

Краткий конспект лекций по дисциплине Прикладные компьютерные программы в судостроении

Раздел 1. Понятие пакетов прикладных программ (ППП).

Классификация

программного обеспечения

1. Предмет, содержание и задачи курса

2. Информационная система (ИС). Автоматизированная система (АС).

Многоуровневое представление информационных систем. Аппаратное обеспечение ИС

3. Программное обеспечение (ПО). Программист. Пользователь.

Прикладное

программное обеспечение. Пакет прикладных программ.

Классификация

программного обеспечения

1. Предмет, содержание и задачи курса

Цели и задачи дисциплины.

Изучение основных принципов, используемых в разработке интегрированных программных продуктов.

Изучение структуры, состава и назначения компонентов интегрированного ПО, а также средств организации взаимодействия между компонентами и инструментальных средств расширения функциональности.

Формирование навыков работы со средствами автоматизации решения прикладных задач.

Формирование навыков использования встроенных средств разработки.

Требования к уровню освоения дисциплины.

В результате изучения дисциплины студенты должны:

знать принципы построения прикладных информационных систем
уметь использовать современные программные средства для обработки
разнородной информации;

уметь автоматизировать процесс решения прикладных задач с помощью
встроенных языков программирования;

иметь представление о современном состоянии и тенденциях развития
рынка прикладного ПО.

2. Информационная система (ИС). Автоматизированная система (АС).

Многоуровневое представление информационных систем. Аппаратное обеспечение

ИС

Основные понятия и определения

Информационная система (ИС) – организационно упорядоченная совокупность документов (массивов документов) и информационных технологий, в том числе с использованием средств вычислительной техники и связи, реализующих информационные процессы.

Информационные системы предназначены для хранения, обработки, поиска, распространения, передачи и представления информации.

Автоматизированная (информационная) система (АС) – совокупность программных и аппаратных средств, предназначенных для хранения и/или управления данными и информацией и производства вычислений и управляемая человеком-оператором (в этом главное отличие автоматизированной системы от автоматической).

Многоуровневое представление ИС – модель представления информационной системы в виде совокупности взаимосвязанных уровней, разделенных по функциональному назначению.

Аппаратное обеспечение ИС – комплекс электронных, электрических и механических устройств, входящих в состав информационной системы или сети.

3. Программное обеспечение (ПО). Программист. Пользователь.
Прикладное программное обеспечение. Пакет прикладных программ.
Классификация программного обеспечения

Программное обеспечение (ПО) – совокупность программ и данных, предназначенных для решения определенного круга задач и хранящиеся на машинных носителях.

Программа – последовательность формализованных инструкций, представляющих алгоритм решения некоторой задачи и предназначенная для исполнения устройством

управления вычислительной машины. Инструкции программы записываются при помощи машинного кода или специальных языков программирования. В зависимости от контекста термин «программа» может относиться к исходным текстам, при помощи которых записывается алгоритм, или к исполняемому машинному коду.

Программист – специалист, занимающийся разработкой и проверкой программ.

Различают системных и прикладных программистов.

Пользователь – человек, принимающий участие в управлении объектами и системами некоторой предметной области и являющийся составным элементом автоматизированной системы.

Прикладное программное обеспечение – программное обеспечение, ориентированное на конечного пользователя и предназначенное для решения пользовательских задач. Прикладное ПО состоит из:

отдельных прикладных программ и пакетов прикладных программ, предназначенных для решения различных задач пользователей; автоматизированных систем, созданных на основе этих пакетов.

Пакет прикладных программ – комплект программ, предназначенных для решения задач из определенной проблемной области. Обычно применение пакета прикладных программ предполагает наличие специальной документации: лицензионного свидетельства, паспорта, инструкции

пользователя и т.п. Классификация программного обеспечения Любая классификация подразумевает выбор некоторого группировочного признака (или нескольких), на основании которого и производится отнесение объектов к тому или иному классу. Так, при классификации программного обеспечения по способу распространения можно выделить следующие категории (список не полный):

Commercial Software – коммерческое (с ограниченными лицензией возможностями на использование), разрабатываемое для получения прибыли.

Freeware – свободное ПО, распространяемое без ограничений на использование, модификацию и распространение.

Shareware – условно-бесплатное ПО, с частичными ограничениями при работе в ознакомительном режиме (например, определенное количество запусков программы).

Abandonware – «заброшенное» ПО, поддержка которого непосредственным разработчиком прекращена, но продолжается третьими лицами (например, партнерами или энтузиастами).

Adware – ПО, в код которого включены рекламные материалы. Такое ПО распространяется бесплатно, но для отключения рекламных блоков необходима оплата.

Careware – «благотворительное» ПО, оплату за которое разработчик (или распространитель) просит переводить на благотворительные нужды.

При классификации программного обеспечения по назначению в качестве критерия используют уровень представления ИС, на который ориентирована та или иная программа. Соответственно выделяют следующие классы ПО:

Системное ПО – решает задачи общего управления и поддержания работоспособности системы в целом. К этому классу относят операционные

системы, менеджеры загрузки, драйверы устройств, программные кодеки, утилиты и программные средства защиты информации.

Инструментальное ПО включает средства разработки (трансляторы, отладчики, интегрированные среды, различные SDK и т.п.) и системы управления базами данных (СУБД).

Прикладное ПО – предназначено для решения прикладных задач конечными пользователями.

Прикладное ПО – самый обширный класс программ, в рамках которого возможна дальнейшая классификация, например по предметным областям. В этом случае группировочным признаком является класс задач, решаемых программой. Приведем несколько примеров:

Офисные приложения – предназначены для автоматизации офисной деятельности (текстовые редакторы и процессоры, электронные таблицы, редакторы презентаций и т.п.)

Корпоративные информационные системы – бухгалтерские программы, системы корпоративного управления, системы управления проектами (Project Management), инструменты автоматизации документооборота (EDM-системы) и управления архивами документов (DWM-системы)

Системы проектирования и производства – системы автоматизированного проектирования (САПР, CAD/CAM-системы), системы управления технологическими (SCADA) и производственными (MES) процессами

Научное ПО – системы математического и статистического расчета, анализа и моделирования

Геоинформационные системы (ГИС)

Системы поддержки принятия решений (СППР)

Клиенты доступа к сетевым сервисам (электронная почта, веб-браузеры, передача сообщений, чат-каналы, клиенты файлообменных сетей и т.п.)

Мультимедийное ПО – компьютерные игры, средства просмотра и редактирования аудио- и видеoinформации, графические редакторы и выюеры, анимационные редакторы и т.п.

С точки зрения конечного пользователя такая классификация оправданна и наглядна, для разработчика же более значимым фактором является структура прикладной программы, в общем случае состоящей из нескольких компонентов. Назначение этих компонентов, связи между ними и способность к взаимодействию определяют

интеграцию прикладного ПО. Чем теснее связаны программные компоненты, тем выше степень интеграции.

В зависимости от степени интеграции многочисленные прикладные программные средства можно классифицировать следующим образом:

1. отдельные прикладные программы;
2. библиотеки прикладных программ;
3. пакеты прикладных программ;
4. интегрированные программные системы.

Отдельная прикладная программа пишется, как правило, на некотором высокоуровневом языке программирования (Pascal, Basic и т.п.) и предназначается для решения конкретной прикладной задачи. Такая программа может быть реализована в виде набора модулей, каждый из которых выполняет некоторую самостоятельную функцию (например модуль пользовательского интерфейса, модуль обработки ошибок, модуль печати и т.п.). При этом доступ к функциям модулей из внешних программ невозможен.

Библиотека представляет собой набор отдельных программ, каждая из которых решает некоторую прикладную задачу или выполняет определенные вспомогательные функции (управление памятью, обмен с внешними устройствами и т.п.). Библиотеки программ зарекомендовали себя эффективным средством решения вычислительных задач.

Они интенсивно используются при решении научных и инженерных задач с помощью ЭВМ. Условно их можно разделить на библиотеки общего назначения и специализированные библиотеки.

Пакет прикладных программ (ППП) – это комплекс взаимосвязанных программ, ориентированный на решение определенного класса задач. Формально такое определение не исключает из числа пакетов и библиотеки программ, однако у ППП, как отдельной категории, есть ряд особенностей, среди которых: ориентация на решение классов задач,

унифицированный интерфейс, наличие языковых средств.

Интегрированная программная система – это комплекс программ, элементами которого являются различные пакеты и библиотеки программ. Примером служат системы автоматизированного проектирования, имеющие в своем составе несколько ППП различного назначения. Часто в подобной системе решаются задачи, относящиеся к различным классам или даже к различным предметным областям.

Понятие пакета прикладных программ Итак, пакет прикладных программ (ППП) – это комплекс взаимосвязанных программ для решения определенного класса задач из конкретной предметной области. На текущем этапе развития информационных технологий именно ППП являются наиболее востребованным видом прикладного ПО. Это связано с упомянутыми ранее особенностями ППП. Рассмотрим их подробнее:

Ориентация на решение класса задач. Одной из главных особенностей является ориентация ППП не на отдельную задачу, а на некоторый класс задач, в том числе и специфичных, из определенной предметной области. Так например, офисные пакеты ориентированы на офисную деятельность, одна из задач которой — подготовка документов (в общем случае включающих не только текстовую информацию, но и таблицы, диаграммы, изображения). Следовательно, офисный пакет должен реализовывать функции обработки текста, представлять средства обработки табличной информации, средства

построения диаграмм разного вида и первичные средства редактирования растровой и векторной графики.

Наличие языковых средств. Другой особенностью ППП является наличие в его составе специализированных языковых средств, позволяющих расширить число задач, решаемых пакетом или адаптировать пакет под конкретные нужды. Пакет может представлять поддержку нескольких входных языков, поддерживающих различные парадигмы. Поддерживаемые языки могут быть использованы для формализации исходной задачи, описания алгоритма решения и начальных данных, организации доступа к внешним источникам данных, разработки программных модулей, описания модели предметной области, управления процессом решения в диалоговом режиме и других целей. Примерами входных языков ППП являются VBA в пакете MS Office, AutoLISP/VisualLISP в Autodesk AutoCAD, StarBasic в OpenOffice.org. Единообразие работы с компонентами пакета. Еще одна особенность ППП состоит в наличии специальных системных средств, обеспечивавших унифицированную работу с компонентами. К их числу относятся специализированные банки данных, средства информационного обеспечения, средства взаимодействия пакета с операционной системой, типовой пользовательский интерфейс и т.п.

Раздел 2. Структура и основные компоненты ППП. Эволюция ППП.
Структура и состав MS Office. Основные приложения

1. Структура ППП. Входной язык. Предметное обеспечение. Системное обеспечение 2. Этапы развития ППП. Примеры современных прикладных пакетов. Структура MS Office и назначение компонентов. Документы Microsoft Office. Программная среда. Интерфейс MS Office

1. Структура ППП. Входной язык. Предметное обеспечение. Системное обеспечение

Современные прикладные пакеты – это сложные программные решения, реализующие множество функций. В ходе проектирования и разработки ППП эти функции группируются по назначению и объединяются в структурные компоненты. Можно выделить по меньшей мере три таких компонента ППП: входной язык, предметное обеспечение и системное обеспечение.

Со времени появления первых компьютеров появилось множество прикладных разработок, но, несмотря на разнообразие, их обобщенную внутреннюю структуру можно представить в виде трех взаимосвязанных элементов: 1. входной язык (макроязык, язык управления) – представляет средство общения пользователя с пакетом;

2. предметное обеспечение (функциональное наполнение) – реализует особенности конкретной предметной области; 3. системное обеспечение (системное наполнение) – представляет низкоуровневые средства, например, доступ к функциям операционной системы.

Входной язык – основной инструмент при работе пользователя с пакетом прикладных программ. В качестве входного языка могут использоваться как универсальные (Pascal, Basic и т.п.), так и

специализированные, проблемно-ориентированные языки программирования (Cobol – для бизнес-приложений, Lisp – списочные структуры данных, Fortran и MathLAB – математические задачи и т.п.).

Развитый пакет может обладать несколькими входными языками, предназначенными для выполнения различных функций в рамках решаемого класса задач.

Так, например в пакете OpenOffice.org поддерживаются языки StarBasic, Python, JavaScript и Java. StarBasic является основным входным языком, предназначенным для автоматизации работы с пакетом, для этого языка имеется интегрированная среда разработки и встроенный отладчик. Скрипты на языках Python и JavaScript загружаются и исполняются из внешних файлов. На Java (через SDK и функции API OpenOffice) можно создавать модули расширения и полнофункциональные приложения-компоненты.

Входные языки отражают объем и качество предоставляемых пакетом возможностей, а также удобство их использования. Таким образом, именно входной язык является основным показателем возможностей ППП. Однако стоит отметить, что в современных пакетах обращение пользователя к языковым средствам обычно происходит косвенно, через графический интерфейс.

Предметное обеспечение отражает особенности решаемого класса задач из конкретной предметной области и включает:

- программные модули, реализующие алгоритмы (или их отдельные фрагменты)

- прикладных задач;

- средства сборки программ из отдельных модулей.

Наиболее распространено в настоящее время оформление программных модулей в

виде библиотек, подключаемых статически или динамически. В зависимости от использованного разработчиками подхода к проектированию и реализации ППП такие библиотеки содержат встроенные классы и описания

их интерфейсов (при использовании объектно-ориентированного программирования). При использовании парадигмы структурного программирования в библиотечных модулях содержатся процедуры и функции, предназначенные для решения некоторых самостоятельных задач.

В обоих случаях библиотеки связаны с другими модулями пакета лишь входной и выходной информацией.

Системное обеспечение представляет собой совокупность низкоуровневых средств (программы, файлы, таблицы и т.д.), обеспечивающих определенную дисциплину работы пользователя при решении прикладных задач и формирующих окружение пакета.

К системному обеспечению ППП относятся следующие компоненты:

монитор – программа, управляющая взаимодействием всех компонентов ППП;

транслятор(ы) с входных языков – для ППП характерно использование интерпретируемых языков;

средства доступа к данным – драйверы баз данных и/или компоненты, представляющие

доступ через унифицированные интерфейсы (ODBC, JDBC, ADO, BDE и т.п.);

информационно-справочный модуль – предоставляет функции поддержки, среди

которых информационные сообщения, встроенная справочная системы и т.п.

различные служебные программы, выполняющие низкоуровневые операции

(автосохранение, синхронизация совместно используемых файлов и т.д.)

Приведенная трехкомпонентная логическая структура ППП достаточна условна, она зависит от использованных подходов к проектированию ПО, используемым технологиям программирования, предметной области и других факторов,

вплоть до индивидуальных предпочтений разработчика. Так, в конкретном ППП может отсутствовать четкое разделение программ на предметное и системное обеспечение.

Например, программа планирования вычислений, относящаяся к прикладному обеспечению, может одновременно выполнять и ряд служебных функций (информационное обеспечение, связь с операционной системой и т.п.). С другой стороны, распределенные приложения добавляют свою специфику в структуру ППП.

Кроме того, одни и те же программы в одном пакете могут относиться к предметному обеспечению, а в другом – к системному. Так, программы построения диаграмм в рамках специализированного пакета машинной графики естественно отнести к предметному обеспечению. Однако те же программы следует считать вспомогательными и относящимися к системному обеспечению, например, в пакете решения вычислительных задач.

2. Этапы развития ППП. Примеры современных прикладных пакетов.

Структура MS Office и назначение компонентов. Документы Microsoft Office.

Программная среда. Интерфейс MS Office

Первые пакеты прикладных программ представляли собой простые тематические подборки программ для решения отдельных задач в той или иной предметной области, обращение к ним выполнялось с помощью средств оболочки ОС или из других программ.

Современный пакет является сложной программной системой, включающей специализированные системные и языковые средства. В относительно короткой истории развития вычислительных ППП можно выделить 4 основных поколения(класса) пакетов.

Каждый из этих классов характеризуется определенными особенностями входящих состав ППП компонентов – входных языков, предметного и системного обеспечения.

Первое поколение

В качестве входных языков ППП первого поколения использовались универсальные языки программирования (Фортран, Алгол-60 и т.п.) или языки управления заданиями соответствующих операционных систем. Проблемная ориентация входных языков достигалась за счет соответствующей мнемоники в идентификаторах.

Составление заданий на таком языке практически не отличалось от написания программ на алгоритмическом языке. Предметное обеспечение первых ППП, как правило, было организовано в форме библиотек программ, т.е. в виде наборов (пакетов) независимых программ на некотором базовом языке программирования (отсюда впервые возник и сам термин «пакет»). Такие ППП иногда называют пакетами библиотечного типа, или пакетами простой структуры.

В качестве системного обеспечения пакетов первого поколения обычно использовались штатные компоненты программного обеспечения ЭВМ: компиляторы с алгоритмических языков, редакторы текстов, средства организации библиотек программ, архивные системы и т.д. Эти пакеты не требовали сколь-нибудь развитой системной

поддержки, и для их функционирования вполне хватало указанных системных средств общего назначения. В большинстве случаев разработчиками таких пакетов были прикладные программисты, которые пытались приспособить универсальные языки программирования к своим нуждам.

Второе поколение

Разработка ППП второго поколения осуществлялась уже с участием системных программистов. Это привело к появлению специализированных входных языков на базе универсальных языков программирования. Проблемная ориентация таких языков достигалась не только за счет использования определенной мнемоники, но также применением соответствующих языковых конструкций, которые упрощали формулировку задачи и делали ее более наглядной. Транслятор с такого языка представлял собой препроцессор (чаще всего макропроцессор) к транслятору соответствующего алгоритмического языка. В качестве модулей в пакетах этого класса стали использоваться не только программные единицы (т.е. законченные программы на том или ином языке программирования), но и такие объекты, как последовательность операторов языка программирования, совокупность данных, схема счета и др.

Существенные изменения претерпели также принципы организации системного обеспечения ППП. В достаточно развитых пакетах второго поколения уже можно выделить элементы системного обеспечения, характерные для современных пакетов:

монитор, трансляторы с входных языков, специализированные банки данных, средства описания модели предметной области и планирования вычислений и др.

Третье поколение

Третий этап развития ППП характеризуется появлением самостоятельных входных языков, ориентированных на пользователей-непрограммистов. Особое внимание в таких ППП уделяется системным компонентам обеспечивающим простоту и удобство. Это достигается главным образом за счет специализации входных языков и включения в состав пакета средств автоматизированного планирования вычислений.

Четвертое поколение

Четвертый этап характеризуется созданием ППП, эксплуатируемых в интерактивном режиме работы. Основным преимуществом диалогового взаимодействия с ЭВМ является возможность активной обратной связи с пользователем в процессе постановки задачи, ее решения и анализа полученных результатов. Появление и интенсивное развитие различных форм диалогового общения обусловлено прежде всего прогрессом в области технических средств (графическая подсистема ЭВМ и средства мультимедиа, сетевые средства).

Развитие аппаратного обеспечения повлекло за собой создание разнообразных программных средств поддержки диалогового режима работы (диалоговые операционные системы, диалоговые пакеты программ различного назначения и т. д.). Прикладная система состоит из диалогового монитора – набора универсальных программ, обеспечивающих ведение диалога и обмен данными, и базы знаний о предметной области.

Информация о структуре, целях и который монитор управляет ходом диалога. Носителями процедурных знаний о предметной области являются прикладные модули, реализующие функции собственно системы.

Таким образом, создание прикладной системы сводится к настройке диалогового монитора на конкретный диалог, путем заполнения базы знаний. При этом программировать в традиционном смысле этого слова приходится лишь прикладные модули, знания о диалоге вводятся в систему с помощью набора соответствующих средств – редактора сценариев. Логично требовать, чтобы редактор сценариев также представлял собой диалоговую программу, отвечавшую рассмотренным выше требованиям. Благодаря готовому универсальному монитору программист может сосредоточиться на решении чисто прикладных задач, выделение же знаний о диалоге в сценарий обеспечивает в значительной степени необходимая гибкость программного продукта.

Большое внимание в настоящее время уделяется проблеме создания «интеллектуальных ППП». Такой пакет позволяет конечному пользователю лишь сформулировать свою задачу в содержательных терминах, не указывая алгоритма ее решения. Синтез решения и сборка целевой программы производятся автоматически. При этом детали вычислений скрыты от пользователя, и компьютер становится интеллектуальным партнером человека, способным понимать его задачи. Предметное обеспечение подобного ППП представляет собой некоторую базу знаний, содержащую как процедурные, так и описательные знания. Такой способ решения иногда называют концептуальным программированием, характерными особенностями которого является программирование в терминах предметной области использования ЭВМ уже на этапе постановки задач, автоматический синтез программ решения задачи, накопление знаний о решаемых задачах в базе знаний.

Структура MS Office и назначение компонентов

ППП Microsoft Office – это совокупность программных средств автоматизации офисной деятельности. В состав пакета входит множество приложений, каждое из которых предназначено для выполнения определенных функций и может быть использовано автономно и независимо от остальных. Весь набор офисных приложений можно разделить на основные и дополнительные.

Набор приложений, входящих в состав MS Office может отличаться в зависимости от версии. Это позволяет подобрать наиболее соответствующий текущим задачам дистрибутив этого пакета.

Основные компоненты Microsoft Office

Список и назначение основных компонентов, входящих в состав Microsoft Office, приведен ниже.

Основные компоненты Microsoft Office

Название приложения	Функциональное назначение приложения
---------------------	--------------------------------------

Microsoft Word	Текстовый процессор
----------------	---------------------

Microsoft Excel	Табличный процессор
-----------------	---------------------

Microsoft PowerPoint	Система подготовки презентаций
----------------------	--------------------------------

Outlook	Система управления персональной информацией
---------	---

Microsoft Access	Система управления базами данных
------------------	----------------------------------

Microsoft Binder	Система управления подшивками
------------------	-------------------------------

Microsoft FrontPage	Система управления Web-узлами
---------------------	-------------------------------

Microsoft PhotoDraw	Графический редактор
---------------------	----------------------

Microsoft Publisher	Настольная издательская система
---------------------	---------------------------------

Microsoft Project	Система управления проектами
-------------------	------------------------------

Microsoft Team Manager	Система управления персоналом
------------------------	-------------------------------

Дополнительные компоненты MS Office

Кроме основных компонентов, в семейство Microsoft Office входит большое количество вспомогательных приложений, которые устанавливаются (или не устанавливаются) вместе с основными. Ими можно воспользоваться из основных приложений или вызвать независимо. В таблице 2 перечислены некоторые из вспомогательных приложений.

Кроме основных и вспомогательных приложений, могут быть установлены и использованы различные расширения (надстройки). Их можно условно разделить на три группы:

1. Самостоятельные приложения, разработанные фирмой Microsoft, которые являются

компонентами семейства Microsoft Office, но формально не входят в состав пакета.

Примерами являются приложения Microsoft Project и Microsoft Team Manager.

2. Надстройки над компонентами Microsoft Office, разработанные фирмой Microsoft и представляющие собой дополнительные функции. Как правило, надстройки оформляются не в виде готовых к выполнению программ, а в виде документов специального типа:

шаблонов, рабочих книг, библиотек динамической компоновки (DLL) и т.п.

3. Приложения третьих фирм, разработанные для пользователей Microsoft Office. В этот класс попадают как продукты сторонних фирм, так и собственные разработки пользователей. Сюда можно отнести средства распознавания текстов (OCR), автоматического перевода текста, средства управления большими массивами документов (перечисленные задачи не реализованы или слабо развиты в самом пакете MS Office).