


Технологии программирования

В процессе разработки программных систем используются различные технологии программирования. В соответствии с обычным значением слова "технология" под технологией программирования (programming technology) понимается совокупность производственных процессов, приводящая к созданию требуемой ПС, а также описание этой совокупности процессов.




*Другими словами,
технология программирования понимается здесь в
широком смысле как технология разработки
программных средств, включая в нее все процессы,
начиная с момента зарождения идеи этого
средства до создания необходимой программной
документации. Каждый процесс этой совокупности
базируется на использовании каких-либо методов и
средств, например, компьютера
(в этом случае речь идет о компьютерной
технологии программирования).*

В литературе имеются и другие, несколько отличающиеся, определения технологии программирования. Используется в литературе и близкое к технологии программирования понятие программной инженерии, определяемой как систематический подход к разработке, эксплуатации, сопровождению и изъятию из обращения программных средств.

Главное различие между технологией программирования и программной инженерией как дисциплинами для изучения заключается в способе рассмотрения и систематизации материала.

В технологии программирования акцент делается на изучении процессов разработки **ПС** (*технологических процессов*)


и порядке их прохождения:
методы и инструментальные средства разработки **ПС** задействуются в этих процессах, их применение и образует технологические процессы. В программной инженерии изучаются различные методы и инструментальные средства разработки **ПС** с точки зрения достижения определенных целей – эти методы и средства могут использоваться в разных технологических процессах (*и в разных технологиях программирования*).



Не следует также путать технологию программирования с методологией программирования.

В **технологии** программирования методы рассматриваются "**сверху**" – *с точки зрения организации технологических процессов,*
а в **методологии** программирования методы рассматриваются "**снизу**" – *с точки зрения основ их построения.*

Например, в работе методология программирования определяется как совокупность механизмов, применяемых в процессе разработки программного обеспечения и объединенных одним общим философским подходом.



***В историческом аспекте в развитии
технологии программирования можно
выделить несколько этапов...***

1. Первый этап: "стихийное"
программирование – *отсутствие сформулированной технологии, когда программирование было, по сути, искусством.*

Этап охватывает период от появления первых **ЭВМ** до середины **60-х годов 20-го века.**

*Развитие программирования шло по пути замены машинных языков **ассемблерами**, а затем алгоритмическими языками (**Fortran, Algol**)*

и повторного использования

Стихийно использовалась разработка
"снизу вверх" – подход, при котором
вначале проектировали и
реализовали сравнительно простые
подпрограммы, из которых потом
пытались построить сложную
программу.

В начале 60-х годов 20-го века
разразился кризис
программирования .

Он выражался в том, что фирмы
превышали все сроки завершения
программных проектов и их
стоимость.

2. Второй этап – структурный подход к программированию.

Этот подход сложился в 60-70-е годы 20-го века и представлял собой совокупность рекомендуемых технологических приемов, охватывающих все этапы разработки программного обеспечения.

В основе структурного подхода лежит декомпозиция сложных систем с целью последующей реализации в виде отдельных небольших подпрограмм.

В отличие от используемого ранее процедурного подхода к декомпозиции, структурный подход требовал представления задачи в виде иерархии

Проектирование осуществлялось "сверху вниз" и подразумевало реализацию общей идеи, обеспечивая проработку интерфейсов подпрограмм.

*Вводились ограничения на конструкции алгоритмов, рекомендовались формальные модели их описания, а также специальный метод проектирования **алгоритмов** – метод пошаговой детализации.*

Поддержка принципов структурного программирования была заложена в основу процедурных языков

Появилась и начала развиваться технология модульного программирования, которая предполагает выделение групп подпрограмм, использующих одни и те же **глобальные данные**,

в отдельно компилируемые модули. *Практика показала, что структурный подход в сочетании с модульным программированием позволяет получить достаточно надежные программы, размер которых не превышает **100000 операторов**.*

Узким местом модульного программирования стали межмодульные интерфейсы, ошибки в которых трудно обнаружить по причине параллельной компиляции модулей

3. Третий этап – *объектный подход к программированию.*

Сложился с середины **80-х** до конца **90-х** годов **20-го** века.

Объектно-ориентированное программирование (ООП) *определяется как технология создания сложного программного обеспечения, основанная на представлении программы в виде совокупности объектов, каждый из которых является экземпляром определенного типа (класса), а классы образуют иерархию с наследованием свойств.*

Основное достоинство объектно-ориентированного программирования

по сравнению с модульным
программированием – более естественная
декомпозиция программного
обеспечения, которая существенно
облегчает его разработку.

Кроме того, объектный подход предлагает
новые способы организации программ,
основанные на **механизмах
наследования, полиморфизма,
композиции.**

*Это позволяет существенно увеличить
показатель повторного использования
кодов и создавать библиотеки классов для*

Развитие объектного подхода в технологии программирования привело к созданию сред визуального программирования.

Появились языки визуального объектно-ориентированного программирования, такие как Delphi, C++ Builder, Visual C++, C# и т. д.

Однако технология ООП имеет и недостатки.

Главный из них – зависимость модулей программного обеспечения от адресов экспортируемых полей и методов, структур и форматов данных.

Эта зависимость объективна, так как модули должны взаимодействовать между собой,

4. Четвертый этап – компонентный подход и *CASE-технологии* (с середины **90-х годов 20-го века** до нашего времени) .

Этот подход предполагает построение программного обеспечения из отдельных компонентов – физически отдельно существующих частей программного обеспечения, которые взаимодействуют между собой через стандартизованные двоичные интерфейсы. В отличие от обычных объектов объекты-компоненты можно собирать в динамически вызываемые библиотеки или исполняемые файлы, распространять в двоичном виде (**без исходных текстов**) и использовать в любом языке программирования, поддерживающем соответствующую технологию. В настоящее время рынок компонентов – реальность, поддерживаемая Интернетом и массовой рекламой и публикациями.

Основы компонентного подхода были разработаны компанией **Microsoft**, начиная с технологии **OLE** (*Object Linking and Embedding* – связывание и внедрение объектов), которая применялась в ранних версиях **Windows** для создания составных документов.

Ее развитием стало появление **COM**-технологии (*Component Object Model* – компонентная модель объектов), а затем ее распределенной версии – **DCOM**, на основе которых были разработаны **компонентные технологии**, решаются различные задачи разработки программного обеспечения.

Среди них следуют отметить **OLE-automation** – технологию создания программируемых приложений, обеспечивающую доступ к внутренним службам этих приложений. На основе **OLE-automation** создана технология **ActiveX**, предназначенная для создания программного обеспечения, как сосредоточенного на одном компьютере, так и распределенного.

Безопасность и стабильная работа распределенных приложений обеспечивается еще двумя технологиями, заложенными в **COM**. Это **MDS (Multitier Distributed Application Sever)** – сервер многозвенных распределенных приложений, и **MTS (Microsoft Transaction Server)** – сервер управления транзакциями.

Компонентный подход лежит также в основе
технологии

CORBA (*Common Object Request Bracer Architecture* –
общая архитектура с посредником обработки
запросов объектов).

Эта технология, которая реализует подход,
аналогичный **COM**, разработана группой
компаний **OMG** (*Object Management Group* –
группа внедрения объектной технологии
программирования).


Программное ядро **CORBA** реализовано для всех
основных аппаратных и программных платформ
и обеспечивает создание программного
обеспечения в гетерогенной вычислительной
среде.

Важнейшая особенность современного этапа технологии программирования – широкое использование компьютерных технологий создания и сопровождения программных систем на всех этапах их жизненного цикла. Эти технологии получили название

CASE-технологий (*ComputerAided Software/System engineering* – разработка программного обеспечения/программных систем с использованием компьютерной поддержки).

Сегодня существуют

CASE-технологии, поддерживающие как структурный, так и объектный, в том числе компонентный, подходы к программированию .



*Зная, какие программные технологии оказали самое сильное влияние за последние **25 лет**, можно оценить их по количеству опубликованных научных статей или выяснить, например, сколько времени продолжался поднятый вокруг них ажиотаж. С другой стороны, с точки зрения пользователя, можно судить о них, оценивая, чего удалось добиться по сравнению с тем, что предварительно было обещано. Так в чем же уникальность последних **25 лет** ?*

Во-первых, программное обеспечение перестало быть прерогативой нескольких компаний и стало частью повседневной жизни практически каждого жителя нашей планеты:

ПК, Internet и мобильные телефоны – свидетельства этой грандиозной эволюции. На [рис. 1.1](#) показаны программные технологии и периоды, когда они достигли важнейших этапов своего развития. Эта таблица построена исходя из структуры, предложенной

Сэмом Редвайном и **Вильямом Риддлом**.

Для простоты показаны только три этапа развития: *основы – когда проведены базовые исследования и созданы краеугольные концепции, ограниченное использование – когда эти концепции были взяты на вооружение некоторыми компаниями и пользователями, широкое использование – когда технология стала применяться примерно третьей частью целевого рынка.*