

Создание пользовательского интерфейса.

Пользовательский интерфейс - совокупность программных и аппаратных средств, обеспечивающих взаимодействие пользователя с компьютером. Основу такого взаимодействия составляют диалоги.

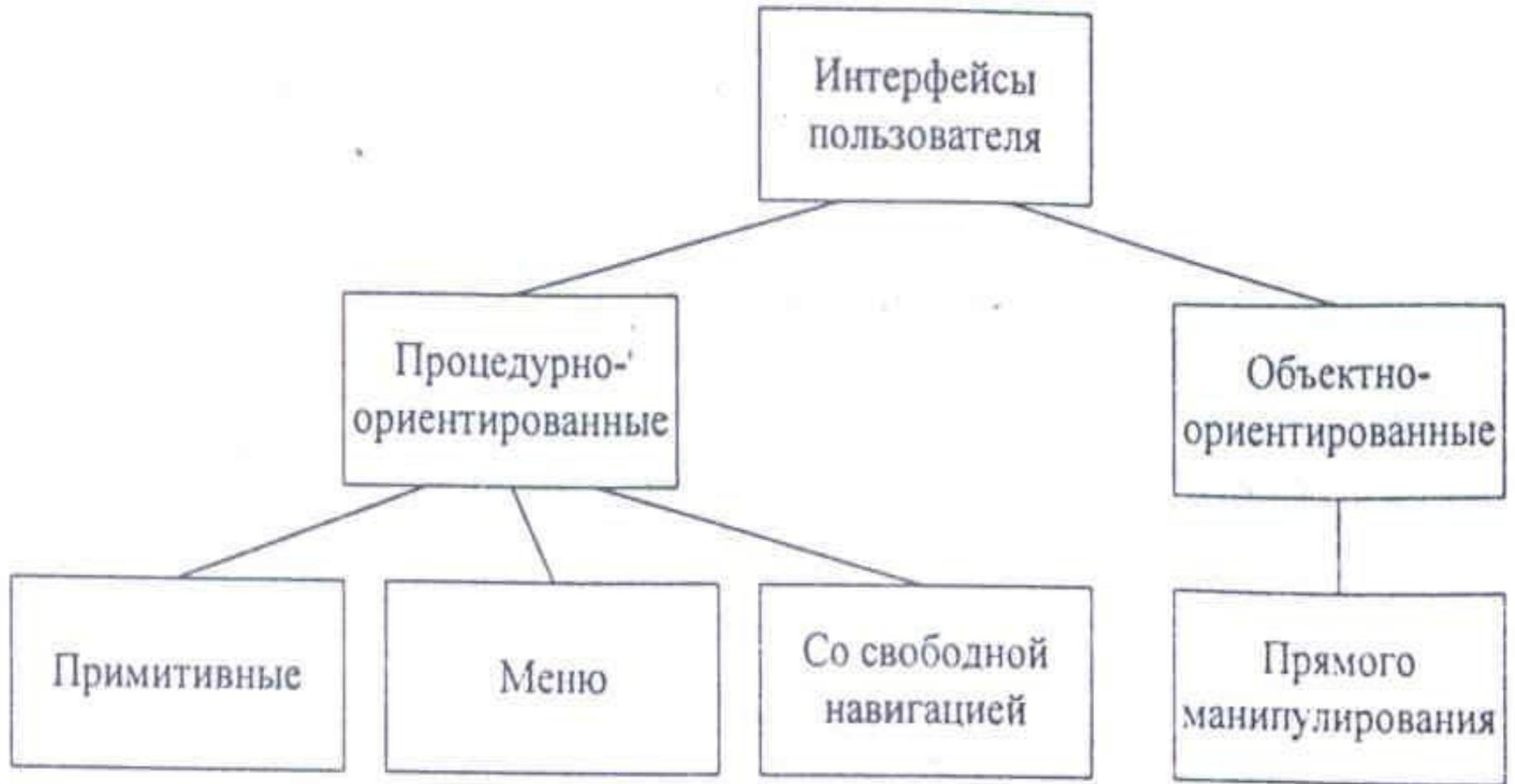
Диалог - регламентированный обмен информацией между человеком и компьютером, осуществляемый в реальном масштабе времени и направленный на совместное решение конкретной задачи: обмен информацией и координация действий. Обмен информацией осуществляется передачей сообщений и управляющих сигналов. *Сообщение* - порция информации, участвующая в диалоговом обмене. *Различают:*

- **входные сообщения**, которые генерируются человеком с помощью средств ввода:

клавиатуры, манипуляторов, например мыши и т. п.;

- **выходные сообщения**, которые генерируются компьютером в виде текстов, звуковых сигналов и/или изображений и выводятся пользователю на экран монитора или другие устройства вывода информации

Типы интерфейсов.



2. Типы интерфейсов

Процедурно-ориентированные интерфейсы используют традиционную модель взаимодействия с пользователем, основанную на понятиях «процедура» и «операция». В рамках этой модели программное обеспечение предоставляет пользователю возможность выполнения некоторых *действий*, для которых пользователь определяет соответствующие данные и следствием выполнения которых является получение желаемых результатов.

Объектно-ориентированные интерфейсы используют модель взаимодействия с пользователем, ориентированную на манипулирование *объектами* предметной области. В рамках этой модели пользователю предоставляется возможность напрямую взаимодействовать с каждым объектом и инициировать выполнение операций, в процессе которых взаимодействуют несколько объектов.

Основные отличия пользовательских моделей интерфейсов процедурного и объектно-ориентированного типов.

Процедурно-ориентированные пользовательские интерфейсы	Объектно-ориентированные пользовательские интерфейсы
Обеспечивают пользователей функциями, необходимыми для выполнения задач	Обеспечивают пользователям возможность взаимодействия с объектами
Акцент делается на задачи	Акцент делается на входные данные и результаты
Пиктограммы представляют приложения, окна или операции	Пиктограммы представляют объекты
Содержание папок и справочников отображается с помощью таблиц и списков	Папки и справочники являются визуальными контейнерами объектов

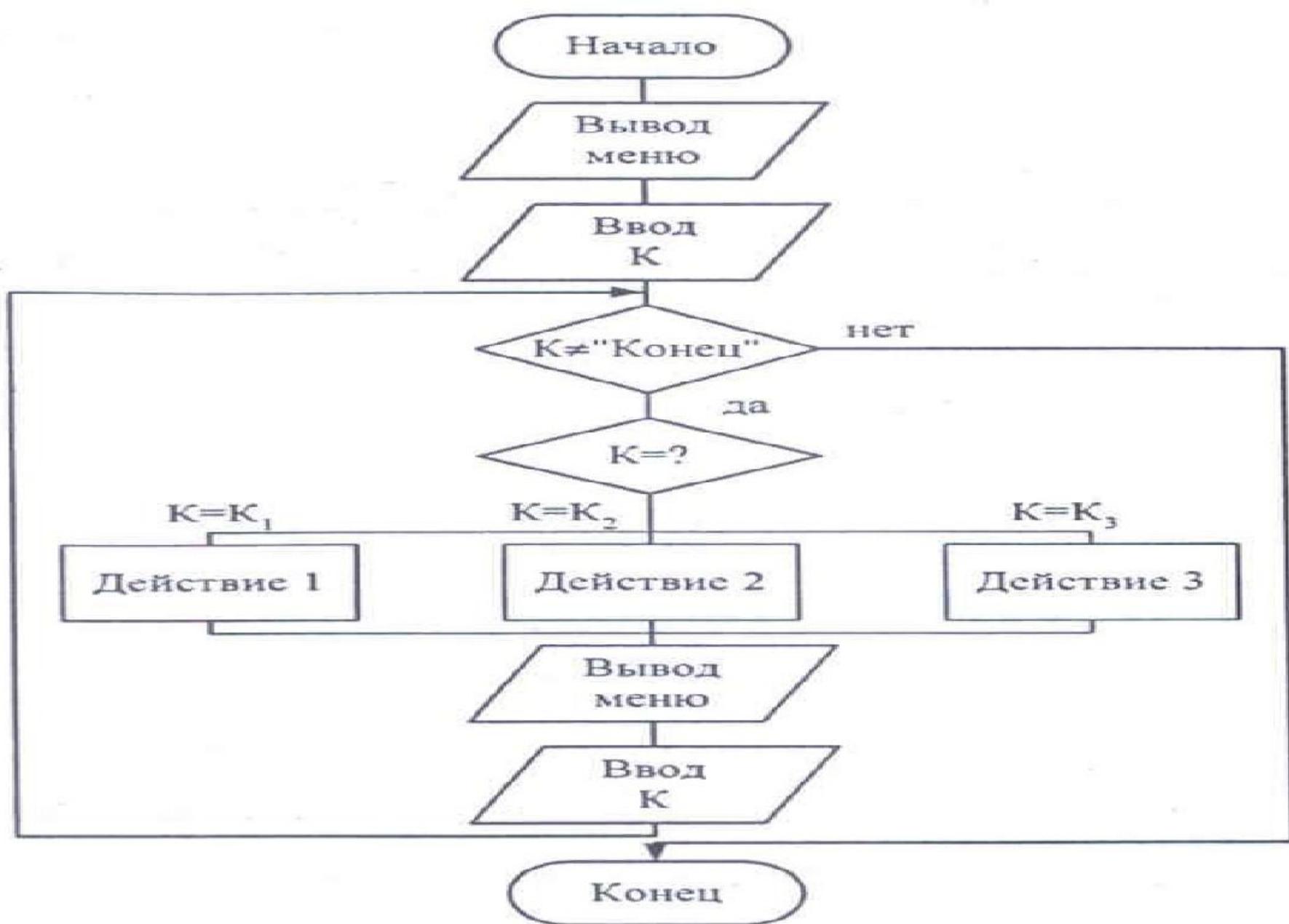
Различают процедурно–ориентированные интерфейсы трех типов:

«примитивные», меню и со свободной навигацией.

Примитивным называют интерфейс, который организует взаимодействие с пользователем в консольном режиме.

Интерфейс-меню в отличие от примитивного интерфейса позволяет пользователю выбирать необходимые операции из специального списка, выводимого ему программой. Эти интерфейсы предполагают реализацию множества сценариев работы, последовательность действий в которых определяется пользователем.

Различают **одноуровневые** и **иерархические меню**. Первые используют для сравнительно простого управления вычислительным процессом, когда вариантов немного (не более 5-7), и они включают операции одного типа, например, Создать, Открыть, Закреть и т. п. Вторые – при большом количестве вариантов или их очевидных различиях. например.



Типичная структура алгоритма программы с одноуровневым меню

Алгоритм программы с многоуровневым меню обычно строится по уровням, причем выбор команды на каждом уровне осуществляется так же, как для одноуровневого меню.

Интерфейс-меню предполагает, что программа находится либо в состоянии Уровень меню, либо в состоянии Выполнение операции.

Древовидная организация меню предполагает строго ограниченную навигацию: либо переходы «вверх» к корню дерева, либо - «вниз» по выбранной ветви. Каждому уровню иерархического меню соответствует свое определенное окно, содержащее пункты данного уровня

Интерфейсы со свободной навигацией также называют *графическими пользовательскими интерфейсами* (GUI - Graphic User Interface) или интерфейсами WYSIWYG (What You See Is What You Get - что видишь, то и получишь, т. е., что пользователь видит на экране, то он и получит при печати)..

Графические интерфейсы поддерживают концепцию интерактивного взаимодействия с программным обеспечением, осуществляя визуальную обратную связь с пользователем и возможность прямого манипулирования объектами и информацией на экране. Кроме того, интерфейсы данного типа поддерживают концепцию совместимости программ, позволяя перемещать между ними информацию (технология OLE)

В отличие от интерфейса-меню интерфейс со свободной навигацией обеспечивает возможность осуществления любых допустимых в конкретном состоянии операций, доступ к которым возможен через различные интерфейсные компоненты. Например, окна программ, реализующих интерфейс Windows, обычно содержат:

- меню различных типов: ниспадающее, кнопочное, контекстное;
- разного рода компоненты ввода данных.

Выбор следующей операции в меню осуществляется как мышью, так и с помощью клавиатуры.

Существенной особенностью интерфейсов данного типа является способность изменяться в процессе взаимодействия с пользователем, предлагая выбор только тех операций, которые имеют смысл в конкретной ситуации. Реализуют интерфейсы со свободной навигацией, используя событийное программирование и объектно-ориентированные библиотеки, что предполагает

Объектно-ориентированные интерфейсы пока представлены только *интерфейсом прямого манипулирования*. Этот тип интерфейса предполагает, что взаимодействие пользователя с программным обеспечением осуществляется посредством выбора и перемещения *пиктограмм*, соответствующих объектам предметной области. Для реализации таких интерфейсов также используют событийное программирование и объектно-ориентированные библиотеки.

Пример.1. Разработать пользовательский интерфейс программы построения графиков или вывода таблицы функций.

Можно предложить четыре варианта интерфейса, соответствующие рассмотренным выше типам.

Вариант 1. Использование примитивного интерфейса предполагает, что пользователь сразу определяет все параметры, необходимые программе для построения графика или вывода таблицы, вводя их в ответ на соответствующие запросы программы, после чего программа выполняет необходимые вычисления и выводит результат.



б

Типичная структура алгоритма программ с примитивным интерфейсом:

а – последовательный; *б* – с возможностью повторения

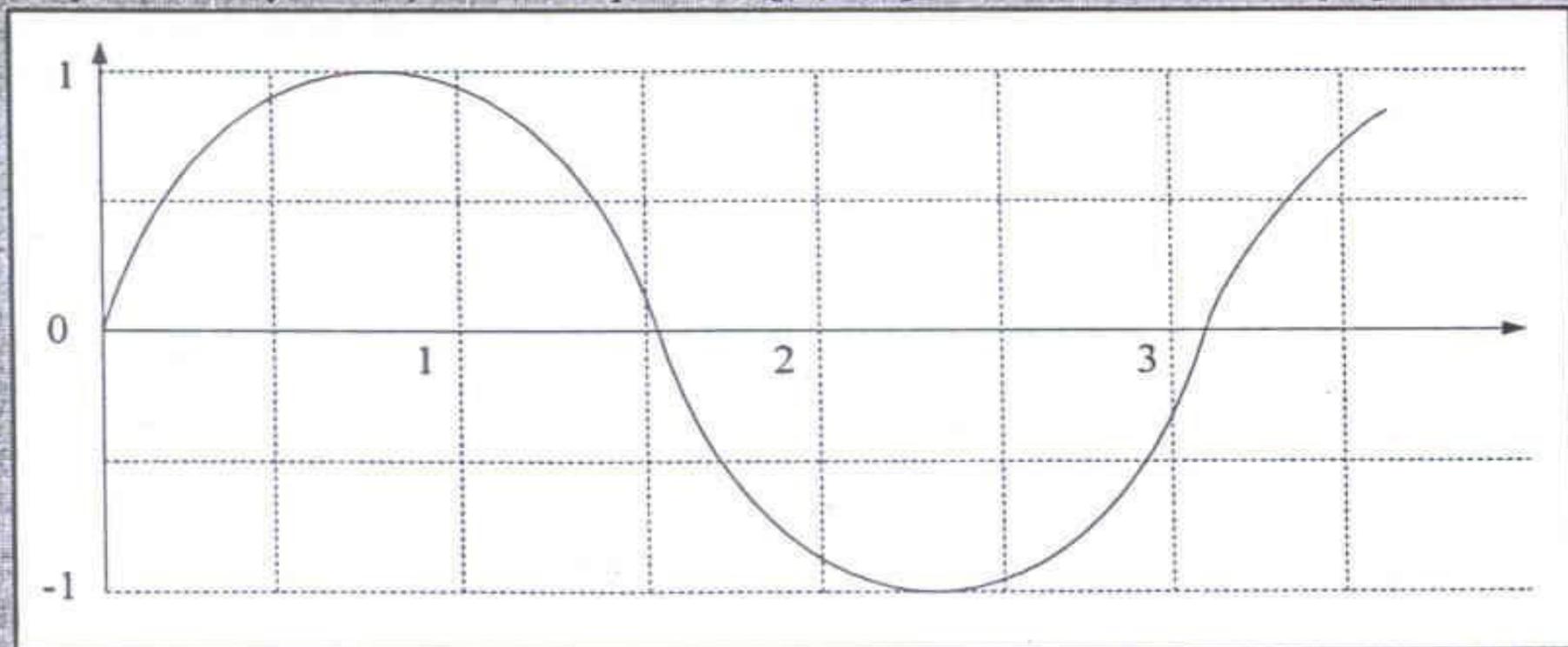
Вариант 2. Можно использовать одноуровневое меню, которое будет включать команды: Функция, Отрезок, Шаг, Тип результата, Выполнить и Выход

Программа построения графиков/таблиц функций на заданном интервале

Функция | Отрезок | Шаг | Вид результата | Выполнить | Выход

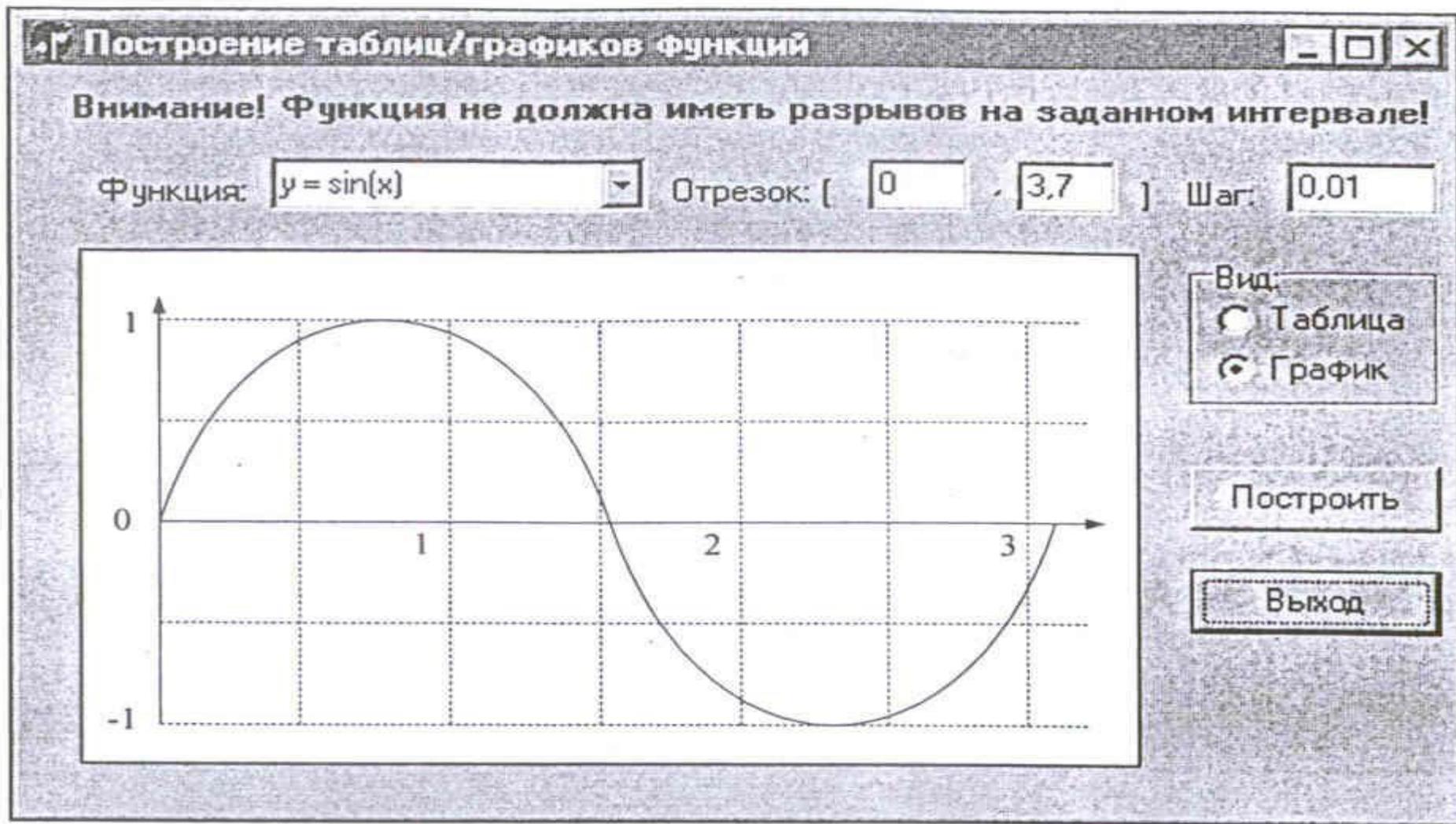
Внимание! Функция не должна иметь разрывов на заданном интервале

Функция $y=\sin(x)$ | Отрезок: $[0, 3,7]$ | Шаг: 0,01 | Вид: график



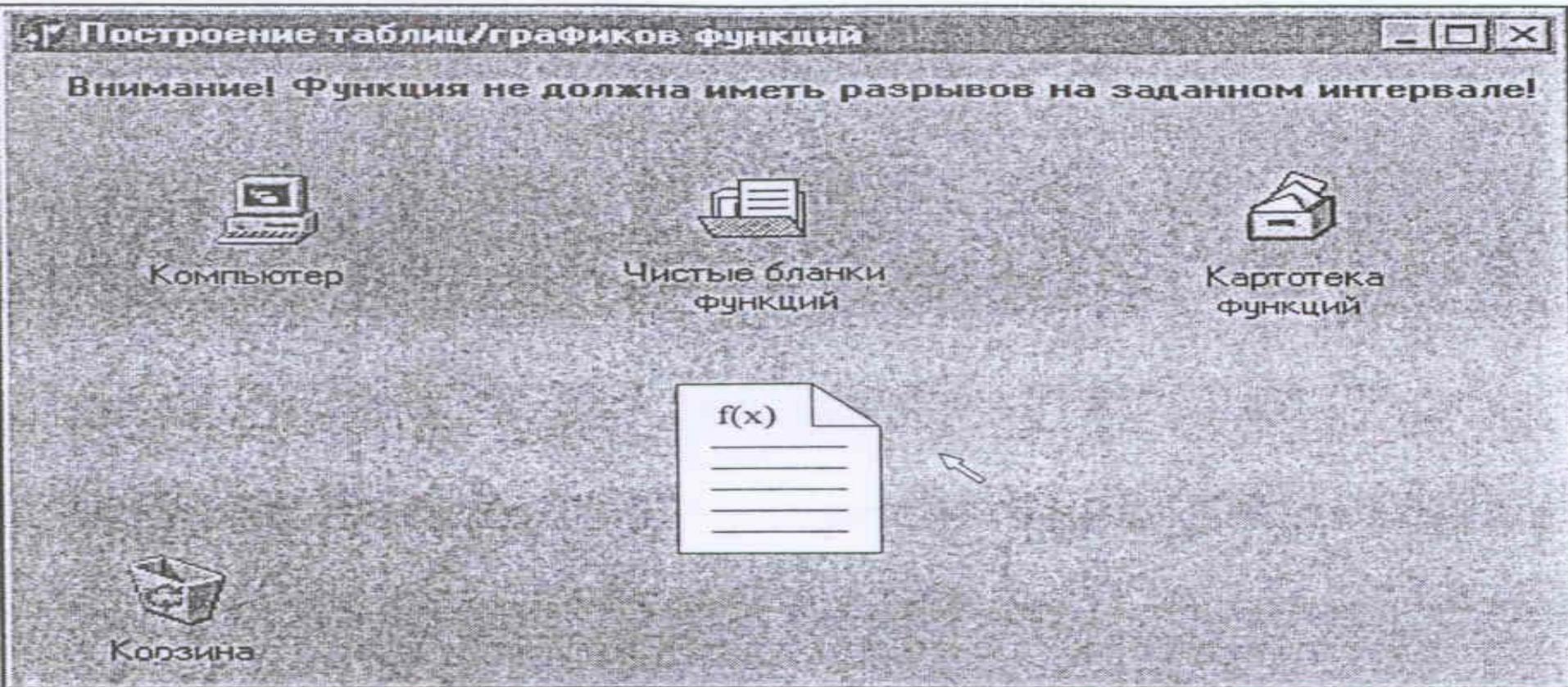
Внешний вид экрана программы построения графиков/таблиц функций (интерфейс – меню)

Вариант 3. Интерфейс со свободной навигацией . Менять данные можно в любой момент и в любом порядке, используя соответствующие компоненты ввода-вывода.



Внешний вид окна программы построения графиков/таблиц функций (интерфейс со свободной навигацией)

Вариант 4. Интерфейс прямого манипулирования. Для того чтобы ввести новую формулу, необходимо взять чистый бланк из папки. Бланк раскрывается двойным щелчком мыши, после чего его необходимо заполнить. Затем его можно «обсчитать», перенеся на пиктограмму компьютера. Заполненные бланки, которые могут еще понадобиться, «кладутся» в папку Функции, остальные - в «корзину». Менять данные и тип результатов можно в любой момент и в любом порядке, «раскрыв» бланк.



Внешний вид окна программы построения таблиц/графиков функций (интерфейс прямого манипулирования)

Различают также однодокументные (SDI -Single Document Interface) и многодокументные (MDI - Multiple Document Interface) интерфейсы.

Однодокументные или «однооконные» интерфейсы организуют работу, как следует из названия, только с одним документом, например, текстом или рисунком. Чтобы посмотреть другой текст, необходимо запустить еще одну копию приложения, что допустимо только в мультипрограммной операционной системе. Поэтому такие интерфейсы используют, если одновременная работа с несколькими документами маловероятна.

Многодокументные или «многооконные» интерфейсы соответственно организуют в тех случаях, когда велика вероятность, что пользователю понадобится одновременно

Этапы разработки пользовательского интерфейса. Разработка пользовательского интерфейса включает те же основные этапы, что и разработка программного обеспечения:

- постановка задачи - определение типа интерфейса и общих требований к нему;
- анализ требований и определение спецификаций - определение сценариев использования и пользовательской модели интерфейса;
- проектирование — проектирование диалогов и их реализация в виде процессов ввода-вывода;
- реализация - программирование и тестирование интерфейсных процессов.

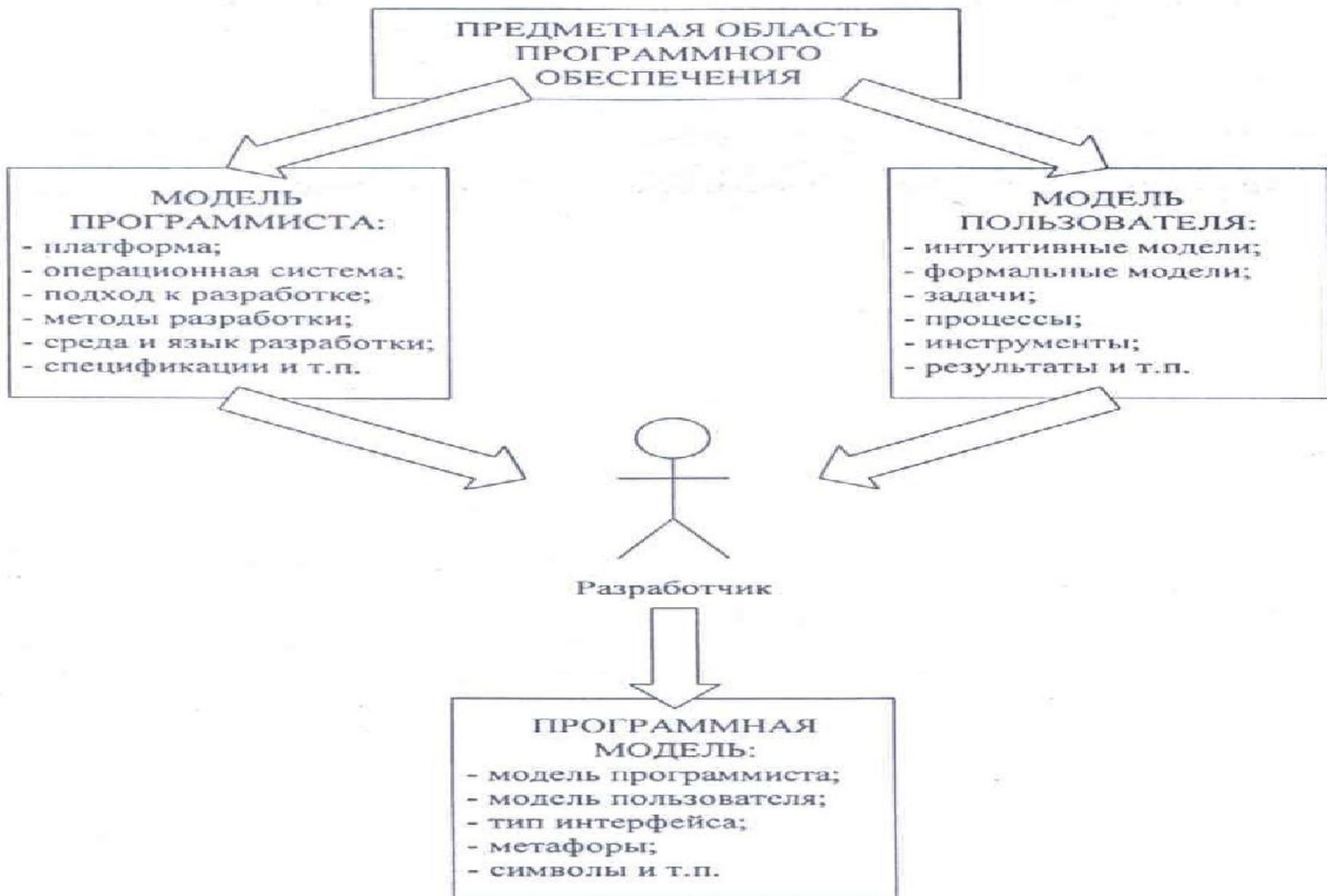
Пользовательская и программная модели интерфейса

Существуют три совершенно различные модели пользовательского интерфейса: **модель программиста, модель пользователя и программная модель.** Программист, разрабатывая пользовательский интерфейс, исходит из того, управление какими операциями ему необходимо реализовать в пользовательском интерфейсе, и как это осуществить, не затрачивая ни существенных ресурсов компьютера, ни своих сил и времени. Его интересуют функциональность, эффективность, технологичность, внутренняя стройность и другие не связанные с удобством пользователя характеристики программного обеспечения.

Пользовательская модель интерфейса - это совокупность обобщенных представлений конкретного пользователя или некоторой группы пользователей о процессах, происходящих во время работы программы или программной системы. Эта модель базируется на *особенностях* опыта конкретных пользователей, который характеризуется:

- уровнем подготовки в предметной области разрабатываемого программного обеспечения;
- интуитивными моделями выполнения операций в этой предметной области;
- уровнем подготовки в области владения компьютером;
- устоявшимися стереотипами работы с компьютером.

Приведение в соответствие моделей пользователя и программиста, а также построение на их базе программной модели интерфейса задача не тривиальная. Чем сложнее автоматизируемая предметная область, тем сложнее оказывается построить программную модель интерфейса, учитывающую особенности пользовательской модели и не требующую слишком больших затрат как в процессе разработки, так и во время работы. С этой точки зрения объектные интерфейсы кажутся наиболее перспективными, так как в их основе лежит именно отображение объектов предметной области, которыми оперируют пользователи. Хотя на настоящий момент времени их реализация достаточно трудоемка.



Процесс проектирования пользовательского интерфейса

Критерии оценки интерфейса пользователем.

Многочисленные опросы и обследования, проводимые ведущими фирмами по разработке программного обеспечения, показали, что основными критериями оценки интерфейсов пользователем являются:

- простота освоения и запоминания операций системы - конкретно оценивают время освоения и продолжительность сохранения информации в памяти;
- скорость достижения результатов при использовании системы - определяется количеством вводимых или выбираемых мышью команд и настроек;
- субъективная удовлетворенность при эксплуатации системы (удобство работы, утомляемость и т. д.).

Классификации диалогов и общие принципы их разработки

Различают тип диалога и его форму.

Типы диалога. Тип диалога определяет, кто из «собеседников» управляет процессом обмена информацией. Соответственно различают два типа диалога: **управляемые программой и управляемые пользователем.**

Диалог, управляемый программой, предусматривает наличие жесткого, линейного или древовидного, т. е. включающего возможные альтернативные варианты, сценария диалога, заложенного в программное обеспечение. Такой диалог обычно сопровождают большим количеством подсказок, которые уточняют, какую информацию необходимо вводить на каждом шаге.

Диалог, управляемый пользователем, подразумевает, что сценарий диалога зависит от пользователя, который применяет систему для выполнения необходимых ему операций. При этом система

Формы диалога. Никакой диалог невозможен, если не существует языка, понятного «собеседникам». Описание языка, на котором ведется диалог, включает определение его **синтаксиса** - правил, определяющих допустимые конструкции (слова, предложения) языка или его форму, и **семантики** - правил, определяющих смысл синтаксически корректных конструкций языка или его содержание. В зависимости от вида используемых в конкретном случае синтаксиса и семантики различают три формы диалога:

- фразовую,
- директивную,
- табличную.

Фразовая форма предполагает «общение» с пользователем на естественном языке или его подмножестве. Содержание диалога в данной форме составляют повелительные, повествовательные и вопросительные предложения и ответы на вопросы. Общение может осуществляться в свободном формате, но возможна и фиксация отдельных фраз.

Организация диалога на естественном языке на современном уровне - задача нерешенная, так как естественный язык крайне сложен и пока не удается в достаточной степени формализовать его синтаксис и семантику.

Основными недостатками фразовой формы при использовании подмножества естественного языка являются:

- большие затраты ресурсов;
- отсутствие гарантии однозначной интерпретации формулировок;
- необходимость ввода длинных грамматически правильных фраз.

Основное достоинство фразовой формы состоит в относительно свободном общении с системой.

Директивная форма предполагает использование команд (директив) специально разработанного формального языка. Командой в этом случае называют предложение этого языка, описывающее комбинированные данные, которые включают идентификатор иницируемого процесса и, при необходимости, данные для него.

Команду можно вводить:

- в виде строки текста, специально разработанного формата, например, команды MS DOS, которые вводятся в командной строке;
- нажатием некоторой комбинации клавиш клавиатуры, например, комбинации «быстрого доступа» современных Windows-приложений;

Основными достоинствами директивной формы являются:

- сравнительно небольшой объем вводимой информации;
- гибкость - возможности выбора операции в данном случае ограничены только набором допустимых команд;
- ориентация на диалог, управляемый пользователем;
- использование минимальной области экрана или неиспользование ее вообще;
- возможность совмещения с другими формами.

Недостатки директивной формы:

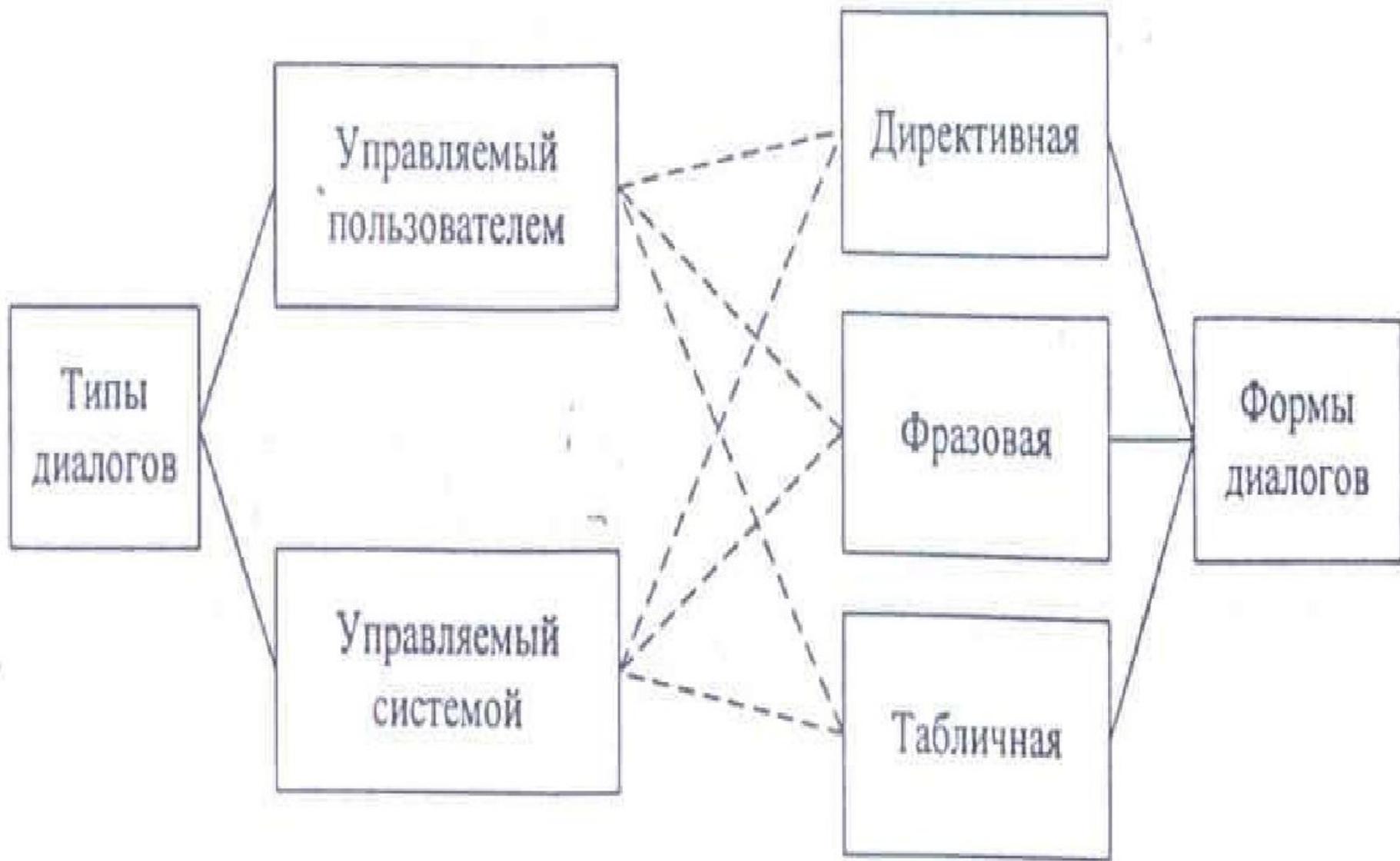
- практическое отсутствие подсказок на экране, что требует запоминания вводимых команд и их синтаксиса;
- почти полное отсутствие обратной связи о состоянии инициированных процессов;

Табличная форма предполагает, что пользователь выбирает ответ из предложенных программой. Язык диалога для табличной формы имеет простейший синтаксис и однозначную семантику, что достаточно легко реализовать. Достоинствами табличной формы являются:

- наличие подсказки, что уменьшает нагрузку на память пользователя, так как данная форма ориентирована не на запоминание, а на узнавание;
- сокращение количества ошибок ввода: пользователь не вводит информацию, а указывает на нее;
- сокращение времени обучения пользователя;
- возможность совмещения с другими формами;
- в некоторых случаях возможность настройки пользователем.

К недостаткам данной формы относят.

- необходимость наличия навыков навигации по экрану;
- использование сравнительно большой площади экрана для изображения визуальных компонентов;
- интенсивное использование ресурсов компьютера, связанное с необходимостью постоянного обновления информации на экране.



Соответствие типов диалогов и его форм

Разработка диалогов. Процесс проектирования и реализации диалогов можно разделить на следующие стадии:

- определение множества необходимых диалогов, их основных сообщений и возможных сценариев — проектирование абстрактных диалогов;
- определение типа и формы каждого диалога, а также синтаксиса и семантики используемых языков - проектирование конкретных диалогов;
- выбор основных и дополнительных устройств и проектирование процессов ввода-вывода для каждого диалога, а также уточнение передаваемых сообщений - проектирование технических диалогов.

В основу абстрактных диалогов должна закладываться идеология технологического процесса, для автоматизации которого предназначается программный продукт. Именно анализируя составляющие автоматизируемого технологического процесса, разработчик определяет сценарии диалогов, которые должны быть предусмотрены в программном обеспечении.

Кроме сценариев, при проектировании абстрактных диалогов используют диаграммы состояний интерфейса или графы диалога.

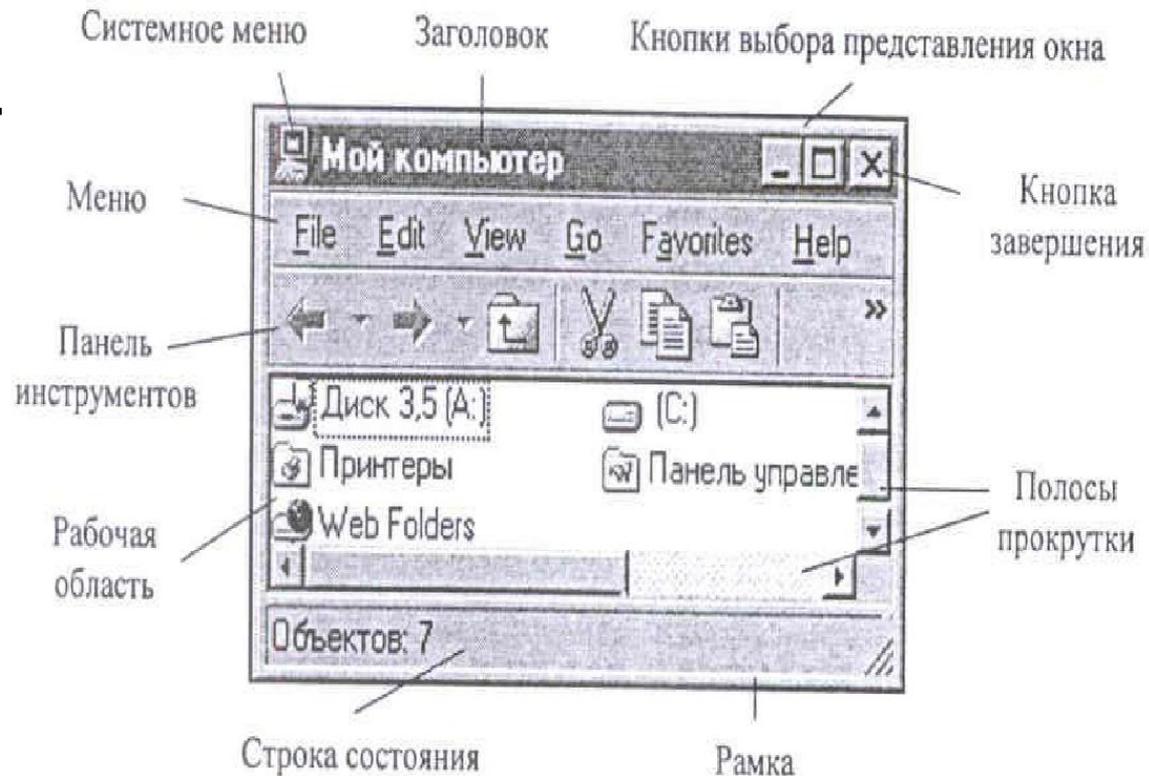
Основные компоненты графических пользовательских интерфейсов

Графические пользовательские интерфейсы поддерживаются операционными системами Windows, Apple Macintosh, OS/2 и т. д. В рамках указанных операционных систем для таких интерфейсов разработаны наборы стандартных компонентов взаимодействия с пользователем. Эти наборы не идентичны, как и основные приемы работы с интерфейсами различных операционных систем.

Пользовательские интерфейсы большинства современных программ строятся по технологии WIMP: W - Windows (окна), I - Icons (пиктограммы), M - Mouse (мышь), P - Pop-up (всплывающие или выпадающие меню). Основными элементами графических интерфейсов, таким образом, являются: окна, пиктограммы, компоненты ввода-вывода и мышь, которую используют в качестве указующего устройства и устройства прямого манипулирования объектами на экране

Окна. Окно - обычно прямоугольная, ограниченная рамкой область физического экрана. Окно может менять размеры и местоположение в пределах экрана. Все окна можно разделить на 5 категорий:

- **основные окна (окна приложений);**
- **дочерние или подчиненные окна;**
- **окна диалога;**
- **информационные окна;**
- **окна меню.**



1. Окно приложения и его элементы

Дочернее окно Windows используют в многодокументных программных интерфейсах (MDI). В отличие от окна приложения дочернее окно не содержит меню. В строке заголовка - специальное имя, идентифицирующее связанный с ним документ или файл. Пиктограммы всех дочерних окон одинаковы.

Диалоговое окно Windows используют для просмотра и задания различных режимов работы, необходимых параметров или другой информации. Оно может содержать:

- строку заголовка с кнопкой системного меню;
- компоненты, обеспечивающие пользователю возможность ввода или выбора ответа;
- вспомогательные компоненты, обеспечивающую подсказку, например, поле предварительного просмотра или кнопку вызова справки.

Как правило, размер диалогового окна неизменяем, но

Информационные окна бывают двух типов: окна сообщений и окна помощи. Окна сообщений, кроме заголовка с кнопкой системного меню, обычно содержат текст сообщения и одну или несколько кнопок реакции пользователя, например, кнопки Yes и No или кнопки Yes, No и Cancel.

Окно помощи имеет более сложную структуру: оно может содержать меню, полосы прокрутки и информационная область, т. е. по структуре оно аналогично окну приложения, но отличается от него тем, что имеет узко специальное назначение, обеспечивая навигацию по справочной информации.

Окна меню Windows можно использовать как открывающиеся панели иерархического меню или как отдельные контекстные меню. Каждой строке окна меню может соответствовать:

- команда;
- меню следующего уровня, что обозначается стрелкой;
- окно диалога, что обозначается тремя точками.

Кроме того, в некоторых строках добавляется указание клавиш быстрого вызова.

Пиктограммы. Пиктограмма представляет собой небольшое окно с графическим изображением, отражающим содержимое буфера, с которым она связана. Различают:

- программные пиктограммы;
- пиктограммы дочерних окон;
- пиктограммы панели инструментов;
- пиктограммы объектов.

Программными пиктограммами, которые связаны с соответствующей программой, управляет операционная система. Например, можно «свернуть» окно приложения в пиктограмму на панели задач Windows или «развернуть» его обратно «на рабочий стол». Многодокументная программная система управляет пиктограммами дочерних окон, обеспечивая доступ к различным документам, одновременно обрабатываемым программной системой.

Пиктограммы панели инструментов обычно дублируют доступ к соответствующим функциям через меню, обеспечивая их быстрый вызов.

Пиктограммы объектов используют для прямого манипулирования этими объектами. Все пиктограммы можно перемещать мышью. Для облегчения работы с пиктограммами обычно используют «всплывающие» подсказки

Прямое манипулирование изображением.

Прямое манипулирование изображением – это возможность замены команды воздействия на некоторый объект физическим действием в интерфейсе, осуществляемым с помощью мыши. При этом любая область экрана рассматривается как адресат, который может быть активизирован при подведении курсора и нажатии клавиши мыши. По реакции на воздействие различают следующие типы адресатов:

- указание и выбор (развертывание пиктограмм, определение активного окна и т. п.);
- буксировка и «резиновая нить» (перенос объекта или его границ);
- экранные кнопки и «скользящие» барьеры

Компоненты ввода-вывода. Как уже упоминалось, в окнах приложения могут размещаться специальные компоненты, используемые для ввода-вывода информации. Интерфейс практически любого современного программного обеспечения включает несколько меню: основное или «ниспадающее» иерархическое меню, пиктографические меню (панели инструментов) и контекстные меню для разных ситуаций. Любое из указанных меню представляет собой компонент ввода-вывода, реализующий диалог с пользователем, используя табличную форму.

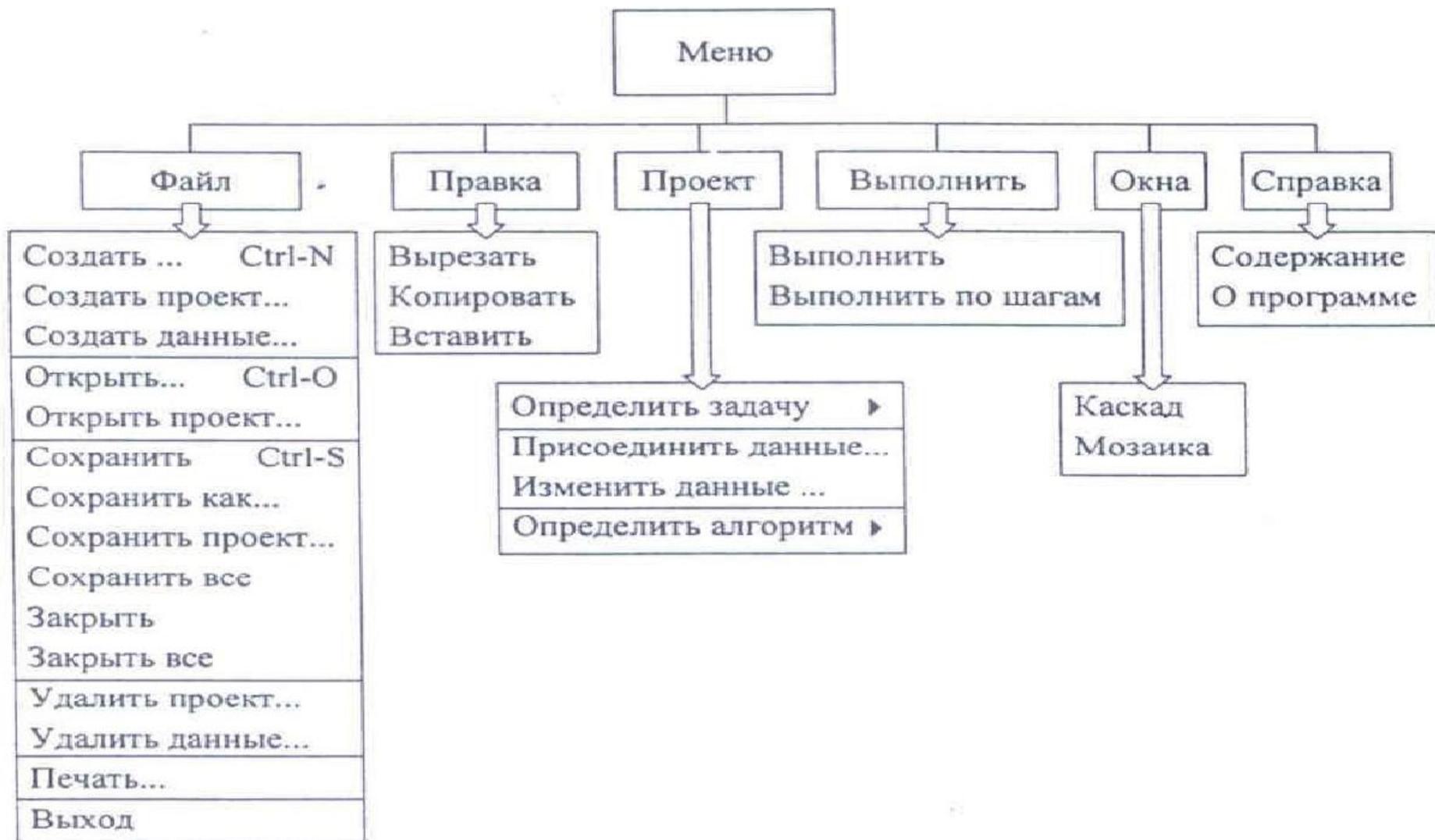
Кроме меню в интерфейсе используют и другие компоненты ввода-вывода, которые можно разделить на три группы в соответствии с тем, какую форму диалога они реализуют: фразовую, табличную или смешанную. Директивная форма диалога обычно предполагает ввод комбинаций клавиш или перемещение пиктограмм, а потому не требует

Компонент	Внешний вид	Реализуемая форма	Особенности использования
Label – метка		Фразовая	Вывод сообщения, как правило, неизменяемого в процессе работы
Edit – однострочный редактор		Фразовая	Ввод-вывод недлинных сообщений: слов, чисел и т. п.
Memo – многострочный редактор		Фразовая	Ввод-вывод текстовой информации
Button – кнопка		Табличная	Инициация операции
CheckBox – элемент выбора		Табличная	Выбор или отмена опций
RadioGroup – группа радиокнопок		Табличная	Выбор одного из вариантов, если их число не превышает 6–10
ListBox – список		Табличная	Выбор одного или нескольких вариантов
TrackBar – указатель		Табличная	Выбор числового значения из заданного интервала
TabControl, PageControl – закладки		Табличная	Улучшение навигации при большом количестве параметров
DBNavigator – навигатор		Табличная	Навигация по таблицам
ComboBox – список с возможностью добавления		Комбинированная	Выбор из списка или ввод значения
SpinEdit – однострочный редактор с возможностью увеличения и уменьшения значения		Комбинированная	Ввод или изменение значения

Реализация диалогов в графическом пользовательском интерфейсе

Для реализации диалогов, управляемых пользователем, применяют меню различных видов: основное, панели инструментов, контекстные и кнопочные, т. е. сформированные из отдельных кнопок. Как альтернативу меню целесообразно использовать директивную форму диалога, поставив в соответствие основным командам определенные к Меню проектируют на основе графов диалогов разрабатываемого программного обеспечения. При этом, если число операций не превышает 5, то обычно используют кнопки. Если число операций не более 9-10, то - одноуровневое меню. И, наконец, если число реализуемых операций более 10, то используют «ниспадающее» двухуровневое иерархическое меню комбинации клавиш.

Ниспадающее меню. Первый уровень иерархического меню должен содержать имена основных групп операций.
Количество уровней иерархического меню не должно



Меню системы решения комбинаторно-оптимизационных задач

Пользовательские интерфейсы прямого манипулирования и их проектирование

Возможность прямого манипулирования, предусмотренная в WIMP интерфейсах, позволяет разрабатывать для приложений объектно-ориентированные интерфейсы прямого манипулирования.

Интерфейсы данного типа на внешнем уровне используют директивную форму диалога: ввод команды осуществляется при выполнении определенных действий с пиктограммой объекта мышью. Основными элементами этих интерфейсов являются: метафоры, объекты, представления объектов и технология Drag and Drop («перетащил и бросил»).

Метафоры. Метафора - мысленный перенос свойств или признаков одного объекта на другой, чем-то аналогичный первому. Использование метафор в интерфейсах предполагает активизацию имеющегося у пользователя опыта

Метафоры и анимация. При реализации метафор все большая роль уделяется средствам мультимедиа, в основном анимации. Движение привлекает внимание, а резкая смена кадров требует некоторого времени на определения связи данного кадра с предыдущим и на изучение этого кадра. Следовательно, используя мультипликацию, можно не только развлекать пользователя, но и «готовить» его к смене кадров, сокращая время, необходимое на адаптацию к изменившейся ситуации.

Объекты интерфейса прямого манипулирования и их представления.

Существует три основных типа объектов интерфейсов прямого манипулирования: объекты-данные, объекты-контейнеры и объекты-устройства.

Объекты-данные снабжают пользователя информацией. Это могут быть тексты, изображения, электронные таблицы, музыка, видео и т. п., а также любая их комбинация. В рамках операционной системы таким объектам соответствуют приложения, которые запускаются при раскрытии объекта. В масштабе приложения объекту соответствует одна или несколько форм, в которых содержимое объекта представляется в разных видах. Операции с содержимым объекта реализуются обработчиками событий формы.

Объекты-контейнеры могут манипулировать своими внутренними объектами, в том числе и другими контейнерами, например, копировать их или сортировать в любом порядке. К типичным контейнерам относятся папки, корзины т. п. При раскрытии контейнера демонстрируются сохраняемые им компоненты, и появляется возможность ими манипулировать. Компоненты при этом могут обозначаться пиктограммами или представляться в виде таблицы.

Объекты-устройства часто представляют устройства, существующие в реальном мире: телефоны, факсы, принтеры и т. д., их используют для обозначения этих устройств в абстрактном мире интерфейса. При раскрытии такого объекта, как правило, можно увидеть его настройки. Каждому объекту соответствует, по крайней мере, одно окно. В исходном состоянии это окно представлено пиктограммой, но при необходимости его можно раскрыть и выполнить требуемые операции, например настройки объекта. Окно объекта в раскрытом состоянии может содержать меню и панели инструментов. Пиктограмме же должно соответствовать контекстное меню, содержащее перечень операций над объектом.

Имя пиктограммы формируют по-своему для каждого типа объектов. Так пиктограммам объектов-данных присваивают имена, соответствующие именам хранимых данных, а тип данных кодируется самой пиктограммой. Имя пиктограммы-контейнера или пиктограммы устройства обозначает сам объект, а потому не зависит от содержимого.

Следует иметь в виду, что различие между типами объектов является условным

Технология Drag and Drop. Технология Drag and Drop («перетащил и бросил») определяет основные принципы прямого манипулирования, описанные в руководстве по разработке пользовательских интерфейсов фирмы IBM (CUA - Common User Access):

- результат перемещения объекта должен соответствовать ожиданиям пользователя;
- пользователи не должны неожиданно терять информацию;
- пользователь должен иметь возможность отменить неправильное действие.

Эта технология также определяет основные принципы визуализации операции прямого манипулирования:

- исходное выделение - используется в качестве обратной связи пользователю, чтобы сообщить ему, что объект захвачен, в Windows с этой целью используется выделение цветом;
- визуализация перемещения - используется для идентификации выполняемого действия;
- целевое выделение - используется для идентификации пункта назначения, показывая, таким образом, куда «упадет» объект, если его отпустить в текущий момент времени;
- визуализация действия - используется для обозначения времени ожидания завершения операции, обычно с этой целью применяют анимацию или изменение формы курсора на «песочные часы».

Следует также иметь в виду, что существует два вида пунктов назначения: один принимает объект, а другой его копию.

Проектирование интерфейсов прямого манипулирования. Проектирование интерфейсов прямого манипулирования выполняется на основе графов диалога, разработанных для конкретного программного обеспечения, и включает следующие процедуры:

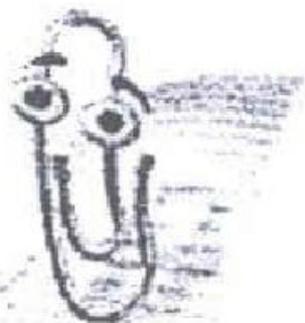
- формирование множества объектов предметной области, которое должно быть представлено на экране, причем в качестве основы в этом случае используют не варианты использования, а концептуальную модель предметной области;
- анализ объектов, определение их типов и представлений, а также перечня операций с этими объектами;
- уточнение взаимодействия объектов и построение матрицы прямого манипулирования;
- определение визуальных представлений объектов;
- разработка меню окон объектов и контекстных меню;
- создание прототипа интерфейса;

Интеллектуальные элементы пользовательских интерфейсов

В последние годы появилось много новых перспективных элементов пользовательских интерфейсов, в основном приносящих в интерфейс элементы искусственного интеллекта, что проявляется в их названиях: **Мастер, Советчик, Агент**.

Сделано множество попыток создания социализированного пользовательского интерфейса. В основе такого интерфейса лежит идея создания персонифицированного, т. е. «имеющего личность», интерфейса. Развлекающие программы, такие как Cats (Кошки) и Dogs (Собаки), реализующие достаточно сложное поведение домашних животных в разных ситуациях, показывают, что технически это вполне решаемая задача. Однако в этой области существуют психологические проблемы. В качестве примера вспомним, что даже «безобидный» Советчик Microsoft Office вызывает у многих пользователей резко отрицательную реакцию. Пока попытки создания такой «личности» успеха не имели.

Советчики. Советчики представляют собой форму подсказки. Обычно их можно вызвать с помощью меню справки, командной строки окна или из всплывающего меню. Советчики помогают пользователям в выполнении конкретных задач, но только, если пользователь представляет, что ему нужно сделать.



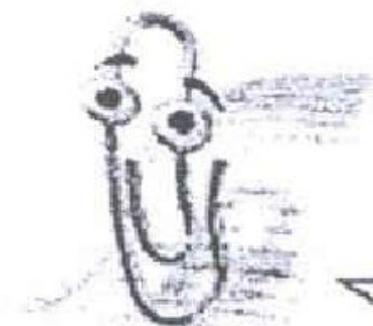
Действие

Введите свой вопрос и нажмите кнопку "Найти".

Параметры

Найти

а



Действие

- Вставка рисунка
- Разрешение вопросов: работа со сканером и камерой
- Разрешение вопросов: работа с рисунками
- Дополнительные места хранения картинок, рисунков, звуков, видеоклипов и шаблонов
- Добавление автофигуры, круга или квадрата
- ▼ См. далее...

Как вставить рисунок

Параметры

Найти

б

Помощник-Скрепка Microsoft Word 2000:

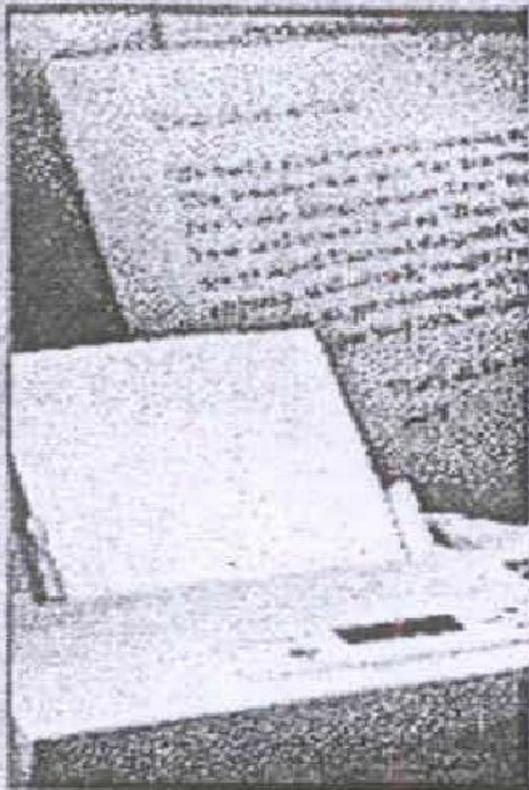
а – запрос; *б* – список связанных тем

Мастера. Программу-мастер используют для выполнения общераспространенных, но редко выполняемых отдельным пользователем задач, таких, как установка программ или оборудования. Выполнение подобных действий требует от пользователя принятия сложных взаимосвязанных решений, последовательность которых и диктует программа-мастер. Интеллектуальные Мастера способны на каждом шаге демонстрировать в окне просмотра результаты ответов пользователя на предыдущие вопросы, помогая последнему сориентироваться в ситуации.

Мастер реализует последовательный или древовидный сценарий диалога, поэтому его целесообразно использовать для решения хорошо структурированных, последовательных задач (рис. 26). При этом необходимо:

- предоставить пользователю возможность возврата на предыдущий шаг;
- предусмотреть возможность отмены работы Мастера;
- нумеровать шаги и сообщать пользователю количество шагов Мастера, особенно, если таких шагов больше трех;
- пояснять пользователю каждый шаг;
- по возможности демонстрировать результат уже выполненных операций на каждом шаге.

Мастер установки принтера



Данный мастер поможет легко и быстро установить драйвер нужного принтера.

Чтобы начать установку, нажмите кнопку "Далее".

< Назад

Далее >

Отмена

Первое окно мастера Установка принтера Windows' 98

Программные агенты. Наибольший интерес на настоящий момент представляют программные агенты, используемые для выполнения рутинной работы. Такой программный агент является элементом программного обеспечения, которому пользователь может передать часть своих обязанностей. Основными функциями Агентов-Помощников являются: наблюдение, поиск и управление. Различают:

- программы-агенты, настраиваемые на выполнение указанных задач;
- программы-агенты, способные обучаться, например, фиксируя действия пользователя (по типу магнитофона). Создание агентов последнего типа, например, доступно через механизм макросов Microsoft Office. Большинство интересных и достаточно сложных программных агентов в настоящее время «живет» в Интернете, где и можно найти последнюю информацию по данной теме.