

Тема 6

Программное обеспечение и
технологии программирования

Вопросы к теме 6

1. Системное программное обеспечение, его классификация.
2. Прикладное программное обеспечение, его классификация.
3. Жизненный цикл программного обеспечения.
4. Технологии программирования.

Вопрос 1. Системное программное обеспечение, его классификация

Программное обеспечение (ПО) — это совокупность программ регулярного применения, необходимых для решения задач пользователя, и программ, позволяющих наиболее эффективно использовать вычислительную технику, обеспечивая пользователям наибольшие удобства в работе и минимум затрат труда на программирование задач и обработку информации.

- В настоящее время весь комплекс ПО делится на системные (системное программное обеспечение) и пользовательские программы (прикладное программное обеспечение).
- Системное программное обеспечение выполняет функции «организатора» всех частей ПК, а также подключенных к нему внешних устройств.
- Прикладное программное обеспечение служит для выполнения каких – либо конкретных задач во всех сферах человеческой деятельности.

Системное программное обеспечение – совокупность программ и программных комплексов для обеспечения работы компьютера и сетей ЭВМ.

- Данный класс программных продуктов тесно связан с типом компьютера и является его неотъемлемой частью, а также они ориентированы в основном на квалифицированных пользователей.

Системное программное обеспечение направлено:

- на создание операционной среды функционирования других программ;
- на обеспечение надежной и эффективной работы самого компьютера и вычислительной сети;
- на проведение диагностики и профилактики аппаратуры компьютера и вычислительных сетей;
- на выполнение вспомогательных технологических процессов (копирование, архивирование, восстановление файлов программ и баз данных и т.д.)

Системное программное обеспечение делится на 3 класса:

1. базовое программное обеспечение, или BIOS – программа, которая отвечает за управление всеми компонентами, установленными на материнской плате и предназначена для поддержки функций ввода – вывода; для тестирования оборудования, установленного на материнской плате; для загрузки операционной системы.

2. **операционные системы** – программы, *управляющие* работой всех устройств компьютерной системы, *организующие* взаимодействие отдельных процессов, протекающих в компьютере во время выполнения программ, в частности, отображение информации на дисплее, диалог на ограниченном естественном языке, *перевод* на машинный язык программ, написанных на языках программирования;
3. **сервисное программное обеспечение (служебные программы)** – отладчики, архиваторы, антивирусники и др.

- **Операционная система** предназначена для управления выполнением пользовательских программ, планирования и управления вычислительными ресурсами ЭВМ.
- Существуют следующие виды операционных систем: Windows 98, Windows XP Professional, Windows Home Edition, Windows 7, Linux, Unix и др.

- Операционные системы для персональных компьютеров делятся на:
 - – одно- и многозадачные (в зависимости от числа параллельно выполняемых прикладных процессов);
 - – одно- и многопользовательские (в зависимости от числа пользователей, одновременно работающих с операционной системой);
 - – непереносимые и переносимые на другие типы компьютеров;
 - – несетевые и сетевые, обеспечивающие работу в локальной вычислительной сети ЭВМ.

- **Сетевые операционные системы** – это комплекс программ, обеспечивающий обработку, передачу и хранение данных в сети.
- Сетевая ОС предоставляет пользователям различные виды сетевых служб (управление файлами, электронная почта, процесс управления сетью и др.), поддерживает работу в абонентских системах.
- Наибольшее распространение имеют LAN Server, NetWare, VINES.

СЕРВИСНОЕ программное обеспечение -

программы и программные комплексы, которые расширяют возможности базового программного обеспечения и организуют более удобную среду работы пользователя.

- Это набор сервисных, дополнительно устанавливаемых программ, которые можно классифицировать по функциональному признаку следующим образом:

- **драйверы специфических и специальных устройств** (те, которые не поставляются в составе ОС).
- **программы диагностики работоспособности компьютера;**
- **антивирусные программы,** обеспечивающие защиту компьютера, обнаружение и восстановление зараженных файлов;

- **программы обслуживания дисков**, обеспечивающие проверку качества поверхности магнитного диска, контроль сохранности файловой системы на логическом и физической уровнях, сжатие дисков, создание страховых копий дисков, резервирование данных на внешних носителях и др.;

- **программы архивирования данных,** которые обеспечивают процесс сжатия информации в файлах с целью уменьшения объема памяти для ее хранения;
- **программы обслуживания сети.** Эти программы часто называются утилитами. (Заметим, что к антивирусным средствам этот термин обычно не применяется)

- Утилиты - программы, служащие для выполнения вспомогательных операций обработки данных или обслуживания компьютеров (диагностики, тестирования аппаратных и программных средств, оптимизации использования дискового пространства, восстановления разрушенной на магнитном диске информации и т.п.).

- Наибольшее распространение сегодня имеют комплекты утилит: Norton Utilities - фирма Symantec; Checkit PRO Deluxe 2.0 - фирма Touch Stone; PC Tools for Windows 2.0; программа резервного копирования HP Colorado Backup for Windows 95.

Вопрос 2. Прикладное программное обеспечение, его классификация

Прикладное программное обеспечение.

Данный класс программных средств наиболее разнообразен, что обусловлено, прежде всего, широким применением средств компьютерной техники во всех сферах деятельности человека, созданием автоматизированных информационных систем различных предметных областей.

Представим примерную классификацию и типовых представителей прикладного программного обеспечения.

1. Проблемно-ориентированные ППП.

ППП (пакет прикладных программ) - любой комплекс программ, ориентированный на решение некоторого класса задач.

Программные продукты данного класса можно классифицировать по разным признакам:

- типам предметных областей;
- типам информационных систем;
- функциям и комплексам задач, реализуемых программным способом, и др.

- Для некоторых предметных областей возможна типизация функций управления, структуры данных и алгоритмов обработки.
- Это вызвало разработку значительного числа ППП одинакового функционального назначения и, таким образом, создало рынок программных продуктов:
 - **ППП автоматизированного бухгалтерского учета;**
 - **ППП финансовой деятельности;**
 - **ППП управления персоналом (кадровый учет);**
 - **ППП управления материальными запасами;**
 - **ППП управления производством;**
 - **банковские информационные системы и т. п.**

Примеры программных продуктов:
RS-Bank 4.0 - банковская система.
RS- Balance - сетевой программный
комплекс
бухгалтерского учета и др.

2. ППП автоматизированного проектирования

Программы этого класса предназначены для поддержания работы конструкторов и технологов, занимающихся построением чертежей, схем, диаграмм, графическим моделированием и конструированием, созданием библиотеки стандартных элементов чертежей и их многократным использованием, созданием демонстрационных иллюстраций и мультфильмов.

Примеры программных продуктов:

AutoCAD R13.

Auto Vision R1.

Autodesk WorkCentre и др.

3. ППП общего назначения

- Данный класс содержит широкий перечень программных продуктов, поддерживающих преимущественно информационные технологии конечных пользователей.
- Кроме конечных пользователей, этими программными продуктами, благодаря встроенным средствам технологии программирования, могут пользоваться и программисты для создания усложненных программ обработки данных.
- Представители данного класса программных продуктов – системы управления базами данных (СУБД), текстовые процессоры, табличные процессоры, средства презентационной графики, интегрированные пакеты.

Примеры программных продуктов:

- Visual FoxPro Standart 3.0.,
- Access 2.0 for Windows.,
- SQL Server for Windows NT 6.00.,
- dBASE for Windows 5.0.,
- Paradox for Windows 7.0.,
- Oracle 7.0.,
- Informix-Online 7.0.,
- Microsoft SQL Server 6.0.,
- InterBase 4.0 NetWare.,
- SQL Server 6.0 for Windows.

Примеры программных продуктов:

Excel for Windows 5.0/7.0., Quattro Pro 6.0 for Windows.,
Lotus 1-2-3 5.0 for Windows., PowerPoint for Windows 95.,
Screen Cam 1.1 for Windows., Multimedia Viewer. Premier 4.0.,
Freelance Graphics 2.1 for Windows., Autodesk Animator Pro 1.3.,
Autodesk 3D Studio R4, Microsoft Office Standart for Windows95.,
Works for DOS 2.0., Works for Windows 3.0.,
Lotus Notes ViP 1.0., Borland Office., Тройка плюс for Windows и
др.

4. Методо-ориентированные ППП

- Данный класс включает программные продукты, обеспечивающие, независимо от предметной области и функции информационных систем, математические, статические и другие методы решения задач.
- Наиболее распространены методы математического программирования, решения дифференциальных уравнений, имитационного моделирования, исследования операций, методы прогнозирования.

Примеры программных продуктов:

- Storm,
- LP83,
- SAS,
- SPSS,
- SYSTAT,
- StatGraphics и др.

5. **Офисные ППП**

Данный класс программных продуктов включает три подкласса программ, обеспечивающих организационное управление деятельностью офиса:

- **Органайзеры (планировщики)** – программное обеспечение для планирования рабочего времени, составления протоколов встреч, расписаний, ведения записной и телефонной книжки.

- **Программы-переводчики**, средства проверки орфографии и распознавания текста включают:
 - программы-переводчики, предназначенные для создания подстрочника исходного текста на указанном языке;
 - словари орфографии, используемые при проверке текстов;
 - словари синонимов, используемые для стилевой правки текстов;
 - программы для распознавания считанной сканерами информации и преобразования в текстовое представление.

Коммуникационные ППП – предназначены для организации взаимодействия пользователя с удаленными абонентами информационной сети.

- В условиях развития глобальной информационной сети Internet появился новый класс программного обеспечения – браузеры, средства создания WWW-страниц.
- Они различаются возможностями поддержки языка HTML, использованием цвета, текста, графики, звука. Большинство браузеров используют язык Java.

Примеры программных продуктов:

- cc-Mail WorkGroup,
- Norton pcANYWHERE,
- WinFax PRO 4.0,
- NetScape Collabra Share 2.0.,
- Novel GroupWise 4.1.,
- Internet Suit,
- Web Server, Web Author,
- LingvoCorrector,
- Fine Reader 2.0 Standard,
- Business Lingvo и др.

6. Настольные издательские системы

- Данный класс программ включает программы, обеспечивающие информационную технологию компьютерной издательской деятельности.
- Например, ППП Adobe Page Maker 6.0 обеспечивает подготовку многостраничных цветных публикаций, гибкий дизайн страниц, высококачественную печать.

Примеры программных продуктов:

PageMaker 6.0 for Windows.

CorelDraw 6.0.

Corel Ventura 5.0.

Publisher for Windows 95.

Illustrator 4.0 for Windows.

Photoshop 3.0 for Windows и др.

7. Программные средства мультимедиа.

- Этот класс программных продуктов предназначен для создания и использования аудио- и видеоинформации.
- Программные продукты мультимедиа заняли лидирующее положение на рынке в сфере библиотечного информационного обслуживания, процесса обучения, организации досуга.
- Базы данных компьютерных изображений произведений искусства, библиотеки звуковых записей и будут составлять основу для библиотечных каталогов и фондов.

Примеры программных продуктов:

- Sierra Club Collection.
- Outer Space Collection.
- Impressionists Collection.
- Bethoven 9th Symphony.
- Stravinsky.
- Shubert.
- Mozart и др.

8. Системы искусственного интеллекта.

- Данный класс программных продуктов реализует отдельные функции интеллекта человека.
- Основными компонентами систем искусственного интеллекта являются база знаний, интеллектуальный интерфейс с пользователем и программа формирования логических выводов.
- Их разработка идет по следующим трем направлениям:

1. программы-оболочки для создания экспертных систем путем наполнения баз знаний и правил логического вывода:

2. **готовые экспертные системы для принятия решений** в рамках определенных предметных областей;
3. **системы управления базами знаний** для поддержания семантических моделей (процедуральной, семантической сети, фреймовой, продукционной и др.).

Как правило, интеллектуальный интерфейс включает:

- диалоговый процессор на естественном языке;
- планировщик, преобразующий описание задачи в программу решения на основе информации в базе знаний;
- монитор, осуществляющий управление компонентами интерфейса.

Примеры программных продуктов:
Интерэксперт,
Guru и др.

Вопрос 3. Жизненный цикл программного обеспечения

Жизненный цикл программного обеспечения (ПО) — период времени, который начинается с момента принятия решения о необходимости создания программного продукта и заканчивается в момент его полного изъятия из эксплуатации.

Стандарты жизненного цикла ПО

- ГОСТ 34.601-90
- ISO/IEC 12207:1995 (российский аналог — ГОСТ Р ИСО/МЭК 12207-99)

Процессы жизненного цикла ПО

- **Основные:**

- Приобретение (действия и задачи заказчика, приобретающего ПО)
- Поставка (действия и задачи поставщика, который снабжает заказчика программным продуктом или услугой)
- Разработка (действия и задачи, выполняемые разработчиком: создание ПО, оформление проектной и эксплуатационной документации, подготовка тестовых и учебных материалов и т. д.)

- Эксплуатация (действия и задачи оператора — организации, эксплуатирующей систему)
- Сопровождение (действия и задачи, выполняемые сопровождающей организацией, то есть службой сопровождения). Сопровождение — внесений изменений в ПО в целях исправления ошибок, повышения производительности или адаптации к изменившимся условиям работы или требованиям.

- **Вспомогательные:**

- Документирование (формализованное описание информации, созданной в течение ЖЦ ПО)
- Управление конфигурацией (применение административных и технических процедур на всем протяжении ЖЦ ПО для определения состояния компонентов ПО, управления его модификациями).
- Обеспечение качества (обеспечение гарантий того, что ИС и процессы ее ЖЦ соответствуют заданным требованиям и утвержденным планам)

- Верификация (определение того, что программные продукты, являющиеся результатами некоторого действия, полностью удовлетворяют требованиям или условиям, обусловленным предшествующими действиями)
- Аттестация (определение полноты соответствия заданных требований и созданной системы их конкретному функциональному назначению)
- Совместная оценка (оценка состояния работ по проекту: контроль планирования и управления ресурсами, персоналом, аппаратурой, инструментальными средствами)

- Аудит (определение соответствия требованиям, планам и условиям договора)
- Разрешение проблем (анализ и решение проблем, независимо от их происхождения или источника, которые обнаружены в ходе разработки, эксплуатации, сопровождения или других процессов)

- **Организационные :**

- Управление (действия и задачи, которые могут выполняться любой стороной, управляющей своими процессами)
- Создание инфраструктуры (выбор и сопровождение технологии, стандартов и инструментальных средств, выбор и установка аппаратных и программных средств, используемых для разработки, эксплуатации или сопровождения ПО)

- Усовершенствование (оценка, измерение, контроль и усовершенствование процессов ЖЦ)
- Обучение (первоначальное обучение и последующее постоянное повышение квалификации персонала)

Модели жизненного цикла ПО

Каскадная модель

- Каскадная модель жизненного цикла ([«модель водопада»](#), [англ. waterfall model](#)) была предложена в 1970 г. [Уинстоном Ройсом](#).
- Она предусматривает последовательное выполнение всех этапов проекта в строго фиксированном порядке.
- Переход на следующий этап означает полное завершение работ на предыдущем этапе.

- **Этапы проекта в соответствии с каскадной моделью:**

1. Формирование требований;
2. Проектирование;
3. Реализация;
4. Тестирование;
5. Внедрение;
6. Эксплуатация и сопровождение.

Спиральная модель

- Спиральная модель (англ. *spiral model*) была разработана в середине 1980-х годов Барри Боэмом.
- Она основана на классическом цикле Деминга PDCA (plan-do-check-act).
- При использовании этой модели ПО создается в несколько итераций (витков спирали) методом прототипирования.

- **Прототип** — действующий компонент ПО, реализующий отдельные функции и внешние интерфейсы.
- Каждая итерация соответствует созданию фрагмента или версии ПО, на ней уточняются цели и характеристики проекта, оценивается качество полученных результатов и планируются работы следующей итерации.

На каждой итерации спиральной модели оцениваются:

- риск превышения сроков и стоимости проекта;
- необходимость выполнения еще одной итерации;
- степень полноты и точности понимания требований к системе;
- целесообразность прекращения проекта.

Итерационная модель

- Естественное развитие каскадной и спиральной моделей привело к их сближению и появлению современного итерационного подхода, который представляет рациональное сочетание этих моделей.
- Различные варианты итерационного подхода реализованы в большинстве современных технологий и методов ([RUP](#), [MSF](#), [XP](#)).

Вопрос 4. Технологии программирования

- **Программирование** — сравнительно молодая и быстро развивающаяся отрасль науки и техники.
- Опыт ведения реальных разработок и совершенствования имеющихся программных и технических средств постоянно переосмысливается, в результате чего появляются новые методы, методологии и технологии, которые, в свою очередь, служат основой более современных средств разработки программного обеспечения.

Технологией программирования называют совокупность методов и средств, используемых в процессе разработки программного обеспечения.

- Изначально понятия технологии программирования как такового не было — это 60-е годы прошлого столетия — это период "стихийного" программирования.
- В этот период отсутствовало понятие структуры программы, типов данных и т.д.

Вследствие этого код получался запутанным, противоречивым.

Программирование тех лет считалось искусством. Конец 60-х — кризис в программировании.

- Выход из этого кризиса — переход к **структурной парадигме программирования**.
- *Парадигма* - базовый метод или подход, определяющий совокупность методов, используемых на разных этапах разработки, или методологию.

- В основе структурного подхода лежит декомпозиция (разбиение на части) сложных систем с целью последующей реализации в виде отдельных небольших подпрограмм.
- Другим базовым принципом структурного программирования является использование при составлении программ только базовых алгоритмических структур, запрет на использование оператора GOTO.

- Поддержка принципов **структурного программирования** была заложена в основу так называемых процедурных языков программирования.
- Как правило, они включали основные "структурные" операторы передачи управления, поддерживали вложение подпрограмм, локализацию и ограничение области "видимости" данных.
- Среди наиболее известных языков этой группы стоит назвать PL/1, ALGOL-68, Pascal, C.

- Дальнейший рост сложности и размеров разрабатываемого программного обеспечения потребовал развития структурирования данных.
- Как следствие этого в языках появляется возможность определения пользовательских типов данных.
- Одновременно усилилось стремление разграничить доступ к глобальным данным программы.
- В результате появилась и стала развиваться технология **модульного программирования**.

Модульное программирование

предполагает выделение групп подпрограмм, использующих одни и те же глобальные данные, в отдельно компилируемые модули (библиотеки подпрограмм), например, модуль графических ресурсов.

- Связи между модулями при использовании данной технологии осуществляются через специальный интерфейс, в то время как доступ к реализации модуля (телам подпрограмм и некоторым "внутренним" переменным) запрещен.
- Эту технологию поддерживают современные версии языков Pascal и C (C++).

- **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.
- Взаимодействие программных объектов в такой системе осуществляется путем передачи сообщений.

- Объектный подход предлагает новые способы организации программ, основанные на механизмах наследования, полиморфизма, композиции, наполнения. Эти механизмы позволяют конструировать сложные объекты из сравнительно простых.
- Рассмотрим следующие понятия, используемые в ООП:
- **Объект ООП** — это совокупность переменных состояния и связанных с ними методов (операций). Упомянутые методы определяют, как объект взаимодействует с окружающим миром.

- Под **методами объекта** понимают процедуры и функции, объявление которых включено в описание объекта и которые выполняют действия. Возможность управлять состояниями объекта посредством вызова методов в итоге и определяет поведение объекта. Эту совокупность методов часто называют интерфейсом объекта.
- **Инкапсуляция** — это механизм, который объединяет данные и методы, манипулирующие этими данными, и защищает и то и другое от внешнего вмешательства или неправильного использования. Когда методы и данные объединяются таким способом, создается

- **Наследование** — это процесс, посредством которого один объект может наследовать свойства другого объекта и добавлять к ним черты, характерные только для него. В итоге создаётся иерархия объектных типов, где поля данных и методов "предков" автоматически являются и полями данных и методов "потомков".
- Смысл и универсальность наследования заключается в том, что не надо каждый раз заново ("с нуля") описывать новый объект, а можно указать "родителя" (базовый класс) и описать отличительные особенности нового класса.

- **Полиморфизм** — это свойство, которое позволяет одно и то же имя использовать для решения нескольких технически разных задач.
- Полиморфизм подразумевает такое определение методов в иерархии типов, при котором метод с одним именем может применяться к различным родственным объектам.
- В общем смысле концепцией полиморфизма является идея "один интерфейс — множество методов".

Современная технология

программирования — компонентный подход, который 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 собрать в динамически вызываемые библиотеки или исполняемые файлы, распространять в двоичном виде (без исходных текстов) и использовать в любом языке программирования, поддерживающем соответствующую технологию.

- Компонентный подход лежит в основе технологий, разработанных на базе *COM* (*Component Object Model* — компонентная модель объектов), и технологии создания распределенных приложений *CORBA* (*Common Object Request Broker Architecture* — общая архитектура с посредником обработки запросов объектов).
- Эти технологии используют сходные принципы и различаются лишь особенностями их реализации.

Технология СОМ фирмы Microsoft является развитием технологии *OLE (Object Linking and Embedding — связывание и внедрение объектов)*, которая использовалась в ранних версиях Windows для создания составных документов.

- Технология СОМ определяет общую парадигму взаимодействия программ любых типов: библиотек, приложений, операционных систем, независимо от того, функционируют ли эти программы на одном компьютере или на разных компьютерах.
- Модификация СОМ, обеспечивающая передачу вызовов между компьютерами, называется *DCOM (Distributed COM — распределенная СОМ)*.