



Разработка пользовательских интерфейсов

Системное и прикладное
программное обеспечение

Малышенко Владислав Викторович



Содержание

- Типы пользовательских интерфейсов и этапы их разработки
- Психофизические особенности человека, связанные с восприятием, запоминанием и обработкой информации
- Пользовательская и программная модели интерфейса
- Классификации диалогов и общие принципы их разработки
- Основные компоненты графических пользовательских интерфейсов
- Реализация диалогов в графическом пользовательском интерфейсе
- Пользовательские интерфейсы прямого манипулирования и их проектирование
- Интеллектуальные элементы пользовательских интерфейсов



Пользовательский интерфейс

Пользовательский интерфейс представляет собой совокупность программных и аппаратных средств, обеспечивающих взаимодействие пользователя с компьютером.

Основу такого взаимодействия составляют **диалоги**. Под диалогом в данном случае понимают регламентированный обмен информацией между человеком и компьютером, осуществляемый в реальном масштабе времени и направленный на совместное решение конкретной задачи.

Каждый диалог состоит из отдельных процессов ввода-вывода, которые физически обеспечивают связь пользователя и компьютера.



Пользовательский интерфейс. Обмен информацией.

Обмен информацией осуществляется передачей сообщений и управляющих сигналов.

Сообщение - порция информации, участвующая в диалоговом обмене.

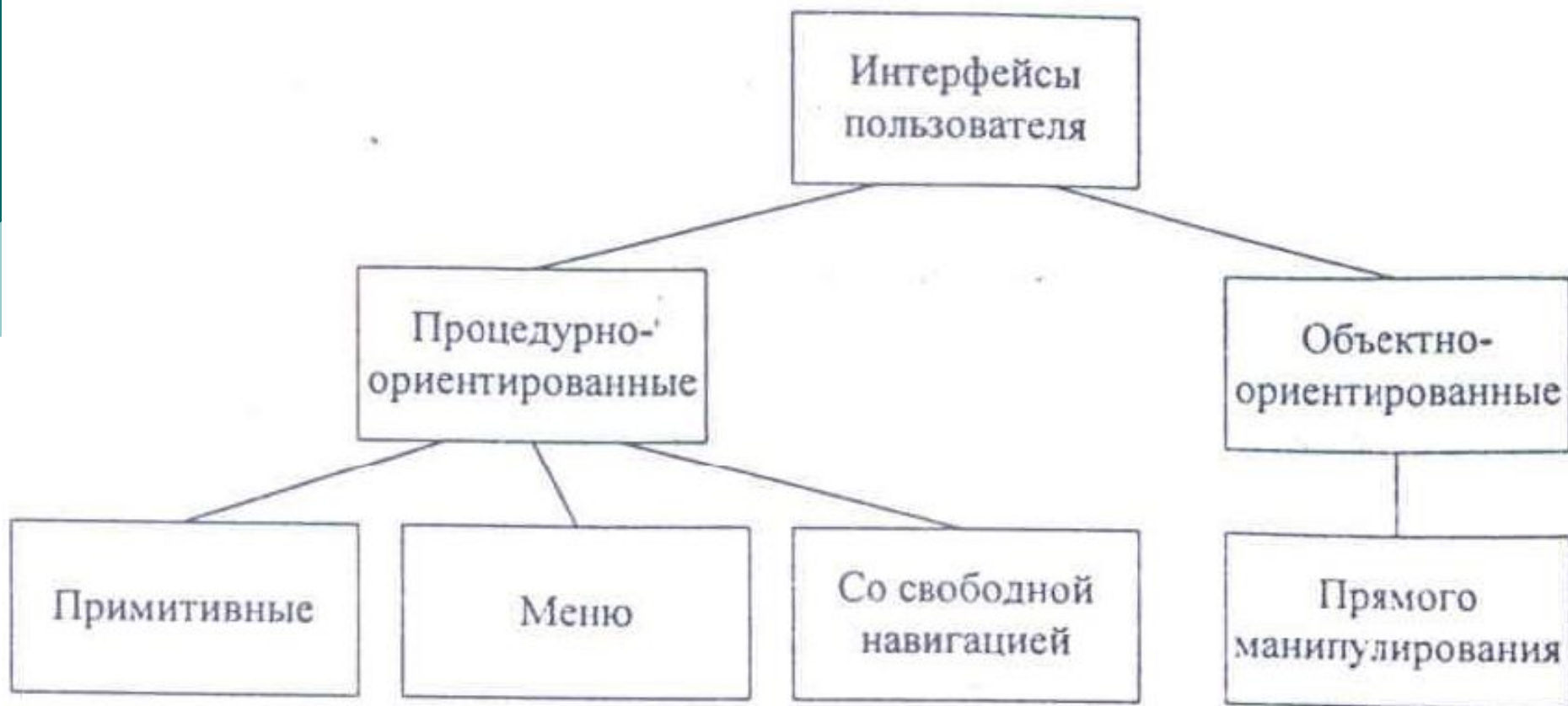
Различают:

- 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.

Организация взаимодействия между компьютером и человеком.



Типы интерфейсов





Типы интерфейсов

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

Примитивный интерфейс

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

Обычно такой интерфейс реализует конкретный сценарий работы программного обеспечения, например: ввод данных - решение задачи - вывод результата.

Подобные интерфейсы в настоящее время используют только в процессе обучения программированию или в тех случаях, когда вся программа реализует одну функцию, например, в некоторых системных утилитах.

Интерфейс-меню позволяет пользователю выбирать необходимые операции из специального списка, выводимого ему программой. Эти интерфейсы предполагают реализацию множества сценариев работы.

Различают **одноуровневые** и **иерархические меню**. Первые используют для сравнительно простого управления вычислительным процессом. Вторые – при большом количестве вариантов или их очевидных различиях, например, операции с файлами и операции с данными, хранящимися в этих файлах.

Интерфейсы данного типа несложно реализовать в рамках структурного подхода к программированию.



Интерфейсы со свободной навигацией

Интерфейсы со свободной навигацией также называют *графическими пользовательскими интерфейсами* (GUI - Graphic User Interface) или интерфейсами WYSIWYG (What You See Is What You Get).

Графические интерфейсы поддерживают концепцию интерактивного взаимодействия с программным обеспечением, осуществляя визуальную обратную связь с пользователем и возможность прямого манипулирования объектами и информацией на экране.

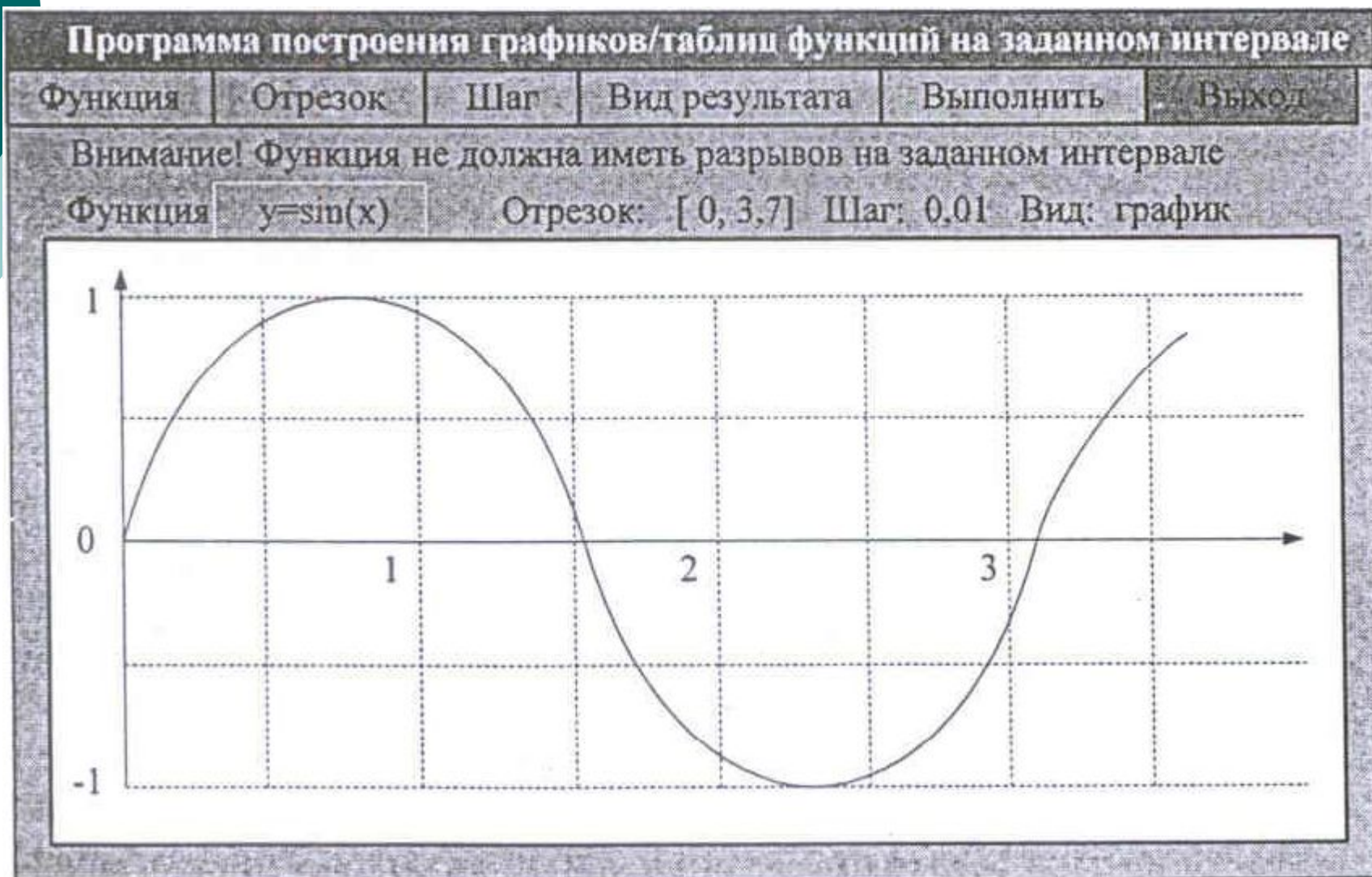
Интерфейсы данного типа поддерживают концепцию совместимости программ, позволяя перемещать между ними информацию.

Интерфейс прямого манипулирования

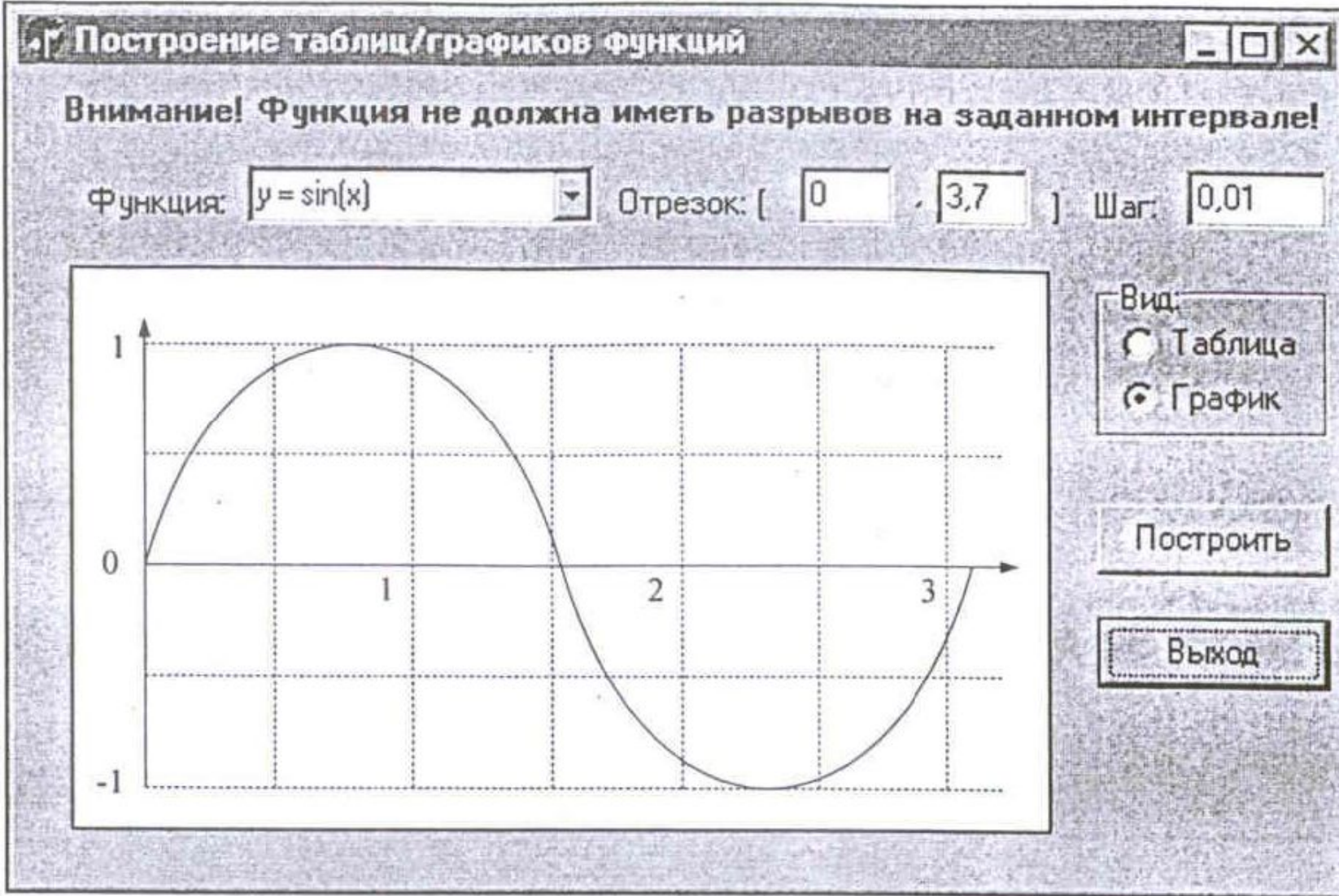
Этот тип интерфейса предполагает, что взаимодействие пользователя с программным обеспечением осуществляется посредством выбора и перемещения **пиктограмм**, соответствующих объектам предметной области.

Для реализации таких интерфейсов также используют событийное программирование и объектно-ориентированные библиотеки.

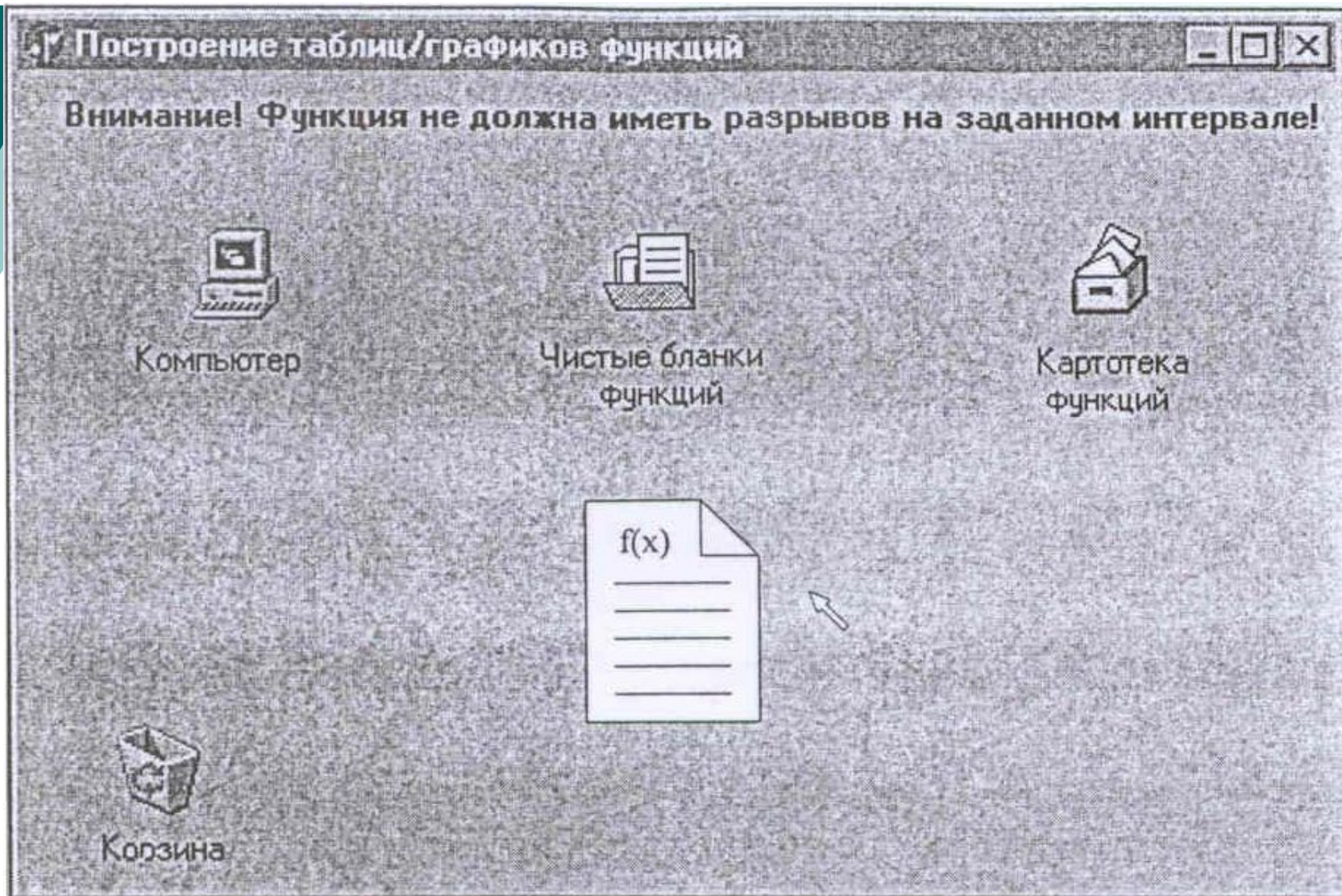
Пример программы построения графиков




Пример программы построения графиков



Пример программы построения графиков





Психофизические особенности человека, связанные с восприятием, запоминанием и обработкой информации

Обработка информации





Особенности восприятия

- Особенности восприятия цвета
- Особенности восприятия звука
- Субъективное восприятие времени



Пользовательская модели интерфейса

- Модель программиста
- Модель пользователя
- Программная модель

Пользовательская модель интерфейса

Пользовательская модель интерфейса - это совокупность обобщенных представлений конкретного пользователя или некоторой группы пользователей о процессах, происходящих во время работы программы или программной системы.

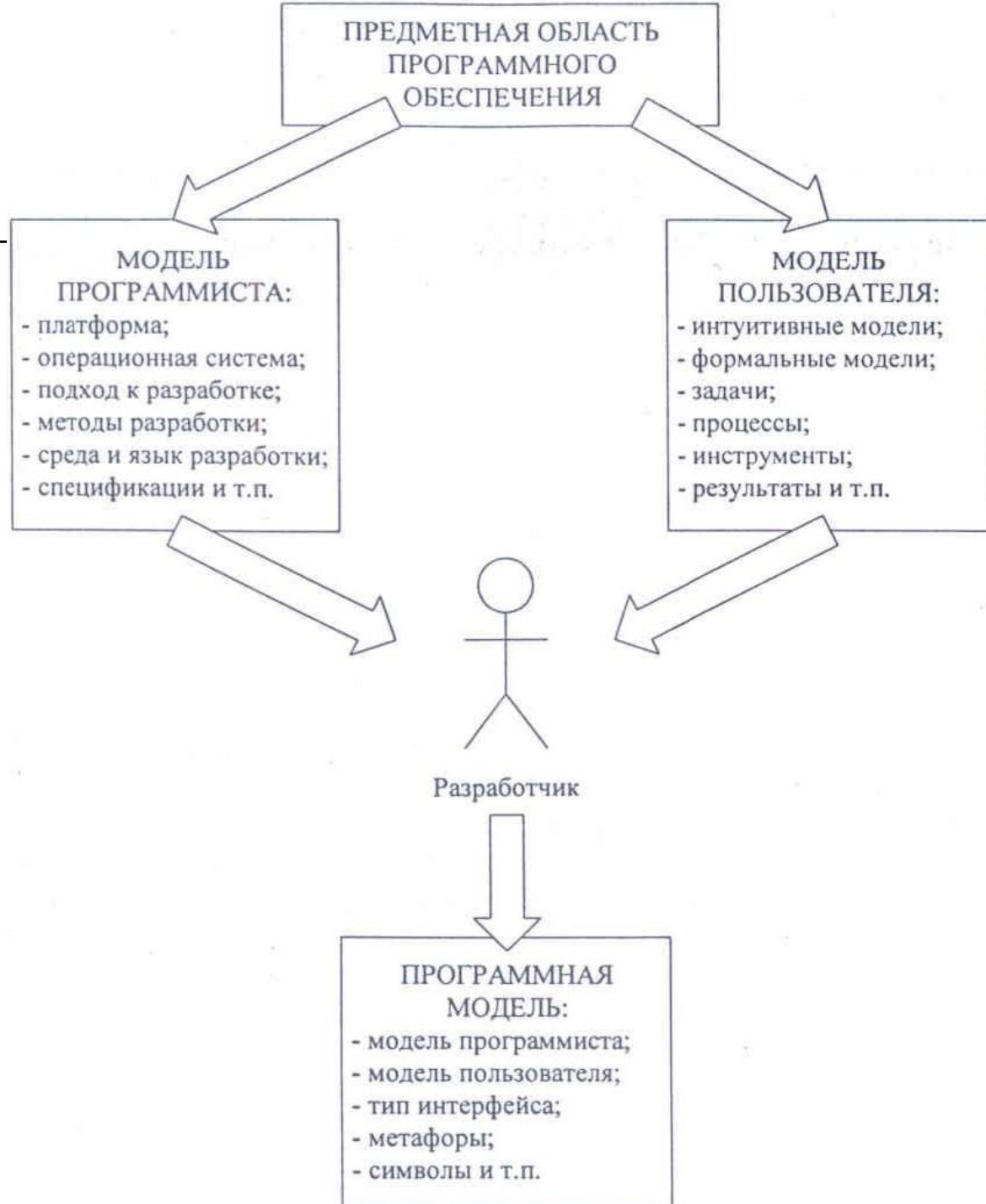
Модель базируется на **опыте пользователей**, который характеризуется:

- 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.

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

диалог - это процесс обмена информацией между пользователем и программной системой, осуществляемый через интерактивный терминал и по определенным правилам.

Различают **тип диалога** и его **форму**.



Типы диалога

Тип диалога определяет, кто из «собеседников» управляет процессом обмена информацией.

Различают два типа диалога:

- управляемые программой;
- управляемые пользователем.

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 обеспечивает возможность реализации различных пользовательских сценариев.



Формы диалога

Описание языка, на котором ведется диалог, включает определение его **синтаксиса** и **семантики**.

В зависимости от вида используемых в конкретном случае синтаксиса и семантики различают три формы диалога:

- Фразовую;
- Директивную;
- Табличную.



Фразовая форма

Директивная форма

Табличная форма

Фразовая форма предполагает «общение» с пользователем на естественном языке или его подмножестве. Содержание диалога в данной форме составляют вопросы и ответы.

Общение может осуществляться в свободном формате, но возможна и фиксация отдельных фраз.

Организация диалога на естественном языке на современном уровне - задача не решенная.



Фразовая форма

Директивная форма

Табличная форма

Пример:

Программа: Введите свой возраст (полных лет):

Пользователь: 48.

Программа содержит описание синтаксиса и семантики используемого **ограниченно-естественного языка**.

Для данного примера достаточно определить синтаксис понятия «целое положительное число» и наложить ограничение на значение числа.



Фразовая форма

Директивная форма

Табличная форма

Создаются интерфейсы на базе **ограниченного подмножества предложений** естественного языка в основном для интеллектуальных систем.

При обработке фраз в этих случаях оперируют понятием словоформа. **Словоформа** – отрезок текста между двумя соседними пробелами или знаками препинания.

Обработка словоформ вне связи с контекстом называется **морфологическим анализом**.



Фразовая форма (морфологический анализ)

Директивная форма

Табличная форма

декларативный анализ –

предполагает, что в словаре находятся все возможные словоформы каждого слова, тогда анализ сводится к поиску словоформы в словаре.

Данный метод обеспечивает возможность обработки сообщений, состоящих из строчных и прописных букв в произвольной комбинации, причем как латинского, так и русского или других алфавитов;

процедурный анализ –

предполагает выделение в текущей словоформе основы, которую затем идентифицируют.



Фразовая форма (достоинства и недостатки)

Директивная форма

Табличная форма

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

Основными *недостатками* фразовой формы при использовании являются:

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



Директивная форма –

предполагает использование команд **специально разработанного формального языка.**

Командой в этом случае называют предложение этого языка.

Команду можно вводить:

- в виде строки текста, специально разработанного формата;
- нажатием некоторой комбинации клавиш клавиатуры;
- посредством манипулирования мышью;
- комбинацией второго и третьего способов.



Фразовая форма

Директивная форма (достоинства и недостатки)

Табличная форма

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.



Фразовая форма

Директивная форма (достоинства и недостатки)

Табличная форма

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;
- необходимость навыков ввода текстовой информации или манипуляций мышью;
- отсутствие возможности настройки пользователем.



Фразовая форма
Директивная форма
Табличная форма

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.

Удобно пользователю, так как выбрать всегда проще.

Однако применение табличной формы возможно не всегда: ее можно использовать только, если множество возможных ответов на конкретный вопрос конечно.

Причем, если количество возможных ответов велико (более 20), то применение табличной формы может оказаться нецелесообразным.



Фразовая форма

Директивная форма

Табличная форма (достоинства и недостатки)

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.



Фразовая форма

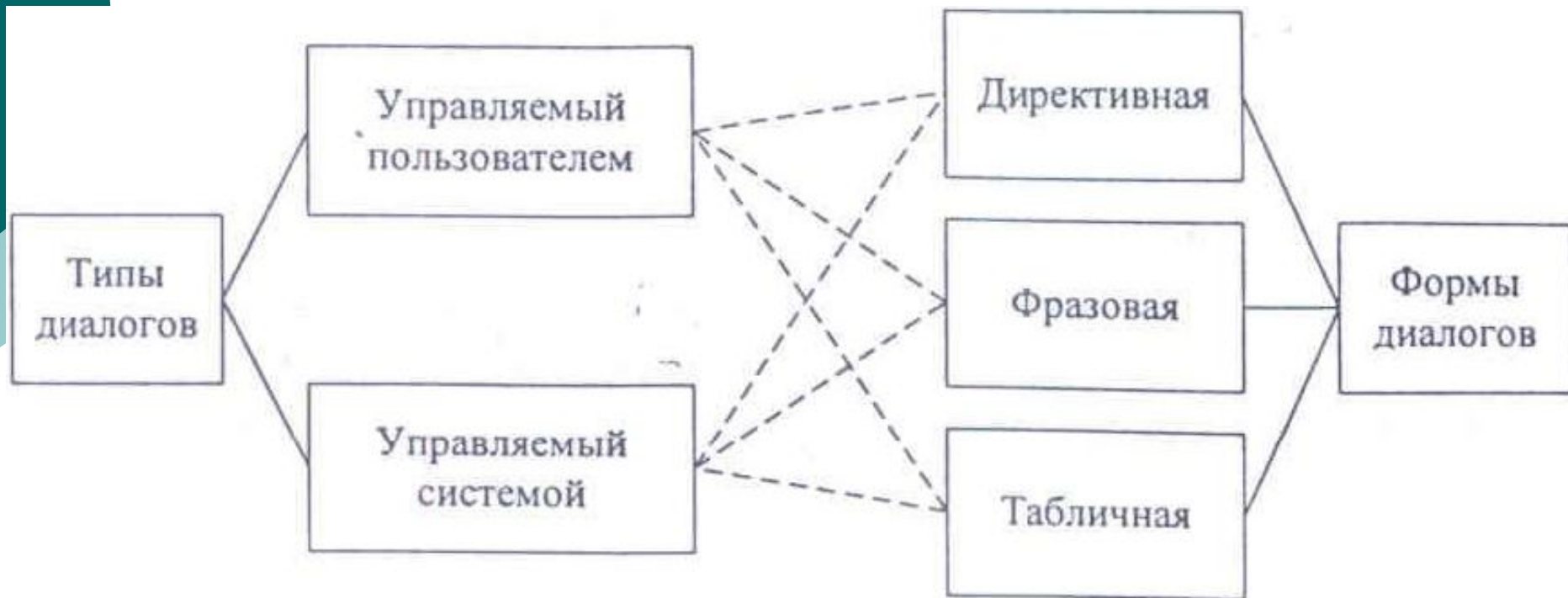
Директивная форма

Табличная форма (достоинства и недостатки)

Недостатками табличной формы являются:

- 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:

- проектирование *абстрактных диалогов*;
- проектирование *конкретных диалогов*;
- проектирование *технических 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*.

Диаграммы состояний интерфейса

Граф диалога — ориентированный взвешенный граф, каждой вершине которого сопоставлена конкретная картинка на экране (*кадр*) или определенное состояние диалога, характеризующееся набором доступных пользователю действий.

Дуги, исходящие из вершин, показывают возможные изменения состояний при выполнении пользователем указанных действий. В качестве весов дуг указывают условия переходов из состояния в состояние и операции, выполняемые во время перехода.

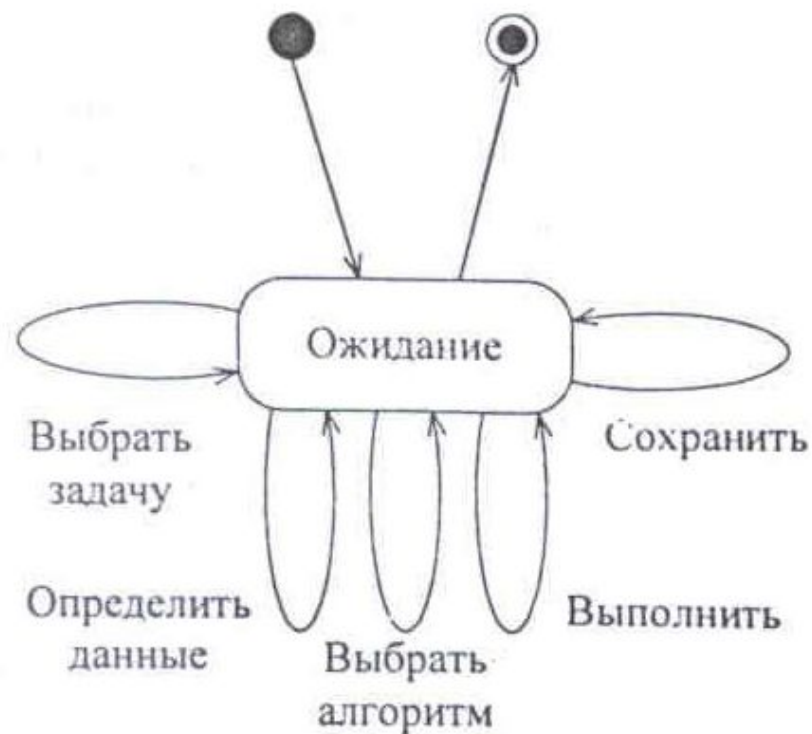
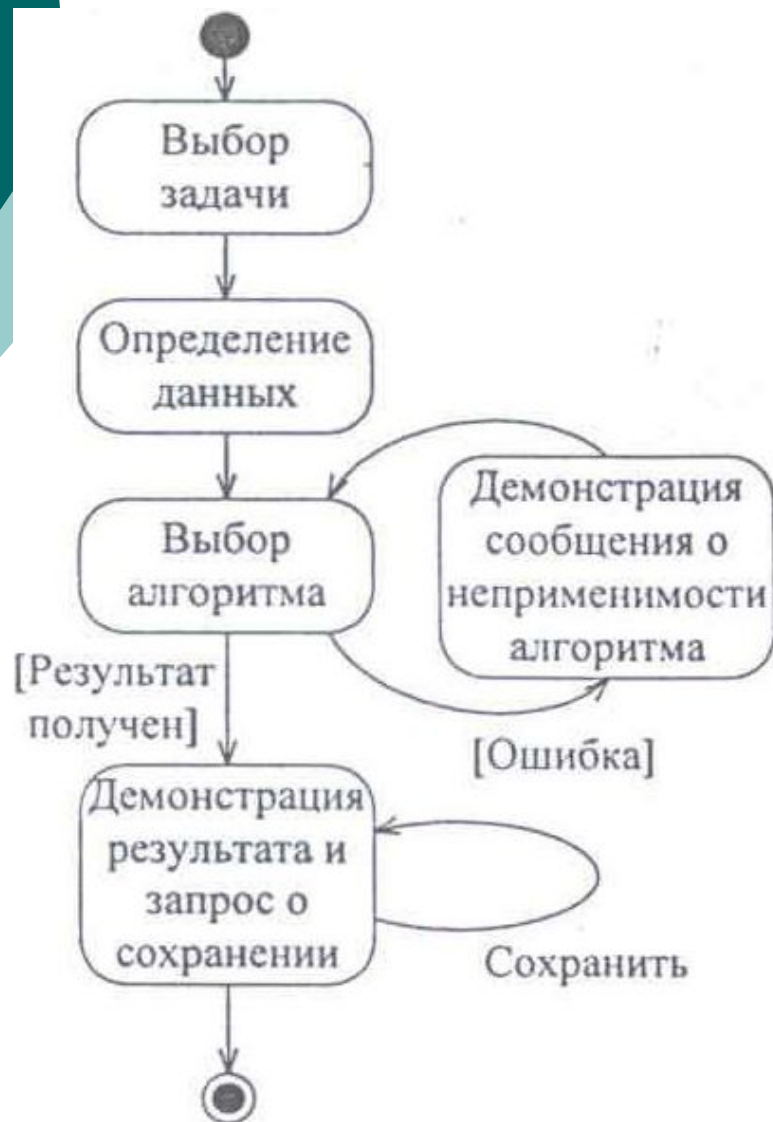


Пример. Разработать граф диалога для системы решения комбинаторно-оптимизационных задач.

Граф абстрактного диалога системы решения комбинаторно-оптимизационных задач



Граф абстрактного диалога «Новое задание»







Содержание

- Типы пользовательских интерфейсов и этапы их разработки
- Психофизические особенности человека, связанные с восприятием, запоминанием и обработкой информации
- Пользовательская и программная модели интерфейса
- Классификации диалогов и общие принципы их разработки
- Основные компоненты графических пользовательских интерфейсов
- Реализация диалогов в графическом пользовательском интерфейсе
- Пользовательские интерфейсы прямого манипулирования и их проектирование
- Интеллектуальные элементы пользовательских интерфейсов



Основные компоненты графических пользовательских интерфейсов

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).

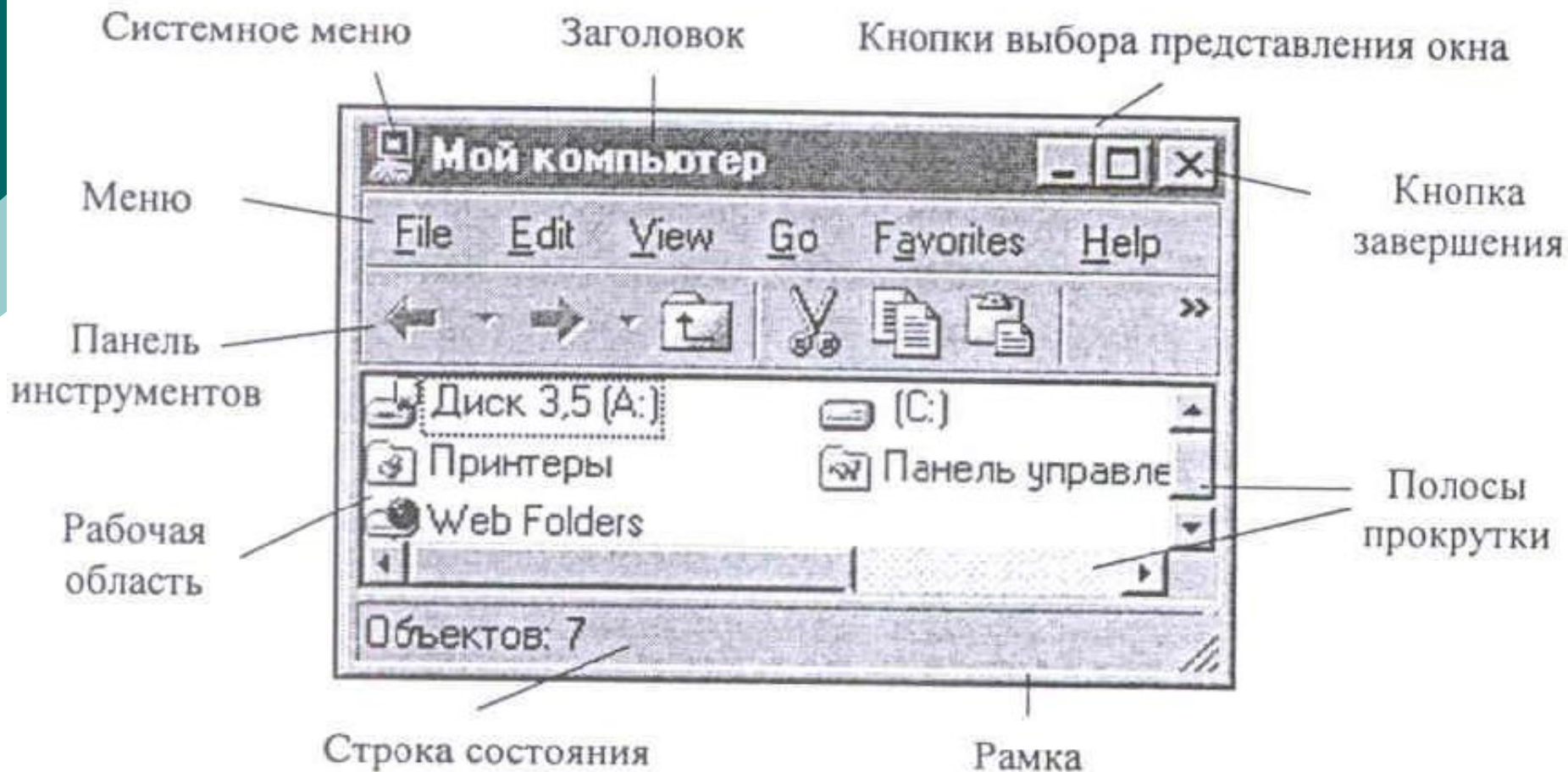
Основными элементами графических интерфейсов, таким образом, являются: окна, пиктограммы, компоненты ввода-вывода и мышь.

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.

Окно приложения и его элементы



Дочернее окно Windows

Дочернее окно Windows используют в

многодокументных программных интерфейсах

(МДПИ) программное

обеспечение несколькими

документами

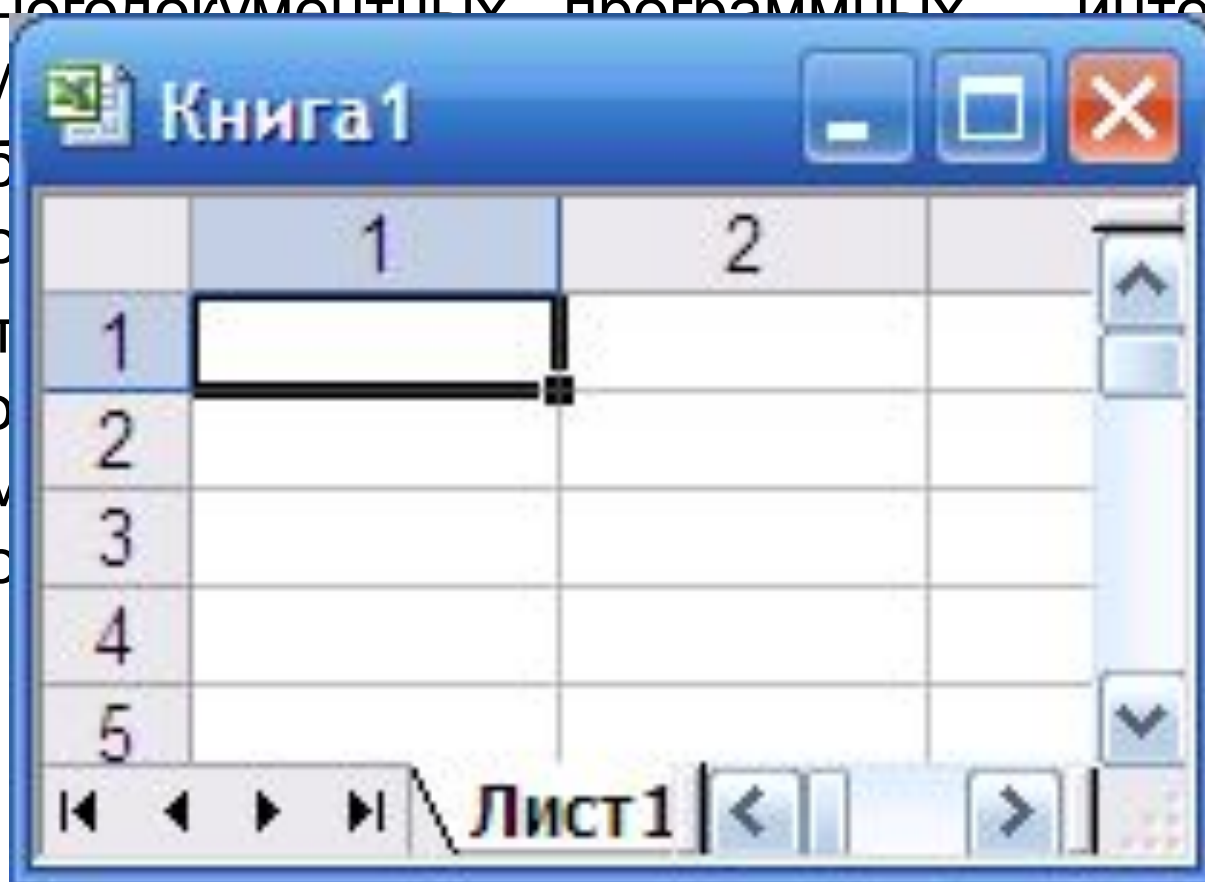
В отличие от главного окна не

содержит панель меню

и имеет собственную панель

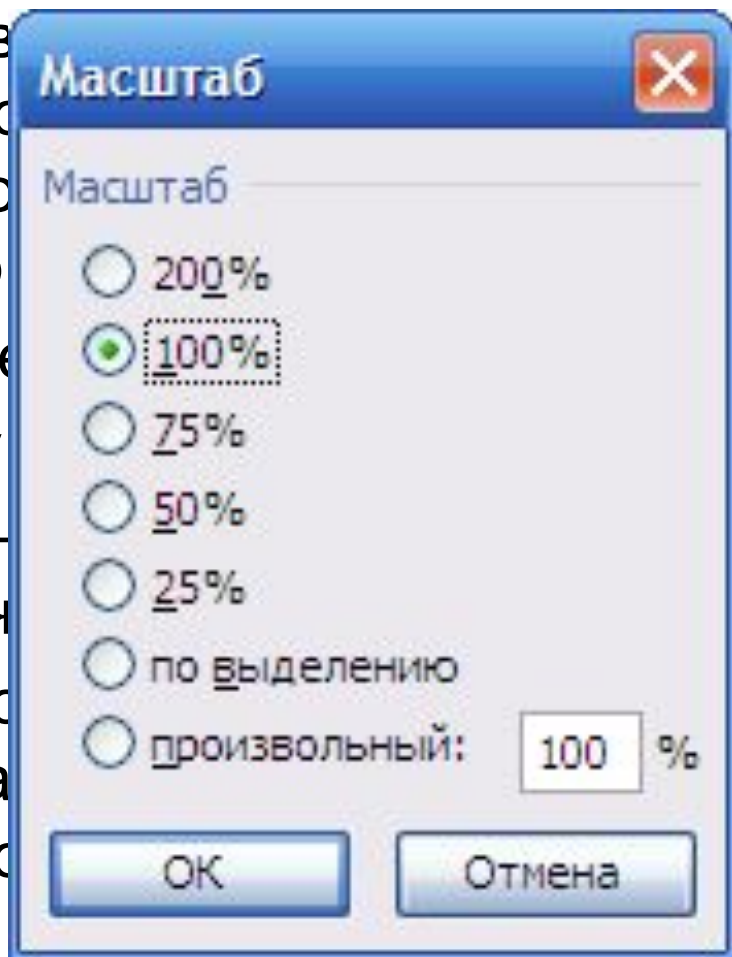
инструментов

документа



Диалоговое окно Windows

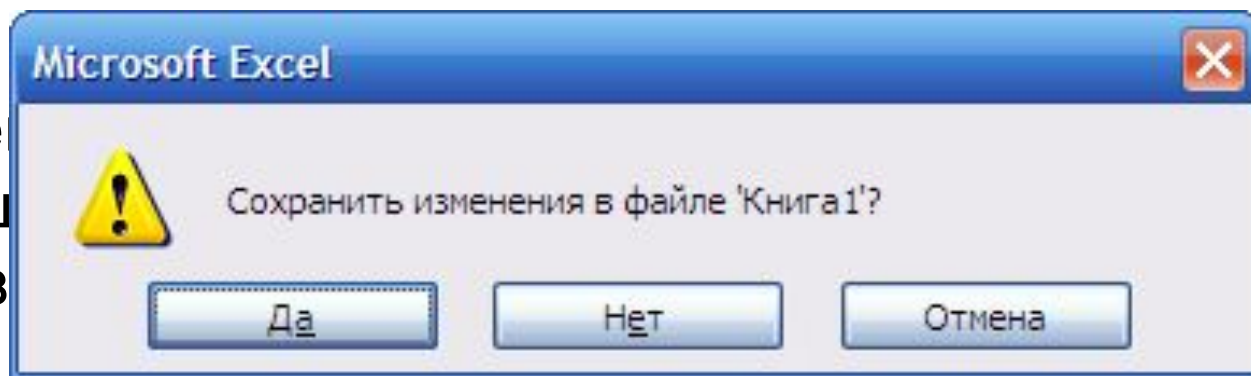
- Диалоговое окно Масштаб используется для изменения размера элементов интерфейса (кнопок, меню, текста) в различных режимах работы, таких как стандартный, адаптивный или другой.
- Оно может использоваться для:
- строку меню;
 - компонент интерфейса;
 - возможность выбора ответа;
 - вспомогательные подсказки, обеспечивающие предварительную проверку.



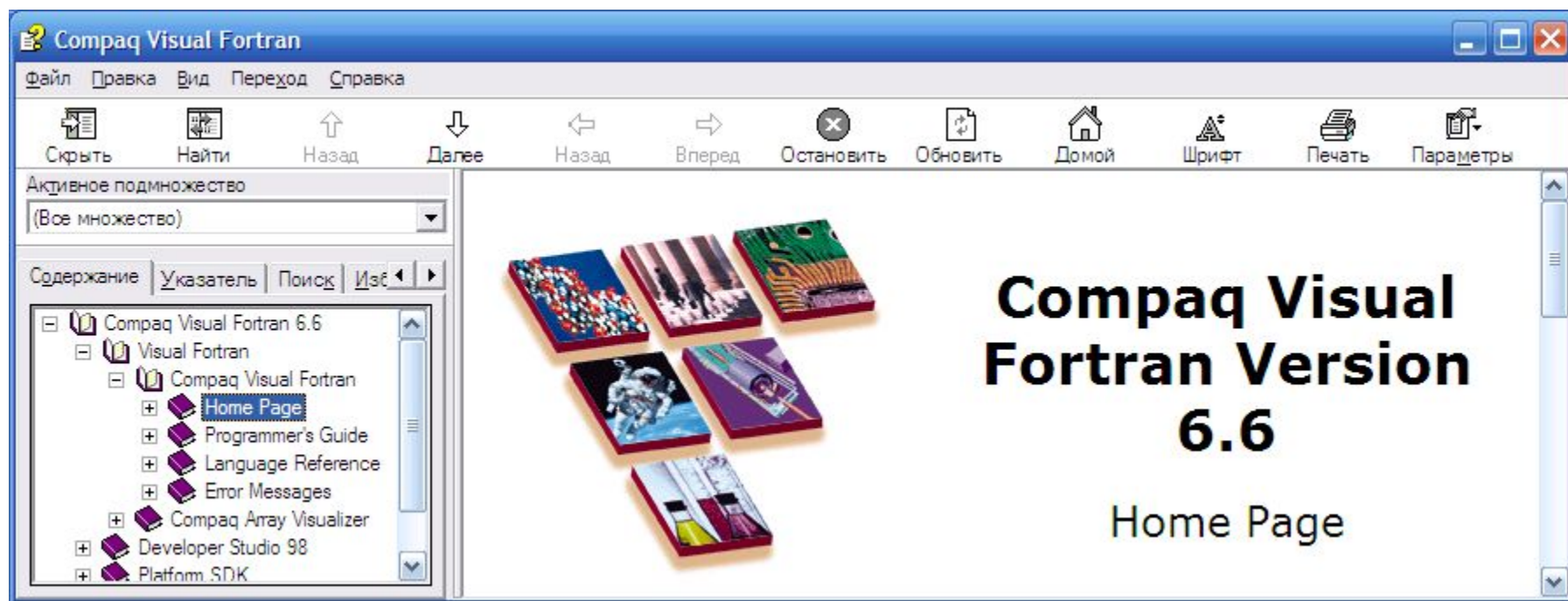


Информационные окна

- Окна системы сообщают о событиях, происходящих в системе.

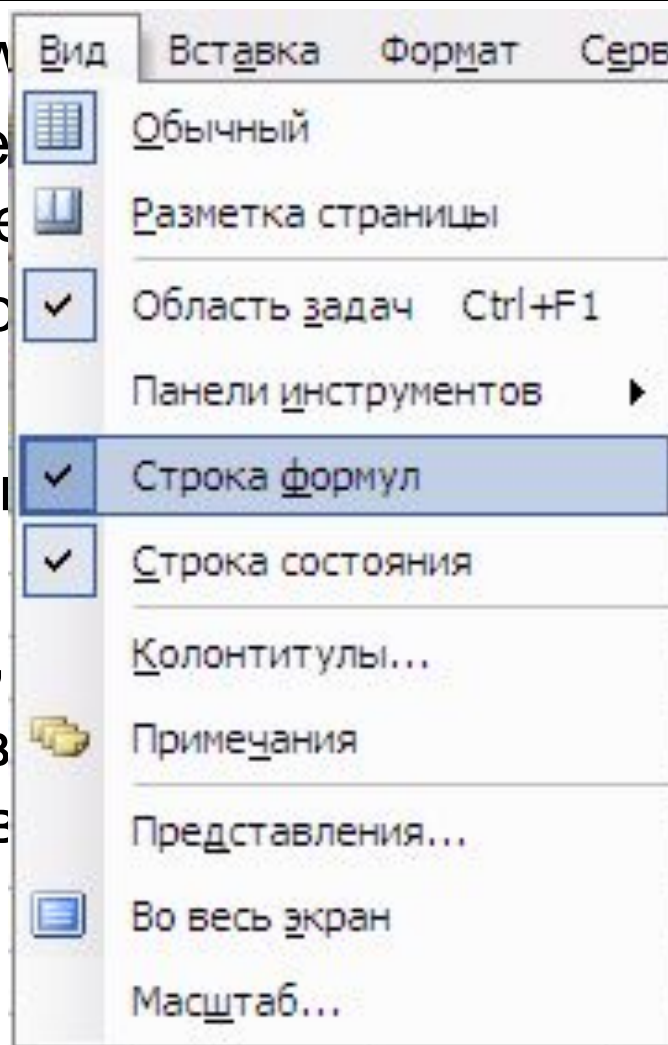


кой
ЭКСТ
ЦИИ
el.



Окна меню Windows

- Окна меню Windows открываются, как отдельные программы. Каждой строке соответствует:
- команда;
 - меню следующего уровня, обозначается стрелкой;
 - окно диалога, обозначается точками.
- Кроме того, в меню добавляется



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.
- Многодокументная программная система управляет **пиктограммами дочерних окон**, обеспечивающими 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.

Прямое манипулирование изображением – это возможность замены команды воздействия на некоторый объект физическим действием в интерфейсе, осуществляемым с помощью мыши.

По реакции на воздействие различают следующие типы адресатов:

- указание и выбор;
- буксировка и «резиновая нить»;
- экранные кнопки и «скользящие» барьеры.



Прямое манипулирование изображением. Динамические визуальные сигналы

Основная цель этих сигналов заключается в предоставлении пользователям дополнительной информации.

Пример:

- изменение изображения курсора мыши при выполнении конкретных операций.
- изменение изображения кнопки при нажатии на нее.

Компоненты ввода-вывода

В интерфейсе используют компоненты ввода-вывода, которые можно разделить на три группы в соответствии с тем, какую форму диалога они реализуют:

- Фразовую;
- Табличную;
- Смешанную.

Директивная форма диалога обычно предполагает ввод комбинаций клавиш или перемещение пиктограмм, а потому не требует использования компонентов ввода-вывода.



Реализация диалогов в графическом пользовательском интерфейсе

диалоги управляемые пользователем
и
диалоги управляемые системой.



Реализация диалогов, управляемых пользователем

Для реализации диалогов, управляемых пользователем, применяют меню различных видов:

- основное,
- панели инструментов,
- контекстные и кнопочные.

Меню

проектируют на основе графов диалогов разрабатываемого программного обеспечения.

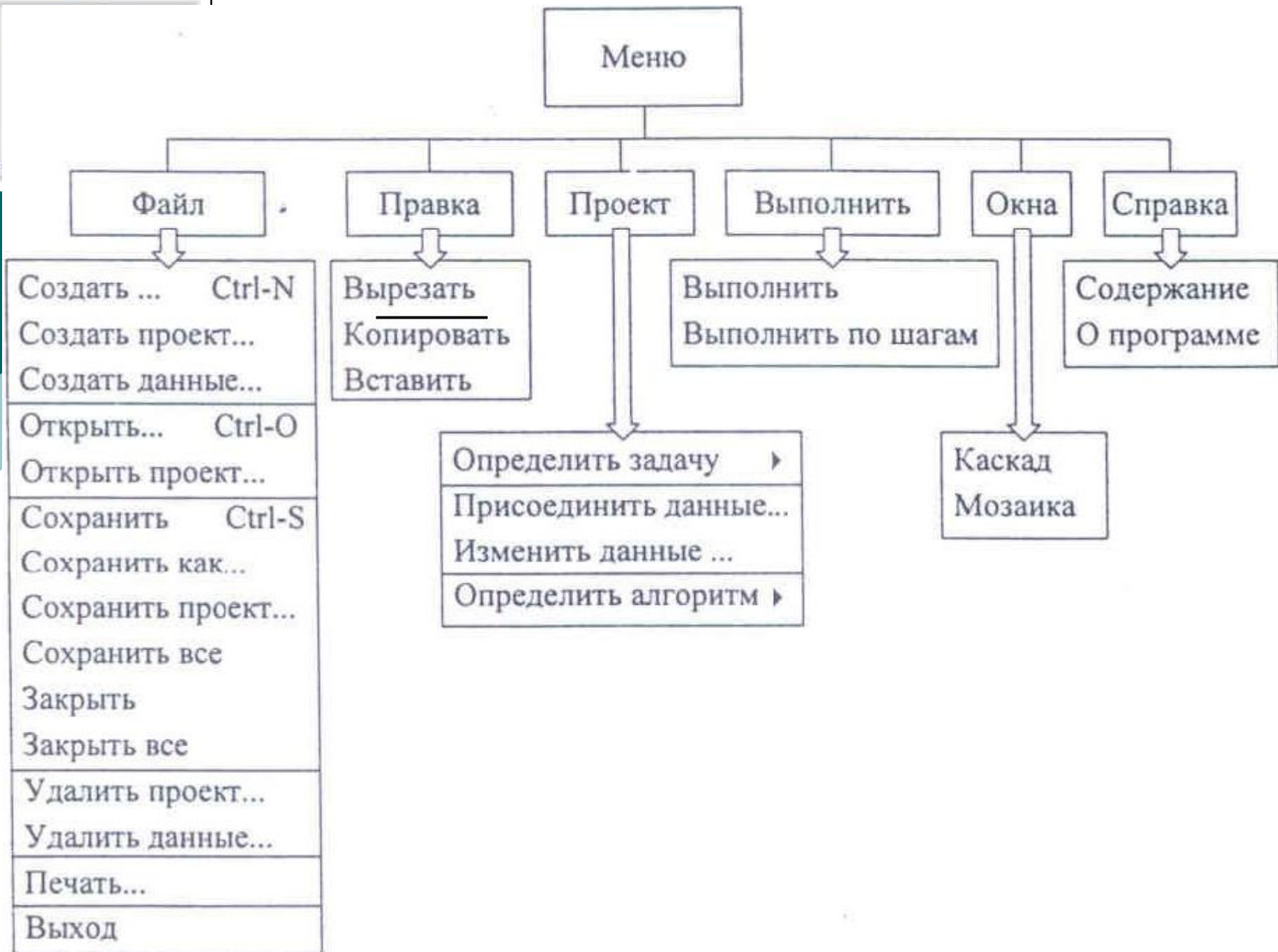
- ✓ Если число операций не превышает **5**, то обычно используют **кнопки**.
- ✓ Если число операций не более **9-10**, то - **одноуровневое меню**.
- ✓ Если число реализуемых операций **более 10**, то используют **«ниспадающее»** двухуровневое иерархическое **меню**.

Первый уровень иерархического меню должен содержать имена основных групп операций.

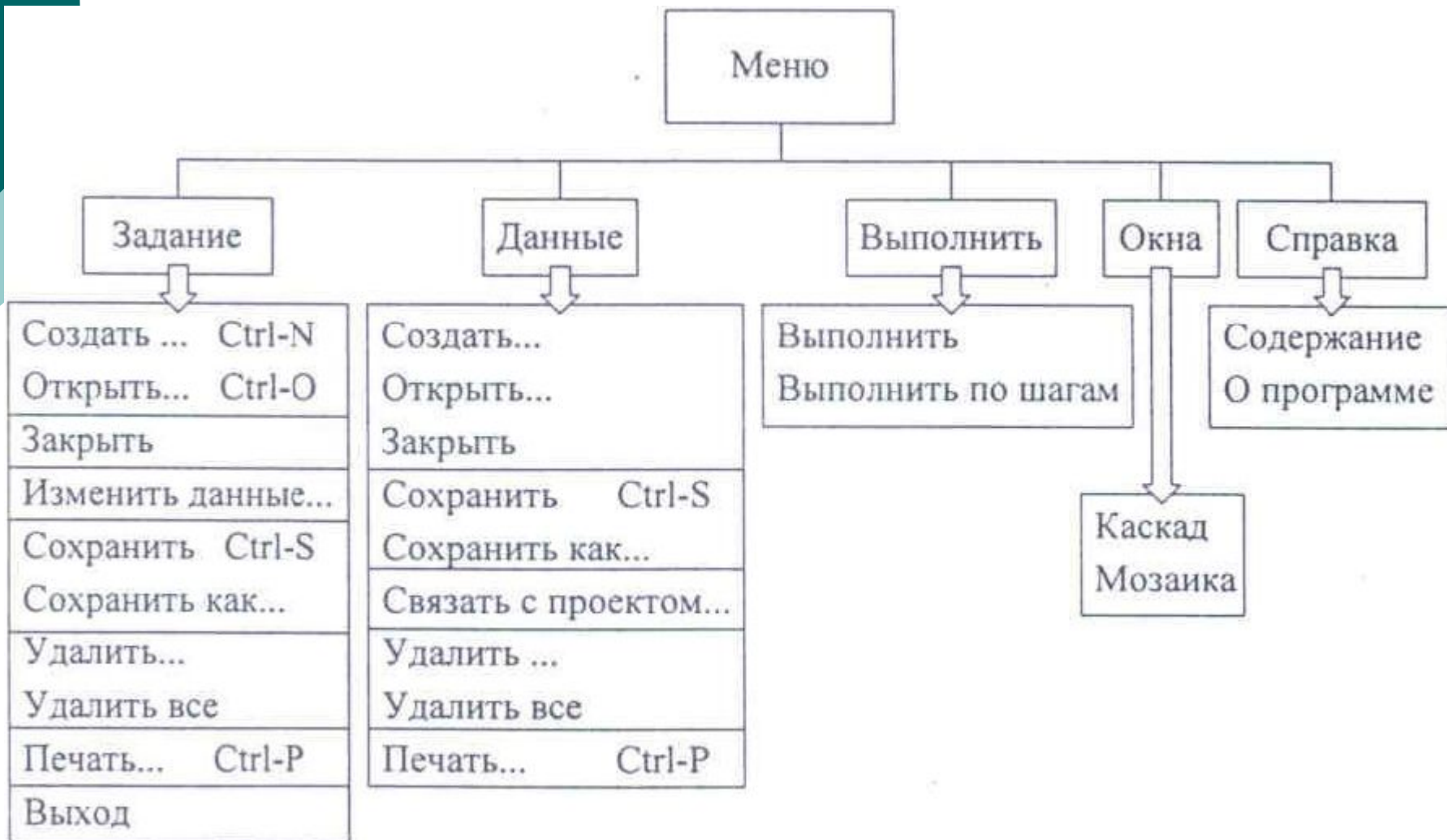
Традиционно первым является пункт Файл, вторым - Правка, третьим - Вид, а последним - Справка.

Количество уровней иерархического меню не должно превышать 2-3, так как при большем числе уровней требуемую операцию будет сложно искать.

Желательно, чтобы число операций в окне меню не превышало 7-8.



Пример №2. Разработать основное меню системы решения комбинаторно-оптимизационных задач.



Панель инструментов

На панель инструментов помещают пиктограммы часто используемых операций.

Если множество таких операций существенно зависит от специфики выполняемых с разрабатываемым программным обеспечением работ, то целесообразно обеспечить пользователю возможность формирования панелей инструментов по собственному усмотрению.

Контекстные меню включают операции, вероятность обращения к которым из данной зоны окна приложения с точки зрения разработчика максимальна.

В процессе тестирования «удобства использования» содержание контекстного меню может уточняться.

Желательно, чтобы число операций этого меню превышает 6-8.

Целесообразно операции контекстного меню горизонтальными линиями делить на группы.



Реализация диалогов, управляемых системой

Для реализации диалогов, управляемых системой, обычно используют **диалоговые окна**.

Если небольшое число элементов и последовательный сценарий, то проектируют одно диалоговое окно. Такое окно часто называют **формой**.

Если же диалог имеет сильно разветвленную структуру, в которой следующий вопрос зависит от уже полученных ответов, или число настраиваемых в процессе диалога элементов велико, то для каждого шага диалога проектируют свое диалоговое окно.



Проектирование форм

закljučается в выборе необходимых компонентов интерфейса и размещении их в пределах диалогового окна.

Если количество компонентов более 4-5, то целесообразно их визуальнo разделить, используя рамки.

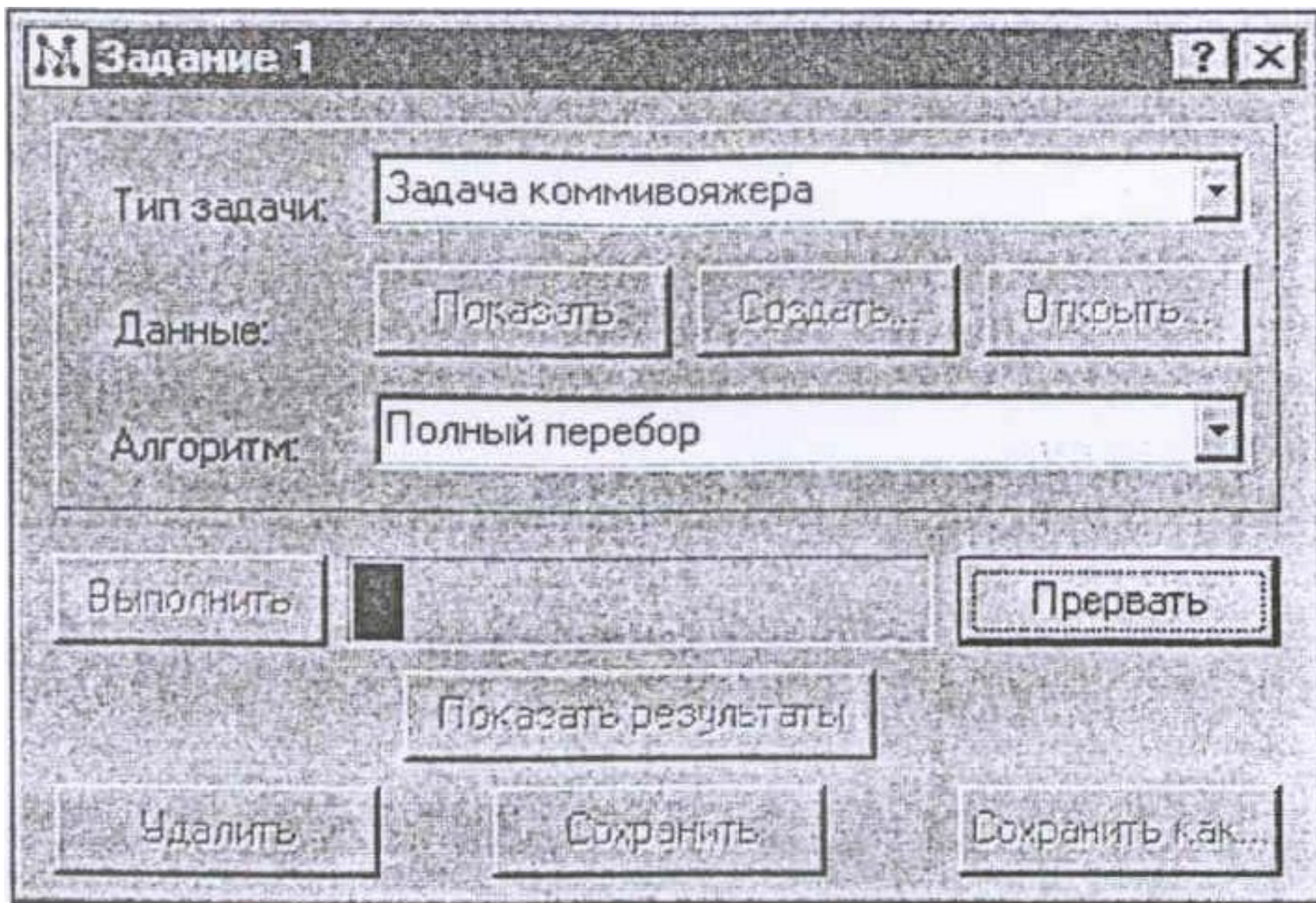
Проектирование последовательностей диалоговых окон.

В основе диалогов, управляемых системой, лежит жестко или нежестко заданный сценарий. Именно этот сценарий должен быть реализован последовательностью диалоговых окон.

Независимо от степени жесткости сценария при проектировании такой последовательности необходимо предусмотреть возможность возврата на предыдущий шаг.

Пример.

Реализовать диалог «Новое задание» системы решения комбинаторно-оптимизационных задач.



Задание 1

Тип задачи:

Данные:

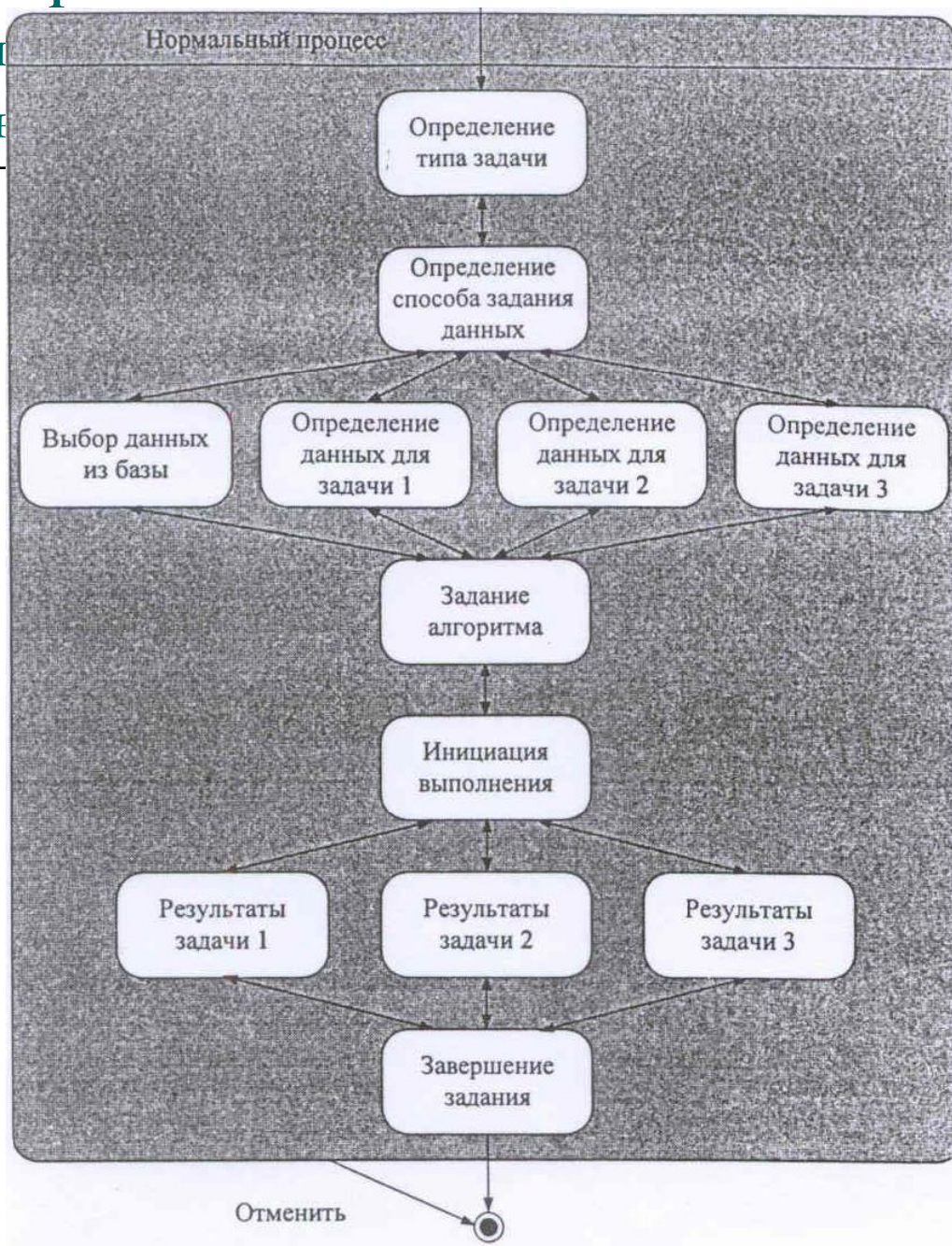
Алгоритм:



Пример №2.

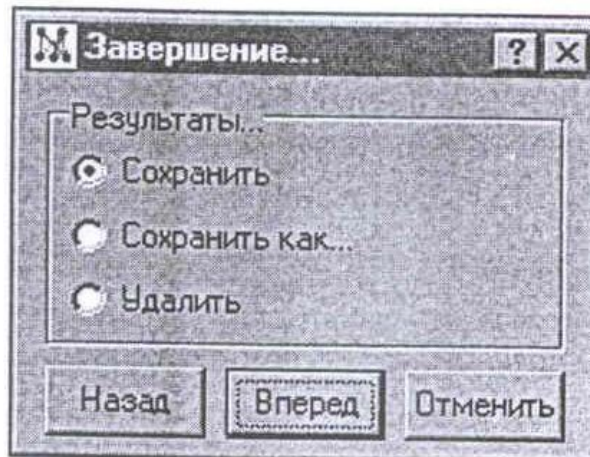
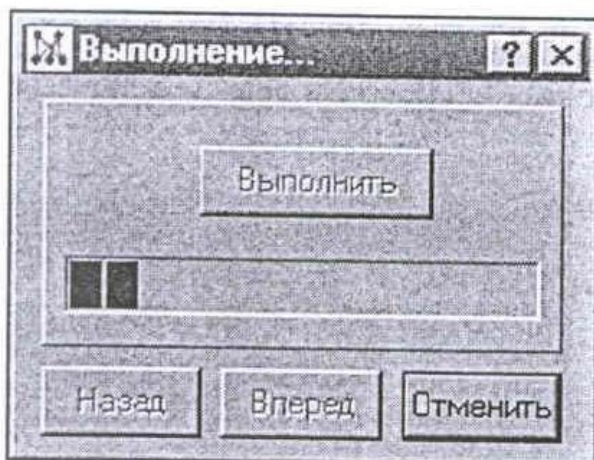
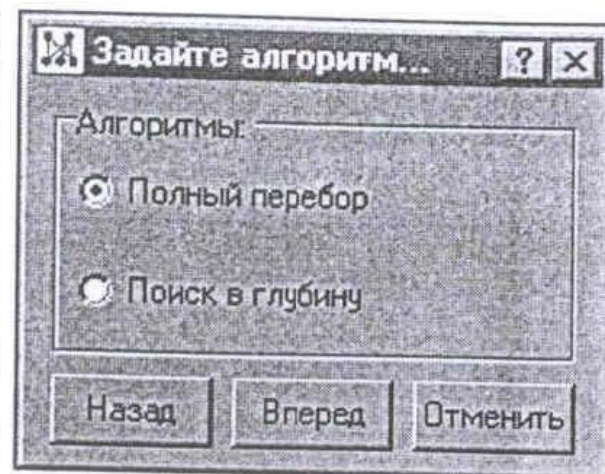
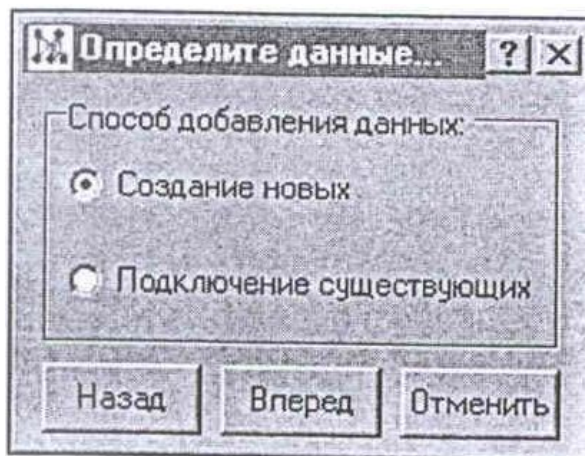
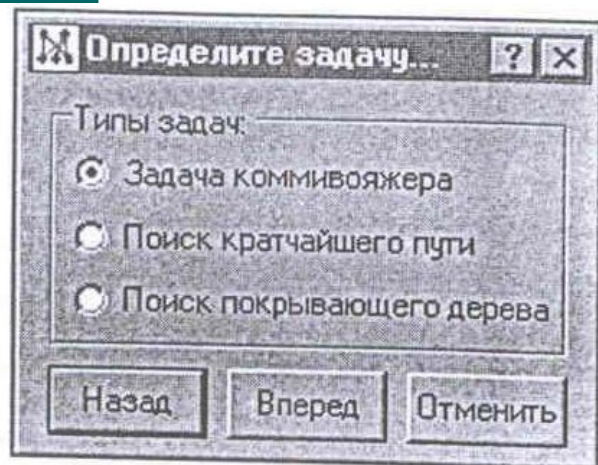
Реализация
решения

И
задач.



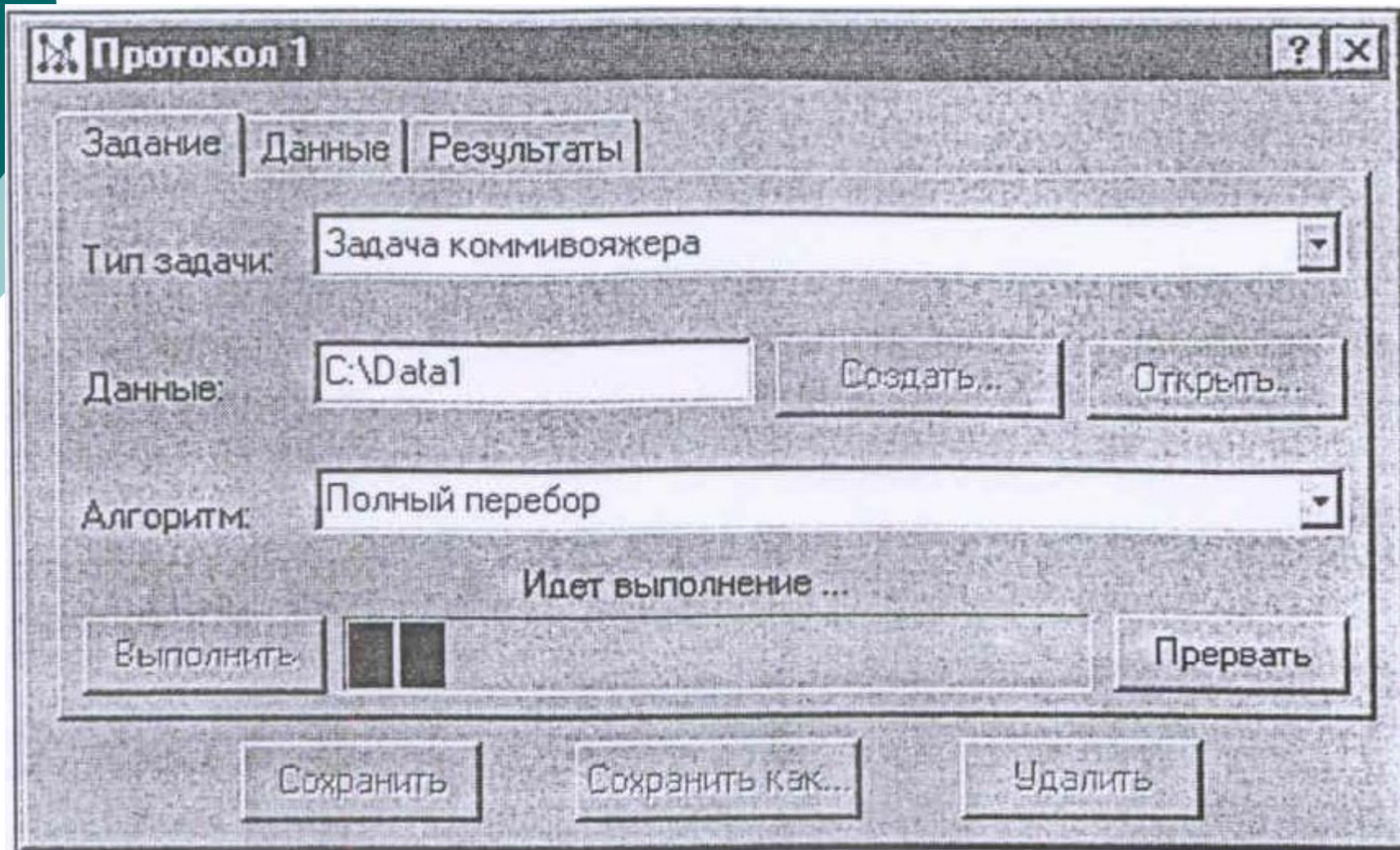
Пример №2.

Реализовать диалог «Новое задание» системы решения комбинаторно-оптимизационных задач.



Пример №3.

Реализовать диалог «Новое задание» системы решения комбинаторно-оптимизационных задач.



Протокол 1

Задание | Данные | Результаты

Тип задачи: Задача коммивояжера

Данные: C:\Data1 Создать... Открыть...

Алгоритм: Полный перебор

Идет выполнение ...

Выполнить Прервать

Сохранить Сохранить как... Удалить



Пользовательские интерфейсы прямого манипулирования и их проектирование

Метафора - мысленный перенос свойств или признаков одного объекта на другой, чем-то аналогичный первому.

Использование метафор в интерфейсах предполагает активизацию имеющегося у пользователя опыта.

Интерфейс прямого манипулирования должен обеспечивать пользователю среду, содержащую знакомые ему элементы.

Использовать метафоры надо очень аккуратно, так как при этом смысл придается всем элементам интерфейса.

Семантическое несоответствие между элементами интерфейса, тем, что от них ожидают, и тем, что они на самом деле выполняют, раздражает и дезориентирует пользователей.



Метафоры и анимация

При реализации метафор все большая роль уделяется средствам мультимедиа, в основном анимации.

Особенности анимационного подхода:

- Сохранение в сознании человека идентичность объектов
- Визуальный уровень программирования
- Временное программирование



Объекты интерфейса прямого манипулирования и их представления

Существует три основных типа объектов интерфейсов прямого манипулирования:

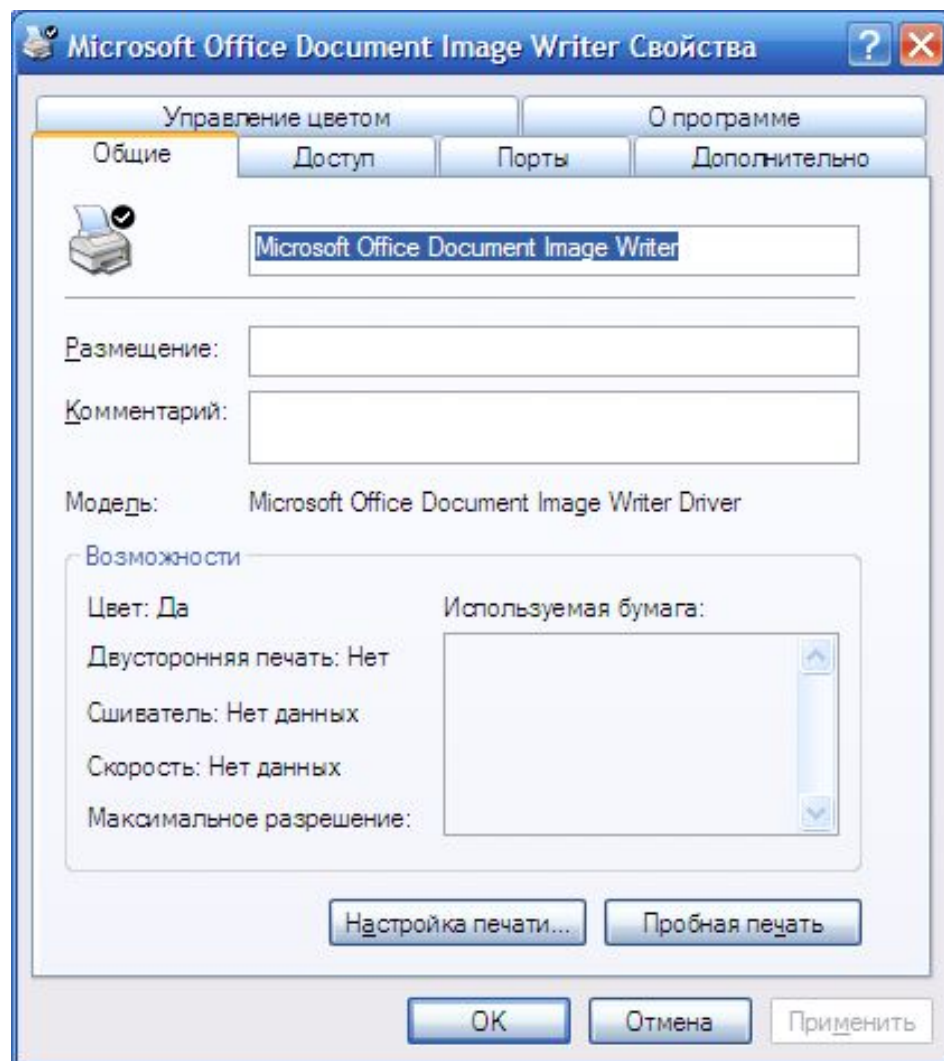
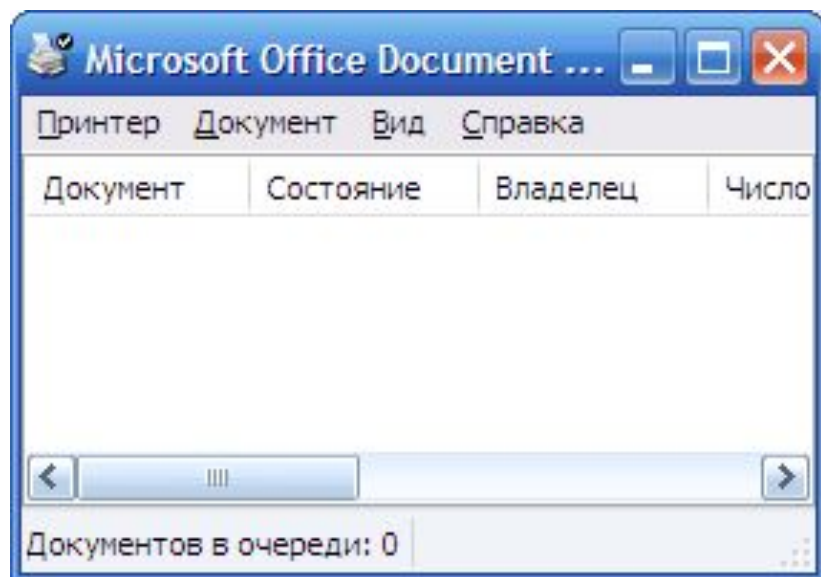
- объекты-данные;
- объекты-контейнеры;
- объекты-устройства.



Объекты интерфейса прямого манипулирования



Microsoft Office Document Image Writer



Технология Drag and Drop

Технология Drag and Drop («перетащил и бросил») определяет основные принципы прямого манипулирования:

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



Технология Drag and Drop

Эта технология также определяет основные принципы визуализации операции прямого манипулирования:

- исходное выделение;
- визуализация перемещения;
- целевое выделение;
- визуализация действия.



Проектирование интерфейсов прямого манипулирования

Проектирование интерфейсов прямого манипулирования выполняется на основе графов диалога и включает следующие процедуры:

- формирование множества объектов предметной области, которое должно быть представлено на экране;
- анализ объектов, определение их типов и представлений;
- уточнение взаимодействия объектов и построение матрицы прямого манипулирования;
- определение визуальных представлений объектов;
- разработка меню окон объектов и контекстных меню;
- создание прототипа интерфейса;
- тестирование на удобство использования.



Пример. Разработать пользовательский интерфейс прямого манипулирования для системы решения комбинаторно-оптимизационных задач.

1. Строим интерфейс, используя метафору «рабочий стол».
2. Для каждого объекта определяем тип и набор операций, связывающих эти объекты с остальными объектами предметной области.
3. К множеству объектов системы добавляем объекты-устройства;
4. Строим **матрицу взаимодействия объектов.**

Таблица: тип объекта и набор операций

Объект	Тип	Действия	Примечание
Протокол	Контейнер	Сохранить, Удалить, Распечатать	Результаты
Список протоколов	Контейнер	Очистить	-
Задание	Данные	Выполнить	Бланк задания
Данные	Данные	Связать с протоколом	-
Список данных	Контейнер	Очистить	-
Результаты	Данные	-	связаны с заданием

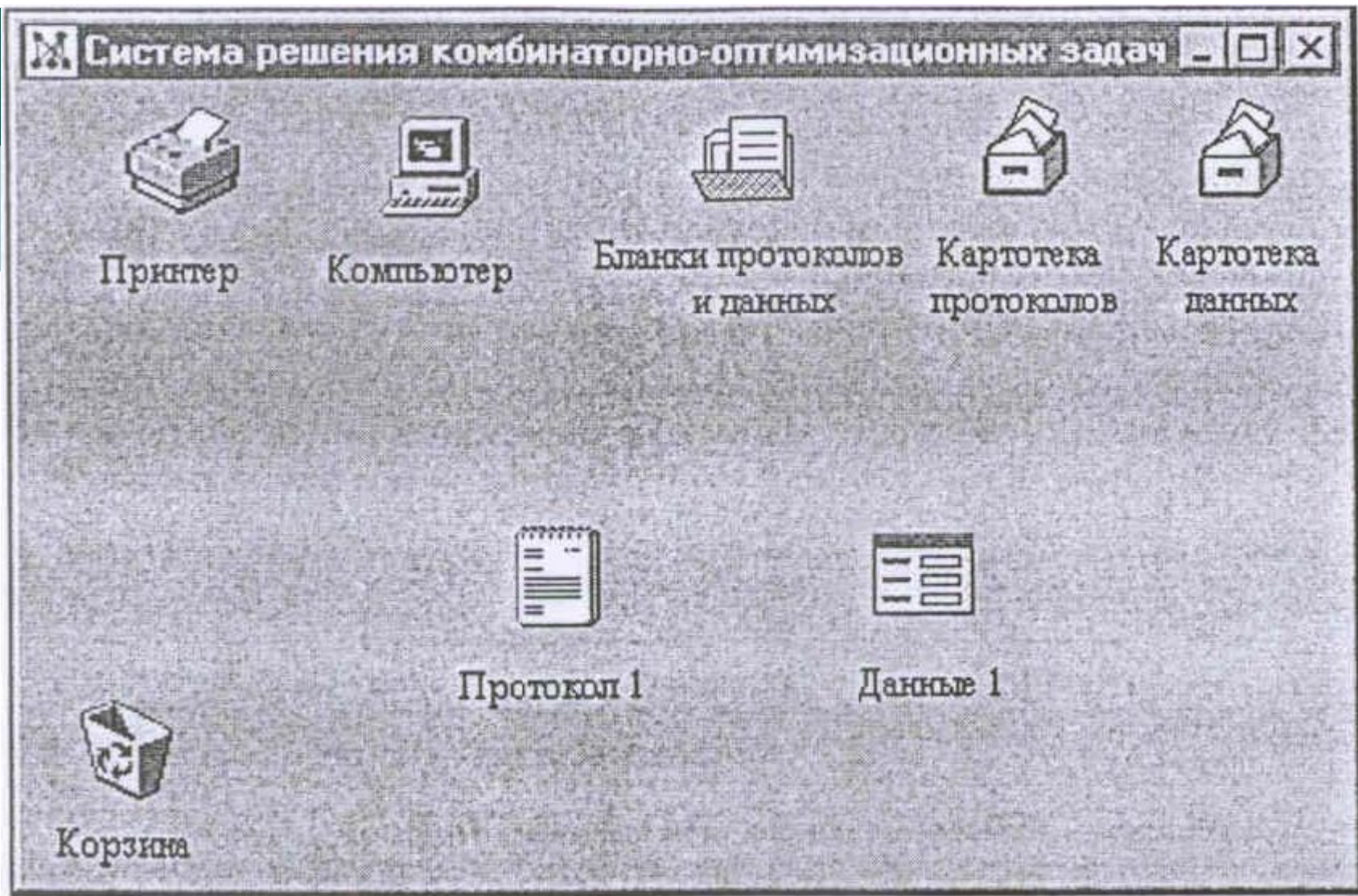
Таблица: тип объекта и набор операций

Объект	Тип	Действия	Примечание
Бланк протокола	Данные	-	Бланк протокола
Бланк данных	Данные	-	-
Компьютер	Устройство	-	-
Принтер	Устройство	-	-
Корзина	Устройство	-	-

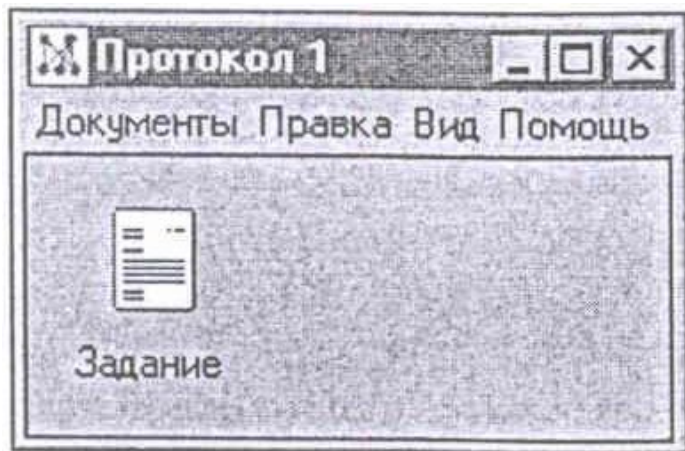
Таблица: матрица взаимодействия объектов

Исходные объекты	Операции, выполняемые при перемещении объектов						
	Исходные объекты						
	Протокол	Список протоколов	Список данных	Компьютер	Принтер	Корзина	Рабочий стол
Протокол	-	Добавить протокол в список	-	Выполнить задание	Распечатать протокол	Удалить протокол	-
Список протоколов	-	-	-	-	-	Удалить все протоколы	-
Задание	Занести задание в протокол	-	-	-	-	Удалить задания	-
Данные	Занести данные в протокол	-	Добавить данные в список	-	-	Удалить данные	-
Список данных	-	-	-	-	-	Удалить все данные	-
Бланк протокола	-	-	-	-	-	-	Создать протокол
Бланк данных	-	-	-	-	-	-	Создать данные

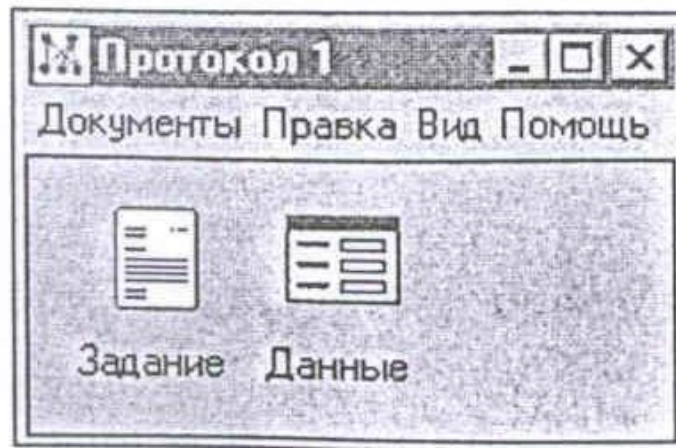
Внешний вид интерфейса



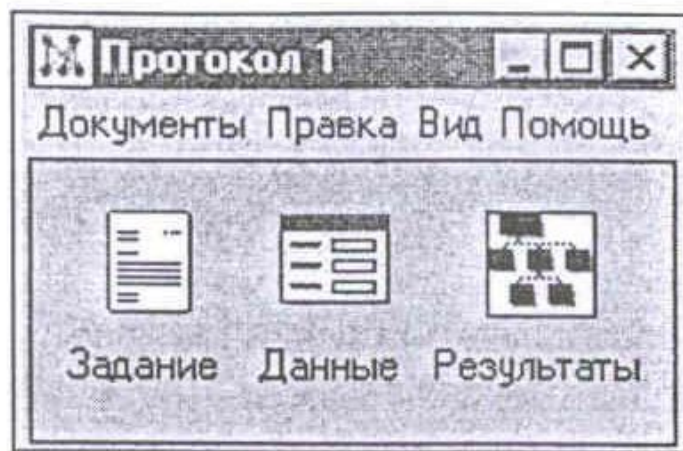
Внешний вид: объект Протокол




а



б



в



Интеллектуальные элементы пользовательских интерфейсов



Интеллектуальные элементы пользовательских интерфейсов

Новые перспективные элементы пользовательских интерфейсов, в основном привносящих в интерфейс элементы искусственного интеллекта:

Пример:

- Мастер,
- Советчик,
- Агент.



Интеллектуальные элементы пользовательских интерфейсов

Сделано множество попыток создания социализированного пользовательского интерфейса.

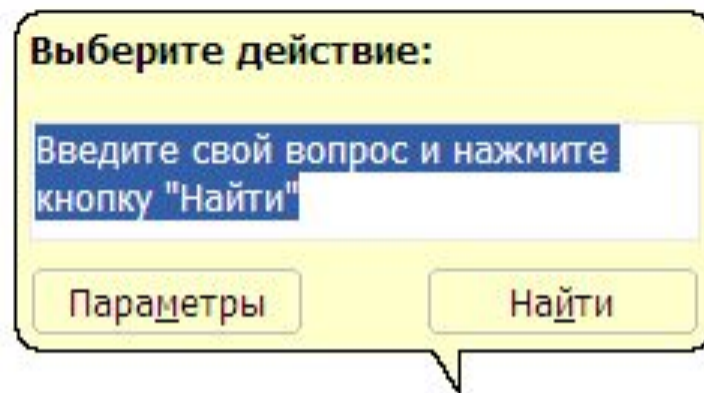
В основе такого интерфейса лежит **идея создания персонифицированного**, т. е. «имеющего личность», **интерфейса**.

Развлекательные 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, рассмотренный ниже, вызывает у многих пользователей резко отрицательную реакцию. Пока попытки создания такой «личности» успеха не имели.

Советчики представляют собой форму подсказки. Обычно их можно вызвать с помощью меню справки, командной строки окна или из всплывающего меню.

Советчики помогают пользователю решать конкретные задачи, но только в том случае, когда он представляет, что ему нужны



Выберите действие:

Введите свой вопрос и нажмите кнопку "Найти"

Параметры Найти





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.

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.

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).



Содержание

- Типы пользовательских интерфейсов и этапы их разработки
- Психофизические особенности человека, связанные с восприятием, запоминанием и обработкой информации
- Пользовательская и программная модели интерфейса
- Классификации диалогов и общие принципы их разработки
- Основные компоненты графических пользовательских интерфейсов
- Реализация диалогов в графическом пользовательском интерфейсе
- Пользовательские интерфейсы прямого манипулирования и их проектирование
- Интеллектуальные элементы пользовательских интерфейсов