и потоки (нити) управления .....	465
7.2.2. Коммуникация и синхронизация процессов в централизованных архитектурах .....	474
7.2.3. Коммуникация процессов в сетях.....	491
7.2.4. Синхронизация процессов в распределенных системах .....	505
7.3. Память .....	514
7.3.1. Основная память.....	514
7.3.2. Внешняя память.....	521
7.4. Драйверы устройств .....	526
7.5. Заключение .....	527
7.5.1. Выводы и рекомендации.....	527
7.5.2. Проблемы и перспективы развития.....	529
7.5.3. Рекомендации по литературе.....	531
7.5.4. Вопросы и задания .....	532
Литература.....	534
<b>Глава 8. Элементы менеджмента и социальные вопросы информатики.....</b>	<b>537</b>
8.1. Введение в элементы менеджмента и социальные вопросы информатики.....	537
8.2. Набор новых сотрудников.....	540
8.2.1. Привлечение кандидатов .....	541
8.2.2. Анализ краткой биографии (резюме) .....	542
8.2.3. Собеседование .....	547
8.2.4. Испытательный срок .....	564
8.3. Карьера в области информационных систем.....	565
8.3.1. Карьера и ее планирование .....	565
8.3.2. Совершенствование мастерства.....	569
8.3.3. Олимпиадное (соревновательное) программирование .....	571
8.3.4. Философия и стратегия программистских коллективов и организаций .....	578
8.3.5. Формы власти и стили руководства.....	579
8.3.6. Договорная основа трудовых отношений .....	580
8.3.7. Заработная плата .....	581
8.4. Почему некоторые программисты на порядок полезнее других? .....	584
8.4.1. Насколько корректна постановка проблемы? .....	585

8.4.2. Характеристики хорошего программиста.....	587
8.4.3. Можно ли стать полезнее?.....	591
8.5. Этические ответственности.....	592
8.5.1. Этика в работе программиста.....	592
8.5.2. "Феномен науки" В. Ф. Турчина.....	595
8.5.3. Отступление "о свободе и стоимости программного обеспечения".....	596
8.6. Заключение.....	599
8.6.1. Выводы и рекомендации.....	599
8.6.2. Рекомендации по литературе.....	601
8.6.3. Вопросы и задания.....	602
Литература.....	603
<b>Предметный указатель.....</b>	<b>605</b>

# Предисловие

## Цель книги

Знать цель, к которой мы стремимся — это рассудительность;  
достигнуть этой цели — верность взгляда;  
остановиться на ней — сила;  
попасть далее цели — дерзость.

*Ш. Дюкло*

В информационном океане давно появился архипелаг "Программирование". К первому однобитному островку, вынырнувшему из пучин, стремительно начали пристраиваться другие. Здесь есть и большие надежные острова, например, операционной системы Unix, языка программирования Pascal, объектно-ориентированной методологии. Есть молодые с бурной вулканической деятельностью и неустойчивой почвой. Есть малые пяточки terra incognita. Некоторые острова, исчерпав себя, распались сами, и о них забыли. Другие, имеющие прочную основу, укрепились надолго.

Если Вы программист-"капитан", программист-"бывалый моряк" или пока просто программист-"юнга", то корабль с компасом и цель путешествия у Вас уже есть. Чтобы хорошо ориентироваться среди островов этого архипелага, нужны еще карта и хороший лоцман. Книга, которую Вы держите в руках, — именно то, что Вам необходимо. Она дает Вам возможность проложить и в дальнейшем уточнять курс движения. Маяки, установленные для Вас, позволят Вам не заблудиться в тумане и обнаружить новые острова.

При ее написании автор ставил перед собой две цели.

- Познакомить Вас с современным состоянием развития основных областей программирования. С помощью этой книги Вы узнаете или пополните свои знания о методологиях, технологиях, языках и системах программирования, аппаратной и операционной платформах, как хорошо себя зарекомендовавших и устоявшихся, так и новейших.
- Предложить Вам систематизацию и классификацию этих областей.

В соответствии с замыслом в книге излагается много фактов и концепций, являющихся весьма разнородными. Ее содержание сложилось в результате анализа большого количества научных и научно-популярных источников и с учетом профессионального опыта автора.

При создании этой книги автор следовал следующим принципам.

- **Взаимосвязанный, системный подход к программированию.** Жизненный цикл программы практически в каждый момент времени имеет отношение к некоторому технологическому процессу, некоторому языку и некоторому инструменту системы программирования. Поэтому нам необходимо не только знание каждого из этих направлений, но и понимание теснейших внутренних и внешних связей и зависимостей между ними. А определяющую роль в выборе языков и систем (и, во многом, технологий) играет методология программирования.
- **Изложение программирования в обобщенном и систематизированном виде.** С одной стороны, здесь мы будем избегать излишнего теоретизирования. С другой стороны, мы не станем углубляться в технические детали, которые лучше изучать с помощью специализированной программной документации. Говоря проще, в большинстве случаев Вы не найдете здесь утверждений типа "переместите указатель на иконку и щелкните кнопкой мыши".

Давайте представим множество знаний в некоторой предметной области в виде дерева. Ствол дерева олицетворяет "практически вечные" знания (их немного), ветви — "долговременные", а листья — "мимолетные" (их подавляющее большинство). Для предметной области программирования эти три группы знаний выглядят так:

- фундаментальные знания, математические основы программирования (такие знания обычно остаются актуальными как минимум на протяжении человеческой жизни);
- знание методологий программирования, групп языков, классов систем и технологий, типов архитектурных и операционных платформ (время устаревания таких знаний около 10—15 лет);
- знание технических особенностей конкретной аппаратуры или конкретных программных средств (устаревание около 3 лет).

Основное внимание в этой книге уделяется второй группе знаний — "золотой середине". Иногда, в виде исключения, мы дадим Вам возможность явно полюбоваться красивым (но, в большинстве случаев, недолговечным) "листочком".

Отметим, что из-за невозможности "объять необъятное", данная книга во многих местах имеет краткий (энциклопедический) характер. Эта особенность компенсируется обилием ссылок на дополнительную литературу и страницы в сети Интернет. Эти ссылки позволят Вам самостоятельно вырастить многочисленные "листья" на уже известных Вам "ветвях".

## Для кого предназначена эта книга

Из всех связей, возникающих между людьми, самой непостоянной, самой запутанной и изменчивой является связь между писателем и читателем.

*А. Шейфсберги*

Книга будет полезна всем тем, кто занимается или собирается заниматься разработкой программного обеспечения.

- Профессионалы, уже имеющие большой опыт разработки программ, могут использовать данную книгу для систематизации своих знаний, а также в качестве справочника по основным разделам программирования. Книга поможет дополнить или уточнить имеющиеся у них данные, а также получить ссылки на подробные источники информации.
- Менеджеры проектов могут использовать информацию о процессе управления разработкой программного обеспечения.
- Начинающие программисты могут получить из книги представление об основных областях профессионального программирования, связанных с разработкой программ. Эта книга может являться для них стартовой площадкой, "печкой", от которой можно и нужно двигаться дальше, переходя к книгам по конкретным языкам и системам, а также к постоянному и систематическому практическому программированию.
- Студенты могут использовать отдельные главы книги в качестве учебного пособия. Особенно полезна книга будет студентам, обучающимся по специальностям 220400 "Программное обеспечение вычислительной техники и автоматизированных систем", 351400 "Прикладная информатика по областям", 351500 "Математическое обеспечение и администрирование информационных систем" и близким к ним.
- Преподаватели могут использовать книгу в целом и отдельные ее главы в качестве основы для построения развернутых лекционных курсов по этой тематике.

## Организация книги

То, что я понял, прекрасно, из этого я заключаю, что остальное, чего я не понял, тоже прекрасно.

*Сократ (по поводу неясностей у Гераклита)*

Книга состоит из восьми глав. Первая и восьмая — стартовая и заключительная — представляют практические знания и рекомендации в области профессионального программирования. Главы со второй по седьмую — это

внутренние главы, посвященные важнейшим направлениям профессионального программирования.

## Профессиональные рамки

### Глава 1. Профессиональное программирование

В этой главе рассматриваются вопросы, над которыми рано или поздно задумываются все программисты.

- Каковы функции, черты и особенности мышления профессиональных программистов?
- Что должны знать и уметь профессиональные программисты?
- Существуют ли стандарты для обучения программированию?
- Чем является программирование: искусством, наукой, ремеслом или технологией?
- Почему многие программные проекты выполняются с отставанием от графика, с превышением сметы расходов, при этом качество продукта не устраивает пользователя?

Завершающая часть главы посвящена основным понятиям и определениям из области профессионального программирования.

### Глава 8. Элементы менеджмента и социальные вопросы информатики

Разделы главы объединены понятием "человеческий фактор". Эта глава в одинаковой степени полезна и программистам, желающим устроиться на работу, и руководителям проектов, которые принимают их на работу. Мы рассмотрим уровни карьеры, вопросы планирования карьеры и примеры составления некоторых важных документов. Далее мы перечислим основные практические вопросы главы.

- Как сделать карьеру в области информационных систем? В чем особенности национальной карьеры?
- Какими должны быть первые шаги в поиске работы? Как правильно составить резюме? Как проводится собеседование и как к нему подготовиться?
- Почему некоторые программисты на порядок полезнее других?
- Что такое олимпиадное (соревновательное) программирование? Какая от него польза и связано ли оно с реальным программированием?
- Каковы этические проблемы, возникающие при разработке программного обеспечения? Как их можно решать с этической и технической точек зрения?

## Основные направления профессионального программирования

Прежде чем ознакомить вас с кратким изложением основных вопросов, затронутых в главах 2—7, определим главные направления. Основные области в профессиональном программировании представим тремя взаимосвязанными группами. Каждая группа — это объединение двух достаточно близких направлений.

### Направляющая группа

Это группа содержит два направления, с которыми следует определиться перед началом работы с программным проектом. Выбранные методы и подходы определяют основную идеологию и принципы проекта.

### Глава 2. Методологии программирования

В этой главе рассказывается об основных методологиях программирования — совокупности методов, применяемых в жизненном цикле программного обеспечения и объединенных общим философским подходом. Основные вопросы главы сформулируем следующим образом.

- Зачем нужны определяющие методологии в программировании?
- В чем заключается поддержка методологии методами и концепциями?
- Каковы вычислительные модели методологий?
- На какой класс практических задач ориентирована каждая из методологий?
- Какова языковая поддержка методологий?
- Каково происхождение методологий?

### Глава 3. Технологии программирования

Прочитав эту главу, вы получите ответы на вопросы, приведенные ниже.

- Что такое жизненный цикл программ и каковы основные технологические подходы к его ведению?
- Что такое классические и стандартные технологические процессы?
- Как формируются технологические стадии и в чем заключается порядок их прохождения?
- Что представляют собой технологии коллективной разработки программ?
- Как добиться создания качественного программного обеспечения?

## Инструментальная группа

Эта группа содержит два инструментальных направления: языковую и системную поддержку проекта.

### Глава 4. Языки программирования

Мы рассмотрим языки, участвующие в создании программы, поддерживающие основные технологические процессы. Для двух основных языковых групп — языков моделирования и языков программирования высокого уровня — можно выделить черты, о которых также пойдет речь в этой главе.

- Язык обладает характеристиками.
- Язык содержит абстракции.
- Язык поддерживает методологию программирования.
- Язык принадлежит к семейству языков.
- Язык может быть специализированным для некоторой предметной области.

Завершается глава кратким обсуждением формальных и естественных языков.

### Глава 5. Системы программирования

В этой главе мы подробно поговорим о нескольких группах инструментов, которые относятся к системам программирования, а именно:

- инструментарий поддержки технологических процессов. Мы выделим инструменты, ориентированные и применяемые только в одном процессе, и универсальные (независимые от процесса) инструменты;
- инструментальные системы (инструментальные среды программирования, средства автоматизации разработки программ, интегрированные среды и репозитории проекта);
- инструментарий поддержки коллективной разработки.

Глава также включает обзор систем поддержки естественно-языкового интерфейса.

## Базисная группа

Эта группа включает два направления, представляющих собой основные платформы, на которых базируется программный проект.

### Глава 6. Архитектурная платформа

Вся эта глава посвящена компьютерным архитектурам. Рассматриваются три основные группы вопросов.

- Основные архитектуры набора команд — классические, современные и нестандартные.

- Организация вычислительной системы — процессор, память, шина, функциональное описание.
- Параллельные и распределенные архитектуры. В том числе сети ЭВМ, как частный случай распределенной архитектуры.

## Глава 7. Операционная платформа

Изучение механизма и структуры операционных систем необходимо по следующим причинам:

- основные идеи, концепции и алгоритмы, лежащие в основе операционных систем, применимы ко многим другим областям программирования, и особенно к системному программированию;
- операционная система — большая и очень сложная программа, на примере которой можно изучать вопросы создания сложных программных продуктов;
- такие популярные программные продукты, как системы управления базами данных, могут рассматриваться как надстройки над операционными системами.

Каждая из глав 2—7 начинается тремя разделами, дающими подробное введение в изучаемую область:

- основные понятия и определения;
- история и эволюция;
- классификации.

Завершают каждую из этих глав четыре раздела, подводящих итоги рассмотренному материалу:

- выводы и рекомендации;
- проблемы и перспективы развития;
- рекомендации по литературе;
- вопросы и задания.

## Литература

Чтение сделало Дон-Кихота рыцарем, а вера в прочитанное сделала его сумасшедшим.

*Джордж Бернард Шоу*

Книга содержит список литературы, охватывающий основные работы в исследуемых областях, изданные в нашей стране и за рубежом. Несмотря на то, что мы стремимся включить в книгу самую актуальную информацию, рано или поздно она устареет. Эффективно бороться с этим можно только с помощью Интернета. Если Вы еще не знаете как, то книга научит Вас методам профессионального поиска.

## Рекомендации по порядку чтения глав

Природа подобна женщине, которая... показывая из-под своих нарядов то одну часть своего тела, то другую, подает своим настойчивым поклонникам некоторую надежду узнать ее когда-нибудь всю.

*Дени Дидро*

Мы предлагаем три основных варианта прочтения этой книги.

- С последовательным обращением. Этот способ классический, когда главы читают в порядке их расположения. Этот вариант предлагается в первую очередь начинающим программистам.
- С параллельным обращением. Рекомендуется желающим систематизировать свои знания. Он требует параллельного чтения нескольких глав, поскольку некоторые изучаемые в книге вопросы образуют тройственную взаимосвязь, которую надо отслеживать. Так, в третьей главе может излагаться некоторый технологический процесс, в четвертой — языки, а в пятой — системы, его поддерживающие.
- С произвольным обращением. Третий способ — обращение к главам по мере необходимости или интереса читателя к изложенным вопросам. Опытные программисты могут использовать книгу как справочник, просматривая материал по интересующим их методологиям и технологиям, языкам и системам, аппаратной и операционной платформам.

## Принятые в книге обозначения

Уже сам знак параграфа выглядит как орудие пытки.

*Ежи Лец*

Для облегчения работы с текстом в книге используются следующие соглашения:

- коды программ и фрагменты документов выделены специальным шрифтом (Courier);
- базовые понятия и определения выделены *курсивом* при их первом появлении в тексте;
- замечания к основному тексту выделены рубленым шрифтом;
- для ссылок в тексте на литературу на бумажном носителе используются квадратные скобки (например, [Лавров 2001]). Списки литературы приводятся в конце каждой главы;

- ссылки на ресурсы Интернета (адреса сайтов) даются непосредственно в тексте книги;
- в книге применяется система нумерации, достигающая четырех уровней вложенности:
  - номера глав состоят из одной цифры, ссылка на главу производится, например, так: *гл. 1*;
  - номера разделов состоят из двух, трех и четырех цифр, ссылки на разделы осуществляются, например, так: *разд. 1.2*, *разд. 1.2.3*, *разд. 1.2.3.4*;
- названия языков и систем программирования, операционных систем и т. п. будут приводиться на русском языке, если они созданы в России (СССР), и на английском, если за пределами России (СССР), например РЕФАЛ и Java;
- для большинства понятий на русском языке при их первом появлении будет приведен англоязычный аналог.

## Благодарности

Благодарность — это тяжкое бремя, лежащее на нашей несовершенной природе.

*Ф. Честерфилд*

Хочу исполнить приятный долг и выразить благодарность всем тем, кто помог в работе над данной книгой. В первую очередь, это студенты математико-механического факультета СПбГУ, которые являлись терпеливыми слушателями различных курсов моих лекций и улучшали их своими вопросами и замечаниями.

Плодотворные длительные дискуссии с Денисом Черноивановым, Романом Шапошником, Михаилом Войтенко, Алексеем Духняковым и Дмитрием Петровым положительно повлияли на содержание книги. Дмитрий Петров также оказал значительную помощь в совершенствовании главы 2.

Глубокую признательность хочу выразить Игорю Некрестьянову за многочисленные обсуждения данной книги, настойчивые критические замечания и предложения, существенную поддержку и дружеское участие. Я благодарен своим коллегам — Юрию Юдину, Валерию Ушакову, Олегу Суходольскому, Екатерине Павловой и Алине Холчевой — за то, что они прочли черновые варианты этой книги. Их конструктивные замечания помогли улучшить изложение глав.

Я хотел бы поблагодарить руководителя проекта издательства "БХВ-Петербург" Евгения Рыбакова, который первый ознакомился с рукописью. Без его поддержки и настойчивости рукопись вряд ли оформилась бы в данную книгу. Хочу выразить благодарность Михаилу Кокореву за работу по редактированию книги.

Я буду благодарен и будущим критикам этой книги, которые, несомненно, появятся. Работа над текстом книги продолжается и она еще далека от завершения.

*Игорь Одинцов*

e-mail: **oio@nwgsm.ru**

*Февраль 2002 г.*

# Предисловие ко 2-му изданию

## Основные изменения в книге и в мире

Мафусаил жил 969 лет. Вы, дорогие мальчики и девочки, в следующие десять лет увидите больше, чем видел Мафусаил за всю свою жизнь.

*Марк Твен*

Прошло около двух лет с момента выхода первого издания. С определенной гордостью за книгу замечу, что подавляющая часть материала и на сегодняшний день остается актуальной. Тем не менее, мы постарались включить в книгу всю новую интересную информацию, которая появилась за это время. Значительная часть глав 1, 2, 3 и 8 переработана, добавлены новые разделы в главы 4—7. Исправлены замеченные в первом издании неточности и опечатки. Создан полноценный предметный указатель. Появились новые рисунки, ссылки на наиболее полезные книги, вышедшие за последние два года, а также новые ссылки на ресурсы сети Интернет. Учтены многие замечания и предложения, высказанные автору читателями книги (на электронный почтовый адрес пришло несколько сотен писем).

Первое издание книги задумывалось как энциклопедический справочник, свод по большей части беспристрастной информации, из которой каждый мог отобрать себе необходимую. Однако, по многочисленным просьбам читателей (особенно студентов), в данном издании появилось большое количество субъективных оценок автора, как непосредственно в тексте разделов, так и в итоговых разделах "Выводы и рекомендации" глав 2—7. Можно сказать, что теперь у читателя в руках не только "удочка", с помощью которой он будет ловить "рыбу", но и некоторое количество "рыбки" про запас на тот период, пока он не научится ее ловить полностью самостоятельно.

Изменения в компьютерном мире заслуживают отдельного анализа. Жестче стала "игра" компьютерных компаний, навязывающих покупателю свою продукцию, подчас сырую и некачественную. После разработки семиуровневой модели взаимодействия открытых систем (см. разд. 7.2.3.1) ряд аналитиков иронически замечал, что к уже существующим семи уровням можно добавить еще два: финансовый и политический. Сейчас очень многие платформы имеют эти два уровня в виде надстройки. Реальность и жесткость (жестокость?)

жизни такова, что по финансовым и политическим соображениям закрываются проекты, безотносительно к их реальным ценностям и пользе для программистского сообщества. Любой специалист в области разработки программного обеспечения должен быть готов к тому, что его проект будет закрыт по прихоти руководства компании, перебрасывающего силы на разработку новомодной идеи. Крупные компании давят на рынок, активно формируя нужный им спрос. Еще одно любимое занятие компаний по разработке программного обеспечения — постоянный выпуск новых версий, появление новых платформ и сред. Последнее время это сильно напоминает ситуацию в мире шоу-бизнеса, а не серьезную профессиональную работу.

Возникает вопрос: справедливо ли утверждение, что ничего нового за последние, например, 10 лет в области разработки программного обеспечения не появилось? В общем-то, нет. Можно найти достаточно много областей, в которых найдены методы и алгоритмы, позволившие сделать существенный прорыв. Типичный пример — новые алгоритмы, применяемые в поисковых системах в Интернете. Их использование значительно повысило адекватность (релевантность) запросу находимой в сети информации.

Впрочем, автор должен (фактически "по определению") являться оптимистом, как и большинство программистов. Следует помнить, что книга, утверждающая, что "во Вселенной есть жизнь", притягивает читателей гораздо сильнее, чем та, в которой более аргументировано говорится, что "во Вселенной нет жизни". Даже если обе книги написаны одним и тем же автором.

## Книга как отправная точка

Мастеръ Гамбсь этимъ полукресломъ  
начинаеть новую партію мебели.

1865 г. Санкт-Петербургъ.

*Илья Ильф и Евгений Петров*

Очевидное, на наш взгляд, утверждение следует напомнить здесь. Профессиональным программистом можно стать после упорной работы и практики, а не после прочтения книги (пусть даже самой лучшей) или после взмаха волшебной палочки. Книга *поможет* Вам стать профессионалом, но не сделает Вас им автоматически. Начинаящим программистам, стремящимся как можно скорее освоить программирование на некотором языке, можно предложить следующий план действий из трех пунктов.

- Прочитайте эту книгу.
- Прочитайте хорошую книгу с описанием языка программирования.
- Напишите хотя бы три достаточно серьезные программы на этом языке.

В последнем пункте можно использовать идею "подражания великим" и взять в качестве примеров аналоги существующих хорошо известных программ. Первая программа — файловый менеджер, аналог программы Norton

Commander (FAR manager). Заметим, что эта программа особенно любима многими отечественными программистами. Существует даже шуточное утверждение, что если за рубежом вы встречаете программиста и замечаете на экране его монитора хорошо знакомые синие панельки с желтыми буквами, то можете смело подходить и начинать разговор с ним по-русски. Впрочем, в каждой шутке есть доля шутки. Вторая программа — ftp-сервер, реализующий который Вы постигнете ряд тонкостей системного и сетевого программирования. Третьей задачей может стать реализация синтаксического анализатора (методом рекурсивного спуска) простого и красивого языка программирования Pascal. После ее написания у Вас появится понимание механизмов представления данных и работы с ними, реализованных в системах программирования.

Многие вопросы не удалось осветить в рамках этой книги. Автором готовится к изданию следующая книга, посвященная вопросу эволюции программных продуктов. В ней будет рассмотрено несколько подробных примеров разработки программных продуктов.

## Благодарности

В первую очередь благодарю Вас, уважаемый читатель этой книги. Собственно, книга без Вас была бы также бессмысленна, как и разработанный программный продукт, не нашедший своего пользователя. Если Вы читаете ее впервые, надеюсь, она оправдает Ваши ожидания, а если Вы читали первое издание — спасибо за постоянство и доверие. Особая благодарность тем читателям первого издания, которые прислали мне письма. Я ответил всем.

Выражаю благодарность человеку, являющемуся легендой российского программирования, — члену-корреспонденту АН Святославу Сергеевичу Лаврову. Его "плотность критических замечаний" по рукописи книги была максимальной среди всех читателей и рецензентов. И если второе издание книги стало лучше, то во многом благодаря Святославу Сергеевичу.

Я благодарен профессору Санкт-Петербургского государственного университета Иосифу Владимировичу Романовскому, внимательно прочитавшему книгу и подготовившему обстоятельную рецензию.

Благодарю Александра Лукичева за многочисленные обсуждения книги. Его бескомпромиссность и стремление следовать только научному подходу вызывают глубокое уважение автора.

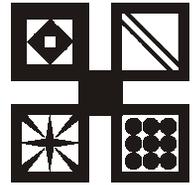
Игорю Некрестьянову, Александру Горшеневу, Денису Черноиванову и Алине Холчевой большое спасибо за прочтение черновых вариантов глав второго издания книги и высказанные замечания и комментарии.

Алексея Лопатина, дважды чемпиона мира по программированию в составе команды Санкт-Петербургского государственного университета, благодарю за помощь в подготовке материала для раздела "олимпиадное программирование".

Студентам пятого курса кафедр информатики и системного программирования выпуска 2004 года спасибо за конструктивный критический анализ первого издания.

Всех сотрудников филиала "Эльбрус МЦСТ. Санкт-Петербург" и лично его директора Игоря Калошина благодарю за прекрасную творческую атмосферу, которая помогала созданию этой книги.

*Игорь Одинцов  
Февраль 2004 г.*



# Глава 1

## Профессиональное программирование

В этой главе рассматриваются вопросы, над которыми рано или поздно задумываются все программисты.

- Каковы функции, черты и особенности мышления профессиональных программистов?
- Является программирование искусством, наукой, ремеслом или технологией?
- Что должны знать и уметь профессиональные программисты?
- Существуют ли стандарты для обучения программированию?
- Почему многие программные проекты выполняются с отставанием от графика, с превышением сметы расходов, при этом качество продукта не устраивает пользователя?

Также здесь приведены основные понятия и определения из области профессионального программирования.

### 1.1. Профессионализм в программировании

Каждое очередное пятилетие приносит нам новый взгляд на программирование. Экстраординарный промежуточный этап в решении задачи, математическая головоломка, дело необычайной трудности, доступное лишь посвященным, своеобразное инженерное конструирование, особого рода логическое рассуждение, наконец, основа любой целеустремленной деятельности, вторая грамотность современного образованного человека.

*А. П. Ершов*

Кто такой *профессионал*? Этим словом мы называем человека, который занимается каким-либо делом не просто как специалист, но и демонстрирует *профессионализм* — отличное владение своей профессией (в нашем случае —

программированием). Профессионализм — это интегральная личностная характеристика человека (программиста), который:

- овладел нормами профессиональной деятельности и общения и осуществляет их на высоком уровне, добиваясь профессионального мастерства в своей области (программировании);
- следует профессиональной ценностной ориентации, соблюдая профессиональную этику;
- развивает свою личность средствами профессии;
- стремится внести творческий вклад в профессию, обогащая ее опыт;
- стремится и умеет вызвать интерес общества к результатам своей профессиональной деятельности, способствует повышению веса и престижа своей профессии в обществе.

Образно можно заметить, что профессионализм делится на профессионализм деятельности и профессионализм личности. Профессионализм закладывается в процессе общего и профессионального образования, в отличие от мастерства, которое может быть накоплено в опыте и подражании. Природа профессионализма такова, что чем он выше, тем он *уже*.

Наряду с понятием "профессионализм" используют также понятия "образованность" и "компетентность". Образованность и компетентность являются неотъемлемыми составляющими профессионализма.

*Образованность* — это наличие разносторонних знаний. В отличие от профессионализма, образованность чем выше, тем шире. Высокообразованный программист — это не только безукоризненный специалист в определенной области программирования, но и человек, имеющий широкий взгляд на современное программирование независимо от языков и платформ, которые он сейчас использует. Это также человек, уверенно ориентирующийся в других сферах науки и искусства, знающий философию, историю, владеющий иностранными языками.

*Компетентность* — это осведомленность в целом ряде компонентов, имеющих надпрофессиональный или внепрофессиональный характер и в то же время абсолютно необходимых сегодня в той или иной степени каждому программисту. Компетентность проявляется в таких качествах личности, как самостоятельность, способность принимать ответственные решения, творческий подход к любому делу и умение довести его до конца, коммуникабельность, умение вести диалог. Просто незаменимыми качествами программиста являются умение постоянно учиться и обновлять свои знания, гибкость мышления и наличие абстрактного, системного и экспериментального мышления.

Результаты работы программистов играют важнейшую роль во многих областях человеческой деятельности. Программирование привлекает как

престижностью этой профессии, так и потенциально высокими заработками. Некоторые люди, написав небольшие программы и не встретив при этом серьезных трудностей, решают, что программирование вообще не очень сложное занятие. Итак, нужны ли программисты-профессионалы в настоящее время, когда каждый выпускник школы может написать программу?

А нужны ли профессиональные писатели, когда все сами грамотные? Ответ очевиден. И мы надеемся, что эта книга поможет вам стать *настоящим* образованным и компетентным профессионалом.

## 1.1.1. Функции, черты и особенности мышления профессиональных программистов

### 1.1.1.1. Функции профессиональных программистов

Математик делает то, что можно, так, как нужно.  
А программист — то, что нужно, так, как можно.

*Программистский фольклор*

Выделим основные функции профессиональных программистов [Цейтин 1990] [Буч, Рамбо, Джекобсон 2000].

- *Разработка систем большого объема.* Профессионалы знают, что "выход за некоторую границу сложности создаваемого программного продукта без надлежащей технологии разработки вообще невозможен". Системы большого объема создают новые принципиальные трудности. Интересно сравнение программирования больших систем со смоляными ямами доисторических времен, в которых увязали динозавры [Брукс 1999]: "Казалось, ничто в отдельности не вызывает трудностей — одну лапу всегда можно вытащить. Однако накопление действующих одновременно и взаимовлияющих факторов все более и более замедляет движение". Еще один образный пример приводится в книге по языку UML [Буч, Рамбо, Джекобсон 2000]: "Если вы хотите соорудить собачью конуру, то можете приступить к работе, имея в наличии лишь кучу досок, горсть гвоздей, молоток, плоскогубцы и рулетку... Хотя это и кажется комичным, многие компании, разрабатывающие программное обеспечение, хотят создать небоскреб, в то время как их подход к делу очень напоминает строительство собачьей конуры".
- *Создание системного и инструментального программного обеспечения,* которым будут пользоваться остальные. Профессиональными программистами создается инструментальное обеспечение в большом диапазоне — от профессионального до массового. Важная подзадача в этой области — создание языков, систем, интерфейсов, позволяющих пользователю работать в его собственных понятиях, а не в тех, которые навязаны

особенностями традиционного языка программирования или архитектурой компьютера.

- *Разработка программного продукта с предъявлением особых требований к его качеству и эффективности.* Различные требования предъявляются к программам, создаваемым только для себя, и к программам, предназначенным для большого количества пользователей. Во втором случае любая неаккуратность и малейшая небрежность может повториться многократно в работах будущих пользователей.
- *Разработка программ на основе подхода открытых систем.* Открытая система (open system) — система, которая состоит из компонентов, взаимодействующих друг с другом через стандартные интерфейсы. Общие свойства открытых систем таковы:
  - *мобильность* (переносимость) кода. Мобильность дает возможность легко переносить программное обеспечение на различные архитектурные и операционные платформы;
  - *интероперабельность.* Дает возможность легко осуществить обмен данными и взаимодействие с прикладными задачами в других системах;
  - *соответствие стандартам.* После того как стандарт принят, он становится силой, помогающей двигаться вперед. Как это ни парадоксально, но следование канонам дает новые степени свободы в современном мире.
- *Применение системного анализа,* включающего умение оценивать конкретные обстоятельства и следовать каждый раз наиболее приемлемому стилю. В голове программиста должны накопиться блоки знаний, чтобы он мог начать сравнивать эти блоки и мыслить блоками. Чтобы отличить структурное программирование от объектно-ориентированного программирования и оценить их пригодность для решения конкретных задач, надо иметь в голове и ту, и другую методологию.

### 1.1.1.2. Черты и особенности мышления программистов

Профессиональные программисты имеют свои черты, качества, способности и особенности мышления. Мы попытались сгруппировать наиболее важные из них, основываясь на работах [Цейтин 1990] [Дейкстра 1975] [Уэзерелл 1982] и собственном профессиональном опыте.

Первая группа черт и особенностей связана со способностью к абстрагированию и пониманию отношений между элементами.

- *Способность определить архитектуру программы,* т. е. разбить сложную задачу на элементарные составляющие и задать их комбинирование. Это основное качество программиста, связанное непосредственно с написанием программ.

- *Умение видеть задачу одновременно на разных уровнях детализации.* Программист должен свободно переходить от описания задачи в крупных понятиях к стоящим за этими понятиями сущностям более низкого уровня.
- *Умение представлять себе проектируемый процесс в динамике.* Обрабатываемые данные в некоторый момент времени могут иметь одни значения и взаимоотношения, а в следующий момент некоторые из них могут измениться.
- *Умение обобщать типичные ситуации.* Эта особенность означает, что необходимо уметь находить в программе идейно однородные участки. Для таких участков необходимо принять решение по их обобщению, выяснить его границы и выбрать наилучший способ программной реализации найденного решения. Это может быть как разработка функции или пакета, так и копирование в новую программу фрагментов из старой.
- *Умение применять и комбинировать хорошо известные программистские приемы и типовые алгоритмы.* Большинство новых идей должно находиться в тесном взаимодействии с уже известными идеями и методами. Простые и изящные решения типичных задач, возникающих в объектно-ориентированном программировании, приведены в книге [Гамма, Хелм, Джонсон, Влссидес 2001].
- *Владение интеллектуальными средствами,* которые применяются для понимания программы. Эдсгер Дейкстра (Edsger Dijkstra) выделяет следующие три основных средства, которыми должен обладать профессиональный программист [Дейкстра 1975]:
  - *абстракция.* Абстракция проникла во все аспекты программирования. Алгоритм — это то, что остается от конкретных значений обрабатываемых данных. Переменная — абстракция соответствующего текущего значения. Абстракция присутствует в процессе присваивания операциям имен и использования этих операций. Язык программирования — это лишь средство описания абстрактных конструкций. Программист должен иметь способность полностью абстрагироваться от несущественных деталей, думая на нескольких уровнях абстракции одновременно;
  - *перечисление.* К перечислению прибегают, когда пытаются проверить правильность вычислений, которые сводятся к обозримому множеству последовательно выполняемых операторов. Сюда также включаются условные операторы выбора одного из двух или более вариантов;
  - *математическая индукция.* Это единственный способ рассуждений, который в конечном итоге позволяет работать с циклами и с рекурсивными процедурами.