

**Программное обеспечение. Прикладное программное обеспечение (библиотеки программ, пакеты прикладных программ, интегрированные системы).**

**Программное обеспечение (ПО)** – комплекс программ, обеспечивающих обработку или передачу данных и предназначенных для многократного использования и применения разными пользователями (часто считают, что в состав ПО следует включать программную документацию). По видам выполняемых функций программное обеспечение подразделяется на: системное, прикладное и инструментальное.

**Системное ПО** – нужно для обеспечения работы компьютера и вычислительных сетей (создание операционной среды функционирования других программ; обеспечение надежной и эффективной работы самого компьютера и вычислительной сети; диагностика и профилактика аппаратуры компьютера и вычислительных сетей; выполнение вспомогательных технологических процессов: копирование, архивация, восстановление файлов и др.).

**Инструментальное ПО** – используется в ходе разработки, корректировки или развития других программ: редакторы, компиляторы, отладчики, вспомогательные системные программы, графические пакеты и др.

**Прикладное ПО** – программное обеспечение, состоящее из отдельных прикладных программ, библиотек программ, пакетов прикладных программ, интегрированных систем, предназначенное для решения различных прикладных задач пользователей.

**Отдельная программа (может быть модульной)** - пишется на некотором языке программирования, обращение к ней осуществляется с помощью штатных средств системы программирования, настройка на конкретную задачу осуществляется с помощью параметров.

**Библиотека программ** – совокупность программ для решения типовых задач некоторой прикладной области. Свойства программ библиотеки: общее программное обеспечение, единообразие способов внутренней организации и использования.

**Пакет прикладных программ** – комплекс взаимосвязанных прикладных программ (библиотека) и системных средств для решения некоторого класса характерных задач прикладной области.

**Интегрированная система** – комплекс программ (элементами которого могут быть отдельные программы, библиотеки, пакеты прикладных программ, системные средства) для решения комплексных (сложных, составных, разнородных) задач.

**Пять поколений языков общения с машиной. Языки 4GL и 5GL.**

**1GL** – машинные языки

**2GL** – автокоды (языки ассемблера)

**3GL** – языки программирования высокого уровня

**4GL** – простейшие языки запросов информационных систем, языки запросов, ЯОД, ЯМД (баз данных), языки генераторов отчетов, языки поддержки принятия решений (статистический / прогнозный анализ), языки генераторов приложений (RPG – Report Program Generator).

**5GL** – должны обеспечивать автоматическое построение прикладных программ на основании данных, которые вводятся в компьютер в виде максимально удобном и привычном для пользователя, не знакомого с программированием.

**Качественное отличие языков 4GL и 5GL** от языков предыдущих поколений – заключается в том, что эти языки служат не для описания алгоритмов, а для спецификации решаемых пользователем задач (описывается, что нужно получить, но не описывается, как это получить).

**Функциональные требования к 4GL:** интегрированное множество функций, язык очень высокого уровня, совместимость с языками 3GL, дружественный интерфейс, средства анализа и моделирования данных, высокая эффективность при сильной нагрузке, поддержка стандартных СУБД.

## Интеллектуализация ЭВМ.

**Интеллектуализация ЭВМ** - внедрение программного обеспечения, позволяющего конечному пользователю решать на ЭВМ любые задачи, возникающие в сфере его профессиональной деятельности без каких бы то ни было посредников (в том числе, программистов).

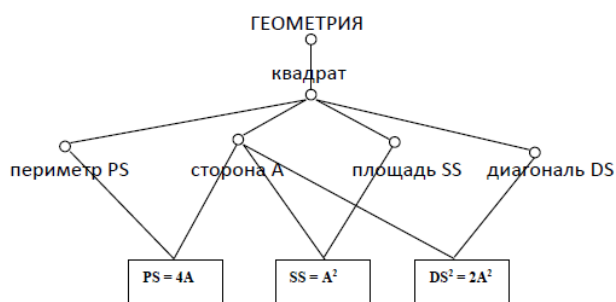
**Автоматический синтез программ** – автоматическое/автоматизированное построение программы по описанию ее назначения или действия, называемому обычно спецификацией программы.

**Решение задач на вычислительных моделях (работы Энна Тыгу); задача на вычислительной модели, отношения вычислимости. Примеры.**

**Идея подхода.** Синтез программ – составление алгоритма решения задачи на вычислительной модели предметной области (ПО). Такая модель строится заранее. Она состоит из совокупности переменных, соответствующих объектам ПО, и отношений вычислимости. Каждое отношение связывает значения нескольких переменных и интерпретируется так: по известным значениям некоторых переменных (входных переменных) отношения можно вычислить значения других (выходных переменных).

**Задача на вычислительной модели** – тройка  $(V, U, M)$ , где  $V$  – множество входных переменных,  $U$  – множество выходных переменных,  $M$  – вычислительная модель.

**Схема отношения вычислимости**



ПРОГРАММА' ПРИМЕР1;

ПО' ГЕОМ;

ПУСТЬ' K1: КВАДРАТ ДИАГОНАЛЬ = 5;

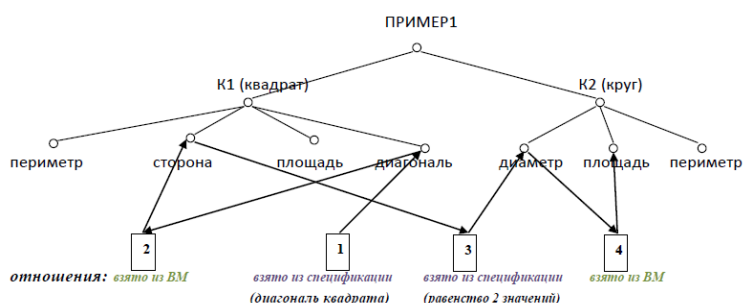
K2: КРУГ ДИАМЕТР = K1.СТОРОНА;

ДЕЙСТВИЯ'

НА ПРИМЕР1' ВЫЧИСЛИТЬ' K2.ПЛОЩАДЬ;

КОНЕЦ' +++

**Схема вычислительной модели**



Формулы этой логической теории строятся из логических связок и **отношений**

**вычислимости**, имеющих вид:  $u \rightarrow v$ .

Интерпретация такого отношения:  $v$  числимо по  $u$  применением функции  $f$ .

Теорема существования решения –

формула  $\exists f (u \rightarrow f \rightarrow v)$ . Она выводится из аксиом, образующих вычислительную модель задачи.

Пример: 
$$\frac{x \rightarrow f_1 Y, Y \rightarrow f_2 Z}{x \rightarrow f_1, f_2, Z}$$

### **Основные особенности экспериментальной диалоговой системы синтеза программ ЛИСС.**

**Система ЛИСС** предназначена для синтеза типичных лисп-программ (функций) обработки списков, используемых как вспомогательные при построении более сложных реальных программ.

#### **Основные особенности подхода:**

способность в случаях, когда человек не был исчерпывающе точен на этапе постановки задачи, – начинать работу с неполной формулировкой задачи и уточнять ее автоматически или с помощью пользователя

вмешательство человека в процесс синтеза для внесения дополнений, поправок и т.п.

средства проверки соответствия спецификации, а при необходимости и самой программы замыслу пользователя

средства исправления найденных ошибок

средства должны обеспечивать возможность изменения исходной спецификации и повторения процесса синтеза.

Используемый в системе ЛИСС лингвистический процессор обеспечивает устойчивое функционирование системы даже в случаях появления в тексте незнакомых ей слов, оборотов, форм с ошибками, характерными при работе человека за терминалом (нажатие соседней клавиши, пропуск буквы и т.п.). Пользователь может ввести удобные для него условные обозначения. Если словесная спецификация оказалась неполной, ЛИСС начинает ее уточнять. В идеале необходимо разрешение всех обнаруженных неясностей и противоречий в диалоге с пользователем. Система ЛИСС реализует компромисс между требованиями надежности результата и краткости диалога. На основе заложенных в нее знаний она автоматически пополняет исходную спецификацию и выбирает наиболее вероятную интерпретацию неоднозначностей, обращаясь к помощи пользователя только в наиболее сложных случаях. Пользователь может проверить текущее состояние процесса синтеза, исправить спецификацию, отменить сделанные системой предположения. При успешном окончании синтеза полученная программа выдается на экран терминала и запоминается. Если пользователь не удовлетворен результатами тестирования, он продолжает отладочные действия, возобновляя работу системы по синтезу нужной ему программы.

### **Эргономические аспекты человеко-машинного интерфейса. Микроэргономика. Эргономика программного обеспечения. Мидиэргономика. Эргономические характеристики интерфейса.**

**Эргономика** – научно-прикладная дисциплина, занимающаяся изучением и созданием эффективных систем, управляемых человеком. Эргономика изучает движение человека в процессе производственной деятельности, затраты его энергии, производительность и интенсивность при конкретных видах работ.

**Микроэргономика** – исследование и проектирование систем «человек-машина».

**Эргономика программного обеспечения** – подраздел микроэргономики, ориентированный на системы «человек-компьютер», «человек-компьютер-человек», «человек-компьютер-процесс», «человек-программа» и т.п.

**Мидиэргономика** – исследование и проектирование систем «человек-коллектив», «коллектив-машина», «человек-сеть», «коллектив-организация». Исследует взаимодействия на уровне рабочих мест и производственных задач. В сферу интересов мидиэргономики входят: проектирование организаций, планирование работ, обитаемость рабочих помещений, гигиена труда, проектирование интерфейсов сетевых программных продуктов и др.

**Макроэргономика** – исследование и проектирование систем «человек-общество», «организация-система организаций».

**Эргономические характеристики:** конструктивные особенности оборудования, доступность и надежность системы, качество разработки диалога.

### **Интерфейс. Пользовательский интерфейс.**

**Интерфейс** – полная совокупность сведений о входных и выходных сигналах, которыми могут обмениваться процессоры данных (устройство – устройство, программа – датчики состояния среды, человек – компьютерная система).

ИЛИ

**Интерфейс** – совокупность программных и аппаратных средств, поддерживающих подобные обмены данными.

**Пользовательский интерфейс** – интерфейс между конечным пользователем и компьютерной системой.

### **Основные требования к интерфейсу.**

**Основные требования к интерфейсу:** точность, удобство (комфорт для пользователя): социальные факторы (возраст, образование), физический комфорт (не может прочитать сообщение из-за неудачного шрифта, фона), психологический комфорт (может прочитать сообщение, но не понимает его).

### **Дружественный интерфейс.**

#### **Дружественный интерфейс:**

естественные языковые конструкции и структуры меню (не требуется знание синтаксиса формальных языков общения с компьютером);

«интуитивный» уровень взаимодействия с компьютером, не требующий длительного обучения (для профессионала – несколько часов);

разнообразные средства общения, пригодные для пользователей различного уровня подготовки (командный язык, меню, пиктограммы, диаграммы и др.);

для каждого уровня пользователя адекватные возможности в: меню, запросах, подсистеме помощи; работа в реальном времени (необходимая скорость в диалоге);

использование манипулятора типа «мышь», «горячих клавиш», сенсорного экрана и др.;

минимальное использование клавиатуры;

«интеллектуальные» средства (устойчивость к ошибкам, широкое использование принципа «по умолчанию», опережающий ввод).

## **Интеллектуальный интерфейс**

**Интеллектуальный интерфейс** – совокупность программных и аппаратных средств, позволяющая конечному пользователю решать на компьютере характерные для его повседневной деятельности задачи без помощи посредников-программистов. Расширение взаимодействия между человеком и компьютером с помощью: увеличения диапазона способов ввода и вывода, обогащения грамматики ввода и вывода, попытки кооперации с пользователем в достижении целей.

## **Основные аспекты интерфейса (процессы ввода-вывода и диалог).**

**Процессы ввода-вывода** служат для того, чтобы принять от пользователя данные и передать ему данные через различные физические устройства.

**Диалог** связывает фоновые процессы в одно целое.

## **Диалог "человек-компьютер". Задачи диалога.**

**Диалог «человек-компьютер»** – обмен информацией между пользователем и вычислительной системой (с помощью терминала и по определенным правилам).

**Задачи диалога:** определение задания, передача/прием входных данных и размещение их в переменных соответствующего процесса в нужной форме, вызов процесса, вывод/получение результатов обработки.

## **Виды сообщений в диалоге [умение указать вид конкретного сообщения в примере диалога].**

### **Пользователь**

**команда** (входные управляющие сообщения), **данные** (входные данные), возможны и сложные сообщения, объединяющие команду и данные.

### **Система**

**подсказка** (побуждение ввести команду или данные в некотором формате), **данные** (результаты, т.е. данные, которые возвращает процесс по окончании обработки), **сообщение о состоянии** (что произошло/происходит в системе, например сообщение о получении и начале обработки введенных по запросу системы данных или о том, что система продолжает работу – при большой задержке), **сообщение об ошибке** (сигнал диалогового процесса о том, что дальнейшее выполнение работы невозможно, т.к. невозможно, например, обработать очередное сообщение пользователя), **справка** (что можно/нужно делать и почему в ситуации, когда пользователь этого не знает).

**Диалог, управляемый системой; диалог, управляемый пользователем.**

**Диалог, управляемый системой**, – система жестко задает, какое задание можно выбрать и какие данные вводить.

Пример:

Продавец: Что желаете?

Покупатель: Яблоки, пожалуйста?

Продавец: Антоновку, коричные, мельбу?

Покупатель: Антоновку.

Продавец: Сколько?

**Диалог, управляемый пользователем**, – пользователь подает команду на выполнение нужного на данном этапе задания.

Пример:

Покупатель: 2 кило антоновки, пожалуйста.

**Проверка входных данных. Подсказки.**

**Проверка входных данных:** сравнение с элементами набора, сравнение с шаблоном (дд/мм/гг), проверка грамматической правильности.

**Подсказки:** меню – если диапазон правильных входных данных не слишком велик, указание на требуемый формат (Счет от (дд/мм/гг)), форма для заполнения, запрос на ввод команды (сообщение о том, что система просто ожидает входные данные, не указывая, какими они могут быть (A >)).

**Типы (структуры) диалога: диалог типа ВОПРОС-ОТВЕТ, диалог на основе КОМАНДНОГО ЯЗЫКА, диалог типа МЕНЮ, диалог на основе ЭКРАННЫХ ФОРМ.**

**Диалог типа ВОПРОС-ОТВЕТ**

Построен по аналогии с обычным интервью. Система берет на себя роль интервьюера и получает информацию от пользователя в форме ответов на ее вопросы. Описание структуры диалога типа ВОПРОС-ОТВЕТ с помощью сетей переходов: Вершина = состояние (выводится сообщение системы, ожидается ответ П), Дуга = переход, Метка (на дуге) = условие перехода.

**Диалог на основе КОМАНДНОГО ЯЗЫКА.**

Система обычно ничего не выводит, кроме подсказки (имя устройства, символ "приглашения": A>, который означает готовность системы к работе/вводу) и, разумеется, результатов выполнения команды пользователя.

**Диалог типа МЕНЮ**

Система предлагает набор возможных вариантов ответа (можно вводить и команды, и данные).

Основные форматы (структуры) МЕНЮ: БЛОК ДАННЫХ, СТРОКА, НАБОР ПИКТОГРАММ.

Выбор из МЕНЮ: просмотр списка объектов, указание на объект или ввод идентификатора

Иерархические МЕНЮ, Всплывающие/спадающие МЕНЮ (Pop-up), Скрытые МЕНЮ.

### **Диалог на основе ЭКРАННЫХ ФОРМ.**

Задается сразу несколько вопросов (есть установки по умолчанию); ввод после ответа на несколько нужных вопросов.

**Их достоинства и недостатки, сравнительный анализ, рекомендации по использованию.**

#### **Диалог типа ВОПРОС-ОТВЕТ**

##### **Достоинства:**

возможна подсказка со стороны системы (например, формат данных: YY/MM/DD);

возможно (в любой момент) обращение к подсистеме ПОМОЩИ (HELP);

гибкость, удобство при большой ветвистости дерева диалога.

**Недостаток:** много нажатий клавиш

Диалог типа **ВОПРОС-ОТВЕТ** пригоден и для ввода данных с клавиатуры и для выбора из списка. На диапазон значений входных данных не накладывается никаких ограничений. Однако эта структура неудобна для использования способа прямого указания.

#### **Диалог на основе КОМАНДНОГО ЯЗЫКА.**

##### **Недостатки:**

Пользователь должен знать синтаксис командного языка.

Часто: плохая диагностика (минимальный контроль со стороны системы).

Диалог на основе **КОМАНДНОГО ЯЗЫКА** ориентирован прежде всего на подготовленного пользователя и требует в ответ на стандартный запрос ввода ответов, описывающих (в соответствии с жесткими синтаксическими ограничениями) задачу и ассоциированные с ней данные.

#### **Диалог типа МЕНЮ**

##### **Достоинства:**

возможные варианты ответов пользователя предлагает сама система (подсказка), пользователю не нужно вводить тексты, тем самым уменьшается вероятность ошибки ввода.

##### **удобно в ситуациях**

диапазон ответов мал и все они могут быть явно отражены на экране, пользователю необходимо видеть возможные варианты ответов.

Диалог типа **МЕНЮ** позволяет предъявить пользователю точный список вариантов ввода (возможно, с использованием иерархии, скрытого меню и др.) и дает возможность выбора одного из них: вводом идентификатора с клавиатуры, просмотром списка на экране и прямым указанием на объект и т.д.).

#### **Диалог на основе ЭКРАННЫХ ФОРМ.**

##### **Достоинства:**

привычно для человека (заполнение таблиц/бланков/анкет),

быстрее чем работа по схеме ВОПРОС-ОТВЕТ или по схеме МЕНЮ,

удобно, если можно заранее определить стандартную последовательность вводимых данных.

**Недостаток:** схема плоха в ситуации, когда процесс выбора ответа имеет структуру дерева.

Диалог **на основе ЭКРАННЫХ ФОРМ** требует от пользователя ответов на группу вопросов. Обычно пользователь может отвечать на вопросы в любом порядке, может не отвечать на вопросы с предустановленным значением по умолчанию, может редактировать свои ответы. Обеспечиваются и гибкость, и удобство для пользователя.

### **Многооконные WIMP-интерфейсы. Метафоры, используемые в соответствующей технологии.**

**Аббревиатура WIMP** трактуется так: W - информация представлена на экране в виде окон (Windows). I - объекты представлены в виде пиктограмм (Icons). M - выборка производится с помощью манипулятора типа «мышь» (Mouse). P - используются меню: «всплывающие» (Pop-up), «вытягиваемые» (Pull-down).

Для их реализации необходимы достаточные ресурсы: память прямого доступа (RAM) для представления изображений в несколько раз превышающих объем физического экрана; экран с поточечной адресацией для представления графических изображений (в том числе, пиктограмм); адекватные вычислительные мощности для быстрой перерисовки экрана и работы в мультипрограммном режиме.

#### **Три метафоры:**

**Конкретный объект** (папка, файл).

Не нужно изучать, запоминать и вводить директивы типа сору MyFile/My Directory. Файл может быть скопирован "буксировкой" его с помощью манипулятора "мышь" по нужную иконку каталога (прямое манипулирование).

**«Что вижу, то и получу»** (фактический эффект любого действия немедленно отображается на экране). Более точно было бы – "видишь то, что получил".

#### **Рабочий стол и Буфер вырезок.**

Метафора Рабочий стол предполагает, что интерфейс поддерживает доступ к множеству различных информационных источников и форматов, позволяет пользователю легко переключаться с одного источника на другой. Столь же легко можно менять тип задания. При этом доступны вспомогательные средства: часы, калькулятор, записная книжка и др.

Как и в прежних интерфейсах есть возможность временно хранить в специальном буфере изъятые части документа. В WIMP-интерфейсах используется "магический" **Буфер вырезок**, который дает возможность вставки столько копий вырезанного фрагмента, сколько требуется пользователю.

### **Окно как элемент интерфейса. Основные операции с окнами. Пиктограммы, ярлыки, баннеры, диалоговые окна.**

**Окно** – специальная область (обычно прямоугольная) физического окна (экрана), с помощью которой пользователь обозревает отдельные аспекты своего взаимодействия с компьютерной системой/задачей. В случае многооконного интерфейса отображение на физическом экране тех частей буфера, которые соответствуют окнам реализует процесс управления отображением. Он реализует основные Операции с окнами: открытие, закрытие, свертка, выбор активного окна, перемещение окна (относительно его буфера), перемещение окна (относительно экрана), изменение атрибутов окна.



**Пиктограмма (Icon)** – небольшое окно с изображением, отражающим содержание/функции некоторого буфера/файла/программы.

**Ярлык (ShortCart)** – пиктограмма + ссылка на соответствующий буфер/файл/программу/адрес в Интернете.

**Баннер** (в Интернете) – окно (графическое изображение или текстовый блок) рекламного характера, являющийся гиперссылкой на веб-страницу с расширенным описанием продукта или услуги.

### **Прямое манипулирование. Указание и выбор. Некоторые дополнительные возможности интерфейсов.**

**Прямое манипулирование** – непосредственный результат метафоры конкретного объекта. Если окна, буферы и другие элементы данных являются "физическими сущностями" то для работы с ними "нужны физические действия", а не команды. Прямое манипулирование: выборка, открытие, буксировка, резиновая нить (изменение размеров окна), экранные кнопки.

Основное достоинство – нет необходимости изучать лексику, синтаксис и семантику языка взаимодействия с системой. Интерфейсы прямого манипулирования позволяют достичь большого разнообразия операций с помощью небольшого набора механизмов (что удобно при разработке крупных проектов). Хотя работа может осуществляться операциями указания и выборки, многим опытным пользователем более привычно и удобно использование традиционной структуры команд.

#### **Указание и выбор**

Пользователь может определить любое окно как активное подведением курсора и нажатием клавиши; пиктограмма может быть раскрыта до нужного окна. Для файла можно открыть буфер. Если файл текстовый, может быть открыт буфер редактирования; если он исполняемый (программа), он может быть активизирован. Указание также используется для позиционирования курсора.

### **Некоторые дополнительные возможности интерфейсов**

#### **Динамические визуальные сигналы**

Изменение изображения на экране с целью дать пользователю дополнительную информацию. Так, при выполнении программой длительных действий курсор мыши приобретает форму песочных часов (или же такие часы появляются рядом с курсором). Второй пример – изменение изображения кнопки при нажатии на нее. Это – сигнал о том, что система «считает», что пользователь взаимодействует именно с этой кнопкой.

#### **Анимационный интерфейс**

Основан на том очевидном факте, что движущийся объект легче привлекает внимание. Во-первых, анимация облегчает восприятие некоторых процессов и структур в интерфейсе. Во-вторых, анимация привлекает внимание. В-третьих, анимация может показывать соответствие между действиями пользователя и событиями в интерфейсе.

#### **Звуковые эффекты**

#### **Мультимедиа**

Компьютерные технологии, обеспечивающие одновременное использование нескольких информационных сред: текст, графика, фото, видео, анимация, звуковые эффекты.