

Высшее профессиональное образование

---

БАКАЛАВРИАТ

Н. В. АНАШКИНА, Н. Н. ПЕТУХОВА,  
В. Ю. СМОЛЬЯНИНОВ

# ТЕХНОЛОГИИ И МЕТОДЫ ПРОГРАММИРОВАНИЯ

*Допущено*

*Учебно методическим объединением*

*по образованию в области информационной безопасности*

*в качестве учебного пособия для студентов*

*высших учебных заведений, обучающихся*

*по направлению подготовки 090900 «Информационная безопасность»*

*(уровень — бакалавр) и специальностям 090301*

*«Компьютерная безопасность», 090303 «Информационная*

*безопасность автоматизированных систем»*



Москва

Издательский центр «Академия»

2012

УДК 681.3.06(075.8)  
ББК 32.973-018я73  
А 64

Рецензенты:

профессор кафедры «Информационная безопасность» МИРЭА(ТУ),  
канд. физ.-мат. наук *В. П. Зязин*; заведующий кафедрой защиты информации  
и криптографии ТГУ, д-р техн. наук, профессор *Г. П. Агибалов*

**Анашкина Н. В.**

А64 Технологии и методы программирования: учеб. пособие для студ. учреждений высш. проф. образования / Н. В. Анашкина, Н. Н. Петухова, В. Ю. Смольянинов. — М.: Издательский центр «Академия», 2012. — 384 с. — (Сер. Бакалавриат).  
ISBN 978-5-7695-8429-9

Учебное пособие создано в соответствии с Федеральным государственным образовательным стандартом по направлению подготовки 090900 «Информационная безопасность» (квалификация «бакалавр»).

Рассмотрены основы технологий создания программных продуктов, типы и структуры данных, используемые в повседневной практике программирования.

Приведены алгоритмы решения наиболее распространенных классов задач: алгоритмы сортировки и поиска, алгоритмы генерации комбинаторных объектов.

Для студентов, обучающихся по специальностям в области информационных технологий, а также аспирантов и преподавателей вузов.

УДК 681.3.06(075.8)  
ББК 32.973-018я73

*Оригинал-макет данного издания является собственностью  
Издательского центра «Академия», и его воспроизведение любым  
способом без согласия правообладателя запрещается*

© Анашкина Н. В., Петухова Н. Н., Смольянинов В. Ю.,  
2012

© Образовательно-издательский центр «Академия», 2012

© Оформление. Издательский центр «Академия», 2012

ISBN 978-5-7695-8429-9

Любая профессиональная деятельность эффективна при условии успешного освоения технологий в соответствующей сфере. Разработка программных систем в этом смысле не исключение. Квалифицированный разработчик программных средств должен владеть современными технологиями их создания, знать несколько языков программирования, уметь выбирать для решения конкретной задачи подходящие структуры данных и алгоритмы.

Учебное пособие «Технологии и методы программирования» разработано в соответствии с требованиями федерального государственного образовательного стандарта (ФГОС) высшего профессионального образования для бакалавров по направлению подготовки «Информационная безопасность» и специалистов по специальностям «Компьютерная безопасность», «Информационная безопасность телекоммуникационных систем», «Информационная безопасность автоматизированных систем», «Информационно-аналитические системы безопасности» и другим в области информационных технологий.

Процесс изучения дисциплины по направлению подготовки 090900 «Информационная безопасность» направлен на формирование у бакалавров, в числе многих других, следующих компетенций: ОК-8, ПК-2, ПК-6, ПК-12, ПК-15, ПК-16, ПК-17, ПК-20, у специалистов по специальности 090301 «Компьютерная безопасность» — ОК-9, ПК-2, ПК-3, ПК-7, ПК-8, ПК-9, ПК-12, ПК-28, ПК-35, по специальностям 090302 «Информационная безопасность телекоммуникационных систем» и 090303 «Информационная безопасность автоматизированных систем» — ОК-9, ПК-2, ПК-3, ПК-4, ПК-10, ПК-12, по специальности 090305 «Информационно-аналитические системы безопасности» — ОК-9, ПК-2, ПК-3, ПК-4, ПК-13, ПК-15, ПК-18, ПК-24, ПК-25.

Для реализации перечисленных компетенций в результате изучения дисциплины «Технологии и методы программирования» студент должен:

- знать современные технологии программирования, базовые структуры данных и классические алгоритмы;
- уметь выбирать необходимые технологии разработки программных средств и эффективные алгоритмы решения прикладных задач.

Материал, изложенный в пособии, также будет полезен при изучении широкого круга других дисциплин, таких как «Информатика»,

«Структуры данных и прикладные алгоритмы», «Методы программирования» и «Технология программирования», предусмотренных ФГОС перечисленных специальностей подготовки бакалавров, специалистов и магистров.

Необходимо отметить, что большое количество разрозненной литературы по тематике дисциплины затрудняет изучение предусмотренного ФГОС объема материала за время, отведенное программами обучения. Основной задачей, стоящей перед авторами, являлось формирование единого пособия, охватывающего все темы учебного курса. Книга предназначена для первичного изучения вопросов технологий и методов программирования. Она будет интересна широкому кругу студентов, обучающихся в области информационных технологий, и полезна преподавателям. Объем материала пособия и глубина его изложения выбраны с учетом требуемого ФГОС уровня подготовки специалистов, магистров и бакалавров по направлению «Информационная безопасность». При обучении бакалавров можно опускать доказательства корректности и выводы вычислительной сложности рассматриваемых алгоритмов, а также сократить число изучаемых тем с учетом особенностей специальностей.

Пособие объединяет в себе вопросы технологий программирования, связанные с эффективным управлением программными проектами, и вопросы, связанные с выбором структур данных для построения эффективных алгоритмов. Все изложенные в пособии алгоритмы для лучшего понимания разобраны на конкретных примерах и для наглядности проиллюстрированы соответствующими рисунками. Приведены примеры реализации алгоритмов на C++. При анализе алгоритмов используются различные подходы к доказательству корректности их работы и разнообразные методы получения оценок вычислительной сложности.

Учебное пособие состоит из пяти глав.

В главе 1 рассмотрены технологии разработки программных продуктов. В ней описаны этапы создания программного обеспечения в соответствии с международными и государственными стандартами. Рассмотрены вопросы, связанные с жизненным циклом программного обеспечения, качеством и методами его контроля. Ограниченный объем пособия не позволил детально рассмотреть все этапы разработки программных средств, но в главе приведены ссылки на специальную литературу по каждому вопросу.

Глава 2 посвящена анализу алгоритмов и структурам данных. Рассматриваются подходы к анализу алгоритмов и оценке их вычислительной сложности. На примере показана пошаговая модификация алгоритма, приводящая к более эффективной его реализации. Рассмотрены наиболее часто используемые динамические структуры данных, такие как списки, очереди, стеки, деки и деревья.

В главе 3 приводится классификация и основные понятия алгоритмов сортировки, подробное описание их работы, сравнительный

анализ их вычислительной сложности и рекомендации по использованию. Глава содержит алгоритмы сортировки сравнениями (вставками, выбором, обментами и слиянием), распределяющие сортировки и основные подходы к реализации внешней сортировки.

Глава 4 включает в себя описание более сложных структур данных, используемых в алгоритмах поиска. Приведены различные подходы к классификации алгоритмов. Обсуждаются последовательные алгоритмы, алгоритмы, использующие древовидные структуры, хеширование. Рассмотрено большое количество типов деревьев, предназначенных для организации эффективного поиска. В разделе хеширования описаны методы разрешения коллизий и идеальное хеширование. Приведен подробный анализ вычислительной сложности рассмотренных в главе алгоритмов.

Глава 5 содержит описание алгоритмов генерации ряда комбинаторных объектов. К ним относятся алгоритмы порождения перестановок, подмножеств множества, комбинаций и разбиений целых чисел. Рассмотрены алгоритмы генерации комбинаторных объектов в лексикографическом порядке и в порядке минимального изменения.

В конце глав пособия приведены вопросы для самоконтроля и упражнения, выполнение которых полезно для успешного усвоения теоретического материала.

Авторы будут благодарны читателям, которые обнаружат в этом пособии ошибки или упущения и сообщат об этом через издательство.

Не верь ничему, где бы ты это ни прочел и кто бы ни сказал это, даже если это сказал я, — если это не согласуется с твоим рассудком и чувством здравого смысла.

Будда

### 1.1. Проблемы разработки программного обеспечения

В настоящее время информационные технологии проникли практически во все сферы человеческой деятельности. Широкое использование *средств вычислительной техники (СВТ)* и *программного обеспечения (ПО)* существенно упростило накопление, хранение и обработку информации. Все больше технических устройств включают в себя элементы СВТ и соответствующего управляющего ПО, которое, зачастую, составляет большую часть их стоимости. За последние десятилетия рынок ПО превратился в движущую силу экономического развития. Существование практически всех отраслей мировой экономики немислимо без использования ПО.

Бурное развитие СВТ привело к росту потребностей в создании ПО и методологической базы управления разработкой программных проектов.

В 1960-х — начале 1970-х гг. появился первый отрицательный опыт создания больших программных проектов: сроки выполнения работ задерживались, затраты могли в несколько раз превосходить сметные оценки, готовые системы в большинстве случаев имели низкие показатели качества и производительности, кроме того, созданное ПО часто не соответствовало своему целевому назначению [49]. И все это при том, что разработкой занимались высококвалифицированные специалисты. Причиной неудач большинства проектов явилось использование неформальных подходов, основанных на опыте управления техническими проектами. Возникла необходимость в новых технологиях и методах управления программными проектами, которые увеличили бы продуктивность, надежность и качество разработки.

Результатом накопления опыта создания ПО стало появление новой инженерной дисциплины — *технологии программирования (инженерии программного обеспечения, software engineering)*, в рамках которой изучаются производственные процессы, приводящие к созданию ПО. Необходимо отметить, что устоявшееся название дисциплины представляется несколько неудачным, поскольку существует множество конкретных *технологий* программирования, которые предлагают отличающиеся подходы к организации производственного процесса. Несмотря на имеющиеся отличия, всем существующим технологиям присуща общность, в связи с чем правомерно говорить о технологии программирования вообще [17].

В рамках дисциплины рассматриваются не технические аспекты создания ПО, а лишь вопросы, связанные с организацией эффективного управления программными проектами, а также с определением принципов, моделей и методов, используемых в инженерном цикле производства ПО. Рассмотрение вопросов создания ПО находится за рамками данного пособия. Для получения дополнительной информации по этим вопросам авторы рекомендуют обратиться к работе [34], являющейся прекрасным введением в разработку ПО.

Специалисты в области разработки ПО адаптируют существующие методы инженерии к решению собственных задач. В процессе разработки им приходится решать традиционные инженерные вопросы обеспечения качества и надежности при наличии ограничений по времени создания и стоимости продукта.

Многие ошибочно отождествляют термин «программное обеспечение» с компьютерными программами. Однако согласно ГОСТ 19.781 — 90 *программное обеспечение* — это совокупность программ системы обработки информации и программных документов, необходимых для эксплуатации этих программ. В приведенное определение помимо программ и сопутствующей документации следует включить конфигурационные данные, необходимые для корректной работы программ.

Несмотря на все достигнутые успехи в области создания ПО, все еще существует ряд нерешенных проблем, обусловленных следующими факторами [29, 49, 66]:

1. **Рост объема и сложности ПО.** Непрерывное повышение сложности функций, реализуемых ПО, приводит к увеличению его объема и трудоемкости создания. С ростом сложности программ возрастает и число выявляемых и остающихся в ПО дефектов и ошибок, что отражается на итоговом качестве продукта.

2. **Недостаток прозрачности и контроля.** Разработка программ отличается от других инженерных видов деятельности тем, что в основном состоит из проектирования. В силу абстрактности ПО сложно сказать, в каком состоянии находится проект и каков процент его завершенности. Для получения ответов на эти вопросы менеджер проекта вынужден полагаться на мнение разработчиков и докумен-

тацию, фиксирующую этапы процесса разработки. Без точной оценки прогресса разработки срываются графики выполнения работ и превышает установленный бюджет.

**3. Изменение требований.** В процессе разработки у пользователей постоянно возникают новые идеи относительно создаваемой системы. Влияние изменений может быть существенным для успеха проекта, поэтому важно уметь оценивать предлагаемые изменения и реализовывать только одобренные, контролируя этот процесс с использованием инструментальных средств.

**4. Недостаточная надежность ПО.** Поскольку число ошибок в программах заранее неизвестно, то заранее неизвестна продолжительность отладки и тестирования программ. Как следствие, не существует гарантий отсутствия дефектов в программах.

**5. Отсутствие стандартных процессов разработки ПО.** На сегодняшний день отсутствуют целостные подходы, годящиеся для создания всех видов программных продуктов. Даже для известных и широко используемых процессов создания ПО недостаточно четко определена область их применимости. Инженерия ПО в настоящий момент находится в стадии становления: каждый раз менеджеру проекта приходится только на основании своего опыта и советов экспертов принимать решение о том, какой процесс разработки использовать и как его модифицировать для достижения большей эффективности в конкретном проекте.

**6. Уникальность больших программных проектов.** Большие программные проекты, как правило, значительно отличаются от проектов, реализованных ранее. Во время создания крупных систем часто возникают новые уникальные задачи, требующие творческого подхода. Кроме того, постоянные изменения в информационных технологиях и методологиях разработки ПО обесценивают ранее накопленный исполнителями опыт.

В рамках относительно небольшого раздела пособия не представляется возможным рассмотреть подробно весь спектр вопросов, связанных с созданием ПО. Для их углубленного изучения читателю следует обратиться к специализированной литературе, ссылки на которую приводятся в соответствующих разделах.

## 1.2. Жизненный цикл программного обеспечения

Согласно стандарту IEEE 1471 *программная система (ПС)* — это любая система, в которой ПО оказывает значительное влияние на ее проект, конструкцию, развертывание и дальнейшее развитие. Под *жизненным циклом (ЖЦ)* понимают весь период времени существования ПС, начиная с момента принятия решения о необходимости



сти создания и заканчивая моментом прекращения всех видов ее использования [17, 49].

Совокупность процессов, приводящих к созданию ПО, называется **процессом создания ПО**. Выделяют следующие фундаментальные процессы создания ПО [49]:

1. **Разработка требований.** Выработка требований, определяющих функциональные характеристики создаваемой системы.

2. **Создание.** Разработка ПО согласно спецификации. Данный процесс включает в себя проектирование, реализацию, интеграцию и отладку системы.

3. **Аттестация.** Прохождение ПО аттестации для подтверждения его соответствия требованиям заказчика.

4. **Эксплуатация и сопровождение.** Модернизация ПО для удовлетворения изменившихся требований и исправления выявленных в ходе эксплуатации недостатков.

Перечисленные процессы реализуются с использованием технологий инженерии программного обеспечения. При этом создаются и перерабатываются различного рода *артефакты* — информационные сущности, участвующие в качестве входных данных и получающиеся в качестве выходных результатов процессов. Примерами артефактов являются: модель предметной области, описание требований, техническое задание, архитектура системы, проектная документация на систему в целом и ее компоненты, прототипы системы и компонент, исходный код, программа и методика испытаний, пользовательская документация, документация администратора системы, руководство по развертыванию и пр.

В настоящий момент в Российской Федерации одновременно действуют две категории стандартов, относящихся к разработке ПС: унаследованные стандарты, созданные в период до 1990 г., и современные национальные стандарты Российской Федерации (ГОСТ Р), являющиеся прямыми аналогами международных стандартов ISO. Работа по разработке аналогов международных стандартов продолжается и в настоящее время. Унаследованные стандарты часто применяются при разработке систем, создаваемых для органов государственной власти. Следует отметить, что стандарты в области ПС носят рекомендательный характер. В соответствии с Законом РФ от 10 июня 1998 г. № 5154-1 «О стандартизации» (с изм. и доп.) эти стандарты становятся обязательными при наличии ссылки на них в договоре на разработку.

Наиболее важным унаследованным комплексом стандартов является **единая система программной документации (ЕСПД)** — набор государственных стандартов, устанавливающих взаимосвязанные правила разработки, оформления и обращения программ и программной документации. ЕСПД принята и разработана в период с 1977 по 1990 г. Стандарты, входящие в ЕСПД, имеют номер серии 19.

ЕСПД вносит упорядоченность в процесс документирования ПО, определяя примерный состав документации. При гибком подходе к

использованию ЕСПД заказчик и руководитель проекта могут выбирать уместное в проекте подмножество стандартов и программных документов, дополнив их необходимыми разделами и исключив ненужные.

В состав ЕСПД входит ГОСТ 19.102—77 «Единая система программной документации. Стадии разработки», устанавливающий стадии разработки программ и программной документации для вычислительных машин и комплексов. Стадия — часть процесса создания ПО, ограниченная временными рамками и заканчивающаяся созданием определенных артефактов. Стандарт также определяет этапы реализации стадий и содержание работ, выполняемых в рамках этапов.

Выделяют следующие стадии разработки: техническое задание (ТЗ), эскизный проект, технический проект, рабочий проект и внедрение. На стадии технического задания производится разработка требований к создаваемой системе, фиксируемых в документе, называемом «техническим заданием». На стадии эскизного проекта создается пояснительная записка, содержащая общее описание алгоритма работы программы, а также обоснование принятых технико-экономических решений. На стадии технического проекта осуществляется детальное проектирование системы, составляются планы по ее разработке и внедрению. В ходе реализации рабочего проекта выполняются программирование и отладка, разработка программных документов, в том числе программы и документации, содержащей требования, подлежащие проверке при аттестации, а также проводятся испытания и соответствующая корректировка программ. На стадии внедрения производится передача ПО для дальнейшего сопровождения, а также оформление акта передачи.

Допускается исключение стадии эскизного проекта, а в технически обоснованных случаях и стадии технического проектирования. Стандарт допускает объединение, исключение этапов работ и (или) их содержания, а также введение других этапов по согласованию с заказчиком.

К числу основных недостатков ЕСПД можно отнести:

- ориентацию на последовательную реализацию стадий;
- отсутствие четких рекомендаций по документированию характеристик качества ПС;
- отсутствие системной увязки с другими действующими отечественными системами стандартов ЖЦ и документированием продукции в целом;
- отсутствие в явном виде сквозных процессов, реализуемых на различных стадиях создания ПО (например, контроль качества, управление конфигурацией).

Стандарты серии 34 задумывались в конце 1980-х гг. как всеобъемлющий комплекс взаимоувязанных межотраслевых стандартов. Согласно ГОСТ 34.003—90 *автоматизированная система (АС)* —

это система, состоящая из персонала и комплекса средств автоматизации его деятельности, реализующая информационную технологию выполнения установленных функций. Объектами стандартизации являются АС любых видов, а не только АС, построенные с использованием программных технологий. Частично стандарты серии 34 уточняют применение ЕСПД в отношении автоматизированных систем.

Согласно ГОСТ 34.601 — 90 «Автоматизированные системы. Стадии создания» *процесс создания автоматизированной системы* представляет собой совокупность *упорядоченных во времени*, взаимосвязанных, объединенных в стадии и этапы работ, выполнение которых необходимо и достаточно для создания АС, соответствующей заданным требованиям. Стандарт предусматривает следующие стадии и этапы создания АС: формирование требований к АС, разработка концепции АС, техническое задание, эскизный проект, технический проект, рабочая документация и ввод в действие.

В стандарте приводится примерное содержание работ каждого этапа. Допускается исключение стадии «Эскизный проект» и отдельных этапов работ на всех стадиях, объединение стадии «Технический проект» и «Рабочая документация» в «Технорабочий проект», а также включение новых дополнительных этапов. В отличие от ранее рассмотренного стандарта ГОСТ 19.102 — 77, более детально определяется процесс разработки требований и аттестации, кроме того в явном виде вводится этап сопровождения АС.

По согласованию с заказчиком в программную документацию могут вводиться дополнительные разделы и создаваться дополнительные документы. Предусматривается возможность создания *частных технических заданий (ЧТЗ)* для реализации крупных структурных единиц (подсистем, комплексов), что позволяет более гибко формировать ЖЦ АС. Следует заметить, что согласно ГОСТ 34.602 — 89 ТЗ разрабатывает исполнитель, но формально ТЗ выдает разработчику заказчик (согласно РД 50-680 — 88).

По сравнению с ранее рассмотренным стандартом ГОСТ 19.102 — 77 ГОСТ 34.602 — 89 вводит большую детализацию стадий, в целом сохраняя структурное соответствие ЕСПД. Следует отметить, что стандарты серии 34 сохранили основные недостатки ЕСПД.

Несмотря на несколько отличающуюся терминологию, стадии ГОСТ 19.102 — 77 и ГОСТ 34.601 — 90 соответствуют перечисленным выше фундаментальным процессам создания ПО. Подход к организации процесса создания ПО в виде стадий принуждает разработчиков к последовательному выполнению работ. Во многих случаях применение подобного подхода представляется затруднительным.

Современные стандарты устраняют указанный недостаток, рассматривая создание ПО как набор процессов, которые могут протекать параллельно во времени.

Базовый стандарт ГОСТ Р ИСО/МЭК 12207 — 99 «Информационная технология. Процессы жизненного цикла программных средств»

основан на зарубежном стандарте ISO/IEC 12207—95 «Information Technology — Software Life Cycle Processes»<sup>1</sup>.

Данный стандарт задает структуру ЖЦ в виде набора процессов, состоящих из множества работ. Работы, в свою очередь, состоят из набора задач. Стандарт не диктует predetermined последовательность выполнения процессов. Более того, определяется процесс адаптации, который состоит в исключении неприменяемых в условиях конкретного программного проекта процессов, работ и задач.

Стандарт задает деление процессов на три категории: основные, вспомогательные и организационные.

*К основным процессам* относятся:

1. **Заказ.** Работы заказчика по идентификации его потребности в ПО, подготовке заявки на подряд, выбору поставщика и контролю над процессом выполнения заказа вплоть до завершения этапа приемки.

2. **Поставка.** Работы поставщика по подготовке предложения в ответ на заявку заказчика или подписанию договора по поставке ПО, а также по определению требуемых для поставки ресурсов, разработке проектных планов и выполнению поставки.

3. **Разработка.** Работы разработчика по анализу требований, проектированию, программированию, сборке, тестированию, вводу в действие и приемке созданного ПО.

4. **Эксплуатация.** Работы по поддержке пользователей в процессе эксплуатации ПО.

5. **Сопровождение.** Работы по анализу возникающих проблем; внесению контролируемых изменений в ПО с целью его модернизации, адаптации и улучшения функционирования при сохранении работоспособности и функциональных возможностей; экспертизе и передаче измененного ПО; изъятию ПО из эксплуатации.

*Вспомогательными процессами* являются:

1. **Документирование.** Работы по планированию, проектированию, разработке, выпуску, редактированию, распространению и сопровождению документов, в которых нуждаются все заинтересованные лица.

2. **Управление конфигурацией.** Работы по идентификации конфигурации, учету и хранению изменений, управлению сборкой ПО, учету текущего состояния программных объектов и поставок ПО.

3. **Обеспечение качества.** Процесс предназначен для обеспечения гарантий того, что программные продукты и процессы в ЖЦ проекта соответствуют установленным требованиям и утвержденным планам. Данный процесс включает в себя работы по обеспечению качества продукта, процесса, а также работы по обеспечению качества в соответствии со стандартом ГОСТ Р ИСО 9001.

<sup>1</sup> Существует более новая версия данного стандарта ISO/IEC 12207:2008.