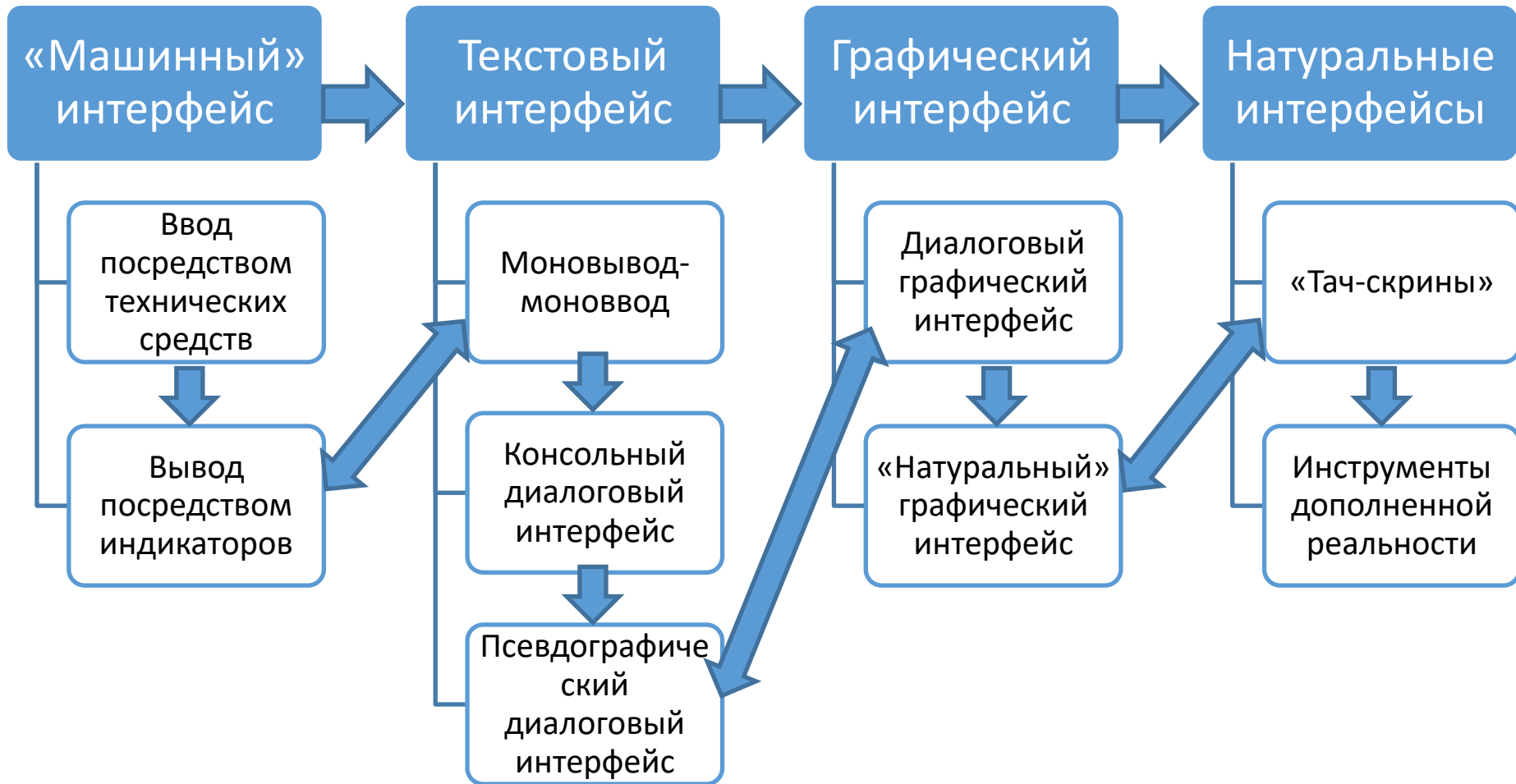


Правила разработки интерфейсов пользователя

Понятие пользовательского интерфейса

- **Пользовательский интерфейс** представляет собой средство общения между пользователем и вычислительной системой.
- **Пользовательский интерфейс или человеко-машинный диалог** – это набор приемов взаимодействия с компьютером
- **Пользовательский интерфейс** – совокупность средств и элементов, с помощью которых пользователь взаимодействует с информационной системой.
 - Средства и элементы вывода;
 - Средства и элементы ввода;
 - Комбинированные средства и элементы.

Этапы эволюции интерфейсов



Факторы, влияющие на удобство работы с ПО

<i>Факторы</i>	<i>Вызываются</i>	<i>Влияют на</i>
Социальные факторы	Психологическим климатом	Эмоциональный комфорт
Физическая эргономика	Аппаратным обеспечением	Физический комфорт
Психологическая эргономика	Качеством разработки ПО	Умственный комфорт

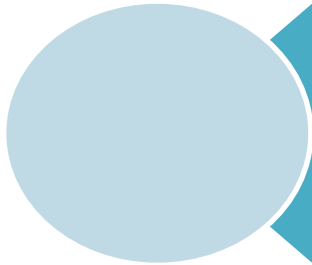
Правила разработки пользовательского интерфейса

Проектирование пользовательского интерфейса (ПИ)

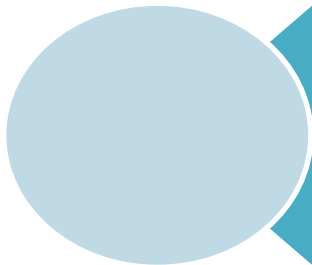
Требования, предъявляемые к ПИ, для обеспечения **максимального удобства пользователю** при работе с программой:

- дизайн интерфейса отвечает правилам эргономики;
- присутствует естественность (интуитивность) работы с программой;
- нагрузка на память пользователя, по возможности, минимальна;
- стандартность приемов работы (согласованность с прошлым навыком);
- подсказки, позволяющие пользователю принять решение в создавшейся ситуации;
- интерактивная помощь (возможность ее вызова из любого места программы);
- очевидность меню (простая формулировка, иерархическая структура, логическое соответствие пунктов и подпунктов);
- действия пользователя должны быть обратимыми (т. е. должна предоставляться возможность отмены);
- возможность использования «горячих» клавиш;
- экстренный выход из программы.

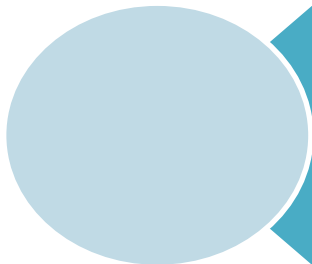
Три правила разработки пользовательского интерфейса



***Контроль пользователем
интерфейса***



***Уменьшение загрузки памяти
пользователя***



Создание совместимого интерфейса

Правила и этапы разработки ПИ

Правила разработки пользовательского интерфейса:

Правило 1: дать контроль пользователю

- Возможность использования мыши и клавиатуры.
- Обеспечение обратимых действий и обратной связи, поясняющих сообщений и текста.
- Учет разного уровня навыков пользователей.
- Прозрачность пользовательского интерфейса.
- Возможность настройки интерфейса на свой вкус.

Правило 2: уменьшить нагрузку на память пользователя

- Не нагружать кратковременную память.
- Обеспечение распознавания, а не повторения.
- Наличие функции отмены действия, его повтора, установки по умолчанию, подсказок.

Правило 3: сделать интерфейс совместимым

- Общая совместимость всех программ.
- Сохранение результатов взаимодействия.
- Эстетическая привлекательность и цельность.

Принципы, дающие пользователю контроль над системой

Безрежимность	<ul style="list-style-type: none">• Используйте режимы благоразумно
Гибкость	<ul style="list-style-type: none">• Предоставьте пользователю возможность выбора: мышь, клавиатура или их сочетание
Прерываемость	<ul style="list-style-type: none">• Позвольте пользователю сфокусировать внимание
Полезность	<ul style="list-style-type: none">• Демонстрируйте сообщения, которые помогут ему в работе
Снисходительность	<ul style="list-style-type: none">• Создайте условия для немедленных и обратимых действий, а также обратной связи
Способность ориентировки	<ul style="list-style-type: none">• Обеспечьте соответствующие пути и выходы
Доступность	<ul style="list-style-type: none">• Приспосабливайте систему к пользователям с различным уровнем подготовки
Облегченность в пользовании	<ul style="list-style-type: none">• Сделайте интерфейс более понятным
Приспосабливаемость	<ul style="list-style-type: none">• Дайте пользователю возможность настраивать интерфейс
Интерактивность	<ul style="list-style-type: none">• Разрешите пользователю напрямую манипулировать объектами интерфейса

Принципы, позволяющие снизить загрузку памяти пользователя

Запоминание	<ul style="list-style-type: none">• Не загружайте кратковременную память
Распознавание	<ul style="list-style-type: none">• Полагайтесь на распознавание, а не на повторение
Информирование	<ul style="list-style-type: none">• Представьте визуальные заставки
Быстрота	<ul style="list-style-type: none">• Предусмотрите «быстрые» пути
Интуитивность	<ul style="list-style-type: none">• Активизируйте синтаксис действий с объектами
Перенос	<ul style="list-style-type: none">• Используйте метафоры из реального мира
Контекст	<ul style="list-style-type: none">• Применяйте раскрытие и объяснение понятий и действий
Организация	<ul style="list-style-type: none">• Увеличьте визуальную яркость
	<ul style="list-style-type: none">• Дайте пользователю возможность настраивать интерфейс
	<ul style="list-style-type: none">• Предусмотрите установки по умолчанию, команды Undo и Redo

Пользовательский интерфейс

С точки зрения программного обеспечения в состав интерфейса входят два компонента:

- *набор процессов ввода-вывода;*
- *процесс диалога.*

Процесс диалога — это механизм обмена информацией, который можно рассматривать как оболочку, включающую все входящие в систему процессы по выполнению определенных заданий.

Задачи диалогового процесса:

- определение задания, которое пользователь возлагает на систему;
- прием логически связанных входных данных от пользователя и размещение их в переменных соответствующего процесса в нужном формате;
- вызов процесса выполнения требуемого задания;
- вывод результатов обработки по окончании процесса.

Пользовательский интерфейс

Сообщения – порции информации, участвующие в диалоговом обмене.

Различают:

- входные сообщения, которые генерируются человеком с помощью средств ввода;
- выходные сообщения, которые генерируются компьютером в виде текста, звуковых сигналов и/или изображений и выдаются на устройства вывода.

Типы сообщений при диалоге пользователя и системы



Типы диалога

- *Диалог, управляемый системой*, — это диалог, в котором процесс жестко задает, какое задание можно выбрать и какие данные вводить.
- *Диалог, управляемый пользователем*, — это диалог, в котором инициатива принадлежит пользователю.

Типы диалога:

- вопрос и ответ;
- меню;
- экранных форм;
- на базе команд.

Критерии хорошего диалога

- Естественность.
- Последовательность.
- Краткость.
- Поддержка пользователя.
- Гибкость.



Естественный диалог

- это, такой, который не вынуждает пользователя, взаимодействующего с системой существенно изменять свои традиционные способы решения задачи

Последовательный диалог

- гарантирует, что пользователь, освоивший работу одной части системы, не запутается, разбираясь с особенностями описания и работы другой части системы.

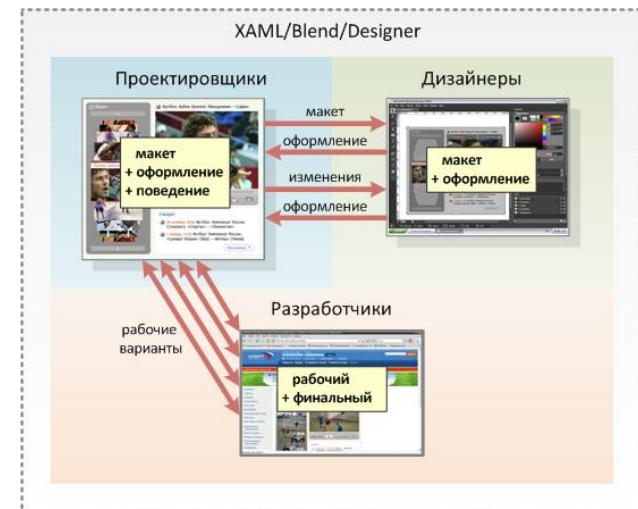
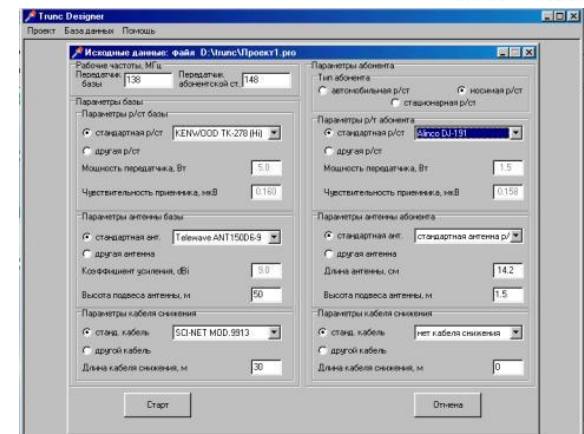


Рисунок 2. Пространство XAML

Краткий диалог

- требует от пользователя ввода только минимума информации, необходимой для работы системы.



Поддержка пользователя

- это мера помощи, которую диалог оказывает пользователю при его работе с системой.
- Основные аспекты
- Количество и качество инструкций
- Характер сообщений об ошибках
- Подтверждение каких-либо действий системы

Гибкость

- это мера того, насколько хорошо диалог соответствует различным уровням подготовки и производительности труда пользователя.

Виды справок

- **Базовая справка** объясняет пользователю сущность и назначение системы.
- **Обзорная справка** рекламирует пользователю функции системы.
- **Справка предметной области** отвечает на вопрос «Как сделать хорошо?».
- **Процедурная справка** отвечает на вопрос «Как это сделать?».
- **Контекстная справка** отвечает на вопросы «Что это делает?» и «Зачем это нужно?».
- **Справка состояния** отвечает на вопрос «Что происходит в настоящий момент?».



Правила проектирования интерфейсов

- Якоб Нильсен (Jakob Nielsen),
- Рольф Молич (Rolf Molich)

В 1990 г разработали набор эвристических правил, которые, наиболее часто используются в области дизайна интерфейса

Эвристические правила Якоба

- Видимость состояния системы (правило обратной связи)
- Информированность пользователя
- Средства обеспечения обратной связи
- Время оповещения
- Равенство между системой и реальным миром
- Свобода действий пользователя
- Последовательность и стандарты
- Предупреждение ошибок
- Понимание лучше, чем запоминание
- Гибкость и эффективность использования
- Эстетичный и минималистический дизайн
- Распознавание и исправление ошибок
- Описание ошибки
- Описание решения проблемы
- Справка и документация

Эвристические правила Якоба

- Возможность видеть состояние системы - система должна всегда информировать пользователей о том, что происходит, через соответствующую обратную связь в разумные сроки.
- Соответствие системы реальному миру - система должна «разговаривать» с пользователем на одном языке, используя слова, фразы и понятия, которые знакомы пользователям, а не системно-ориентированные термины. Следуйте условиям реального мира, предоставляя информацию в естественном и логичном порядке.
- Управление пользователями и их свобода - пользователи часто выбирают системные функции по ошибке, и необходим четко обозначенный "запасной выход" для того, чтобы выйти из ненужного состояния, не прибегая к помощи расширенного диалога. Обеспечьте возможность перехода на «шаг назад» и на «шаг вперед».

Эвристические правила Якоба

- Последовательность и стандарты - пользователи не должны быть в замешательстве по поводу использования тех или иных функций (не должен возникать вопрос, значат ли слова или действия одно и то же). Следуйте общепринятым стандартам.
- Предотвращение ошибок - тщательная разработка, которая в первую очередь предотвращает возникновение проблем, даже лучше хороших сообщений об ошибках. Либо устраните элементы, предрасположенные к ошибкам, либо произведите их проверку и введите для пользователей опцию подтверждения перед совершением действия.
- Не нагружайте память пользователя - не заставляйте пользователя хранить много информации в памяти. Делайте так, чтобы информация была по возможности на виду, и пользователю не приходилось запоминать информацию из одной части диалога для работы с другой. Инструкция по использованию системы должна быть видна и легкодоступна всякий раз, когда она может понадобиться.

Эвристические правила Якоба

- Гибкость и эффективность использования - кнопки быстрого перехода, которые незаметны для начинающих пользователей, зачастую могут ускорить взаимодействие опытных пользователей системы. Таким образом, удовлетворяются требования как опытных, так и неопытных пользователей. Дайте возможность пользователям приспособиться к выполнению часто повторяемых действий.
- Эстетичный и минималистский дизайн - диалоги не должны содержать информацию, которая несущественна или редко требуется. Каждый дополнительный блок информации в диалоге отвлекает внимание от основной, значимой информации и делает её менее заметной.
- Помощь пользователям в распознавании, диагностике и восстановлении после ошибок - сообщения об ошибках должны быть написаны простым языком, точно указывать проблему и предлагать конструктивное решение.

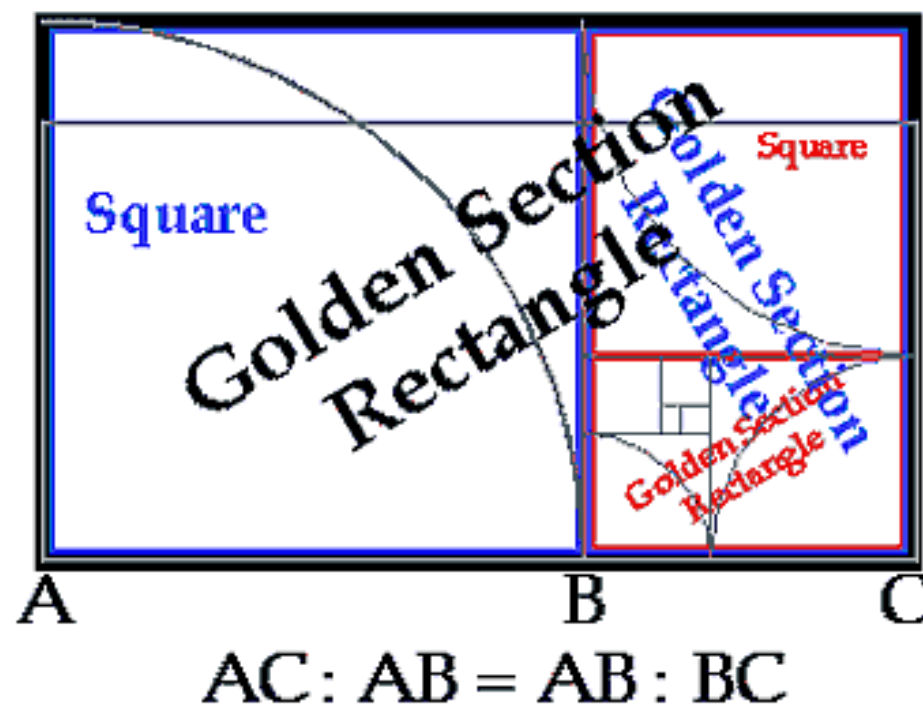
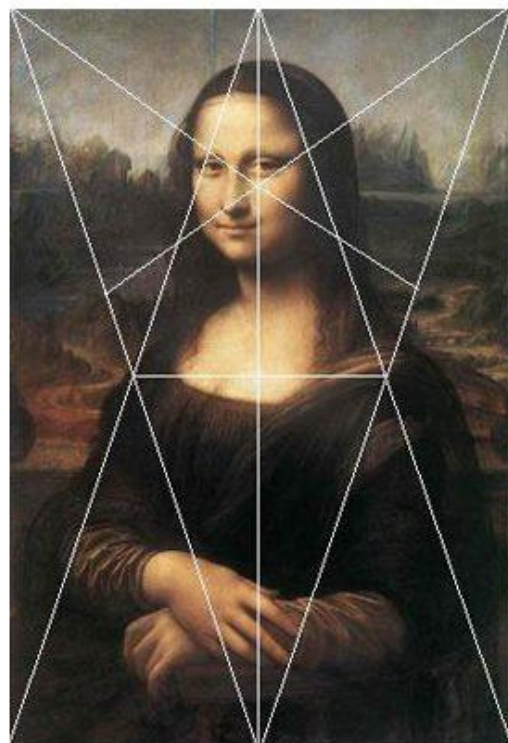
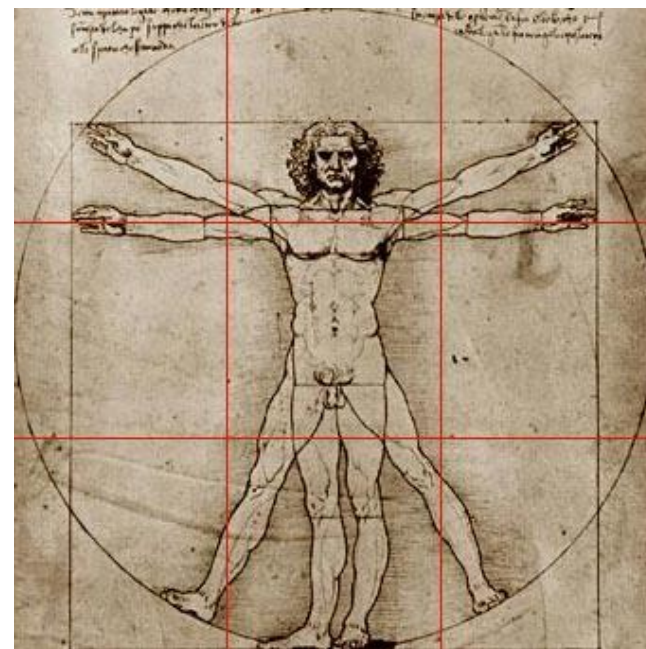
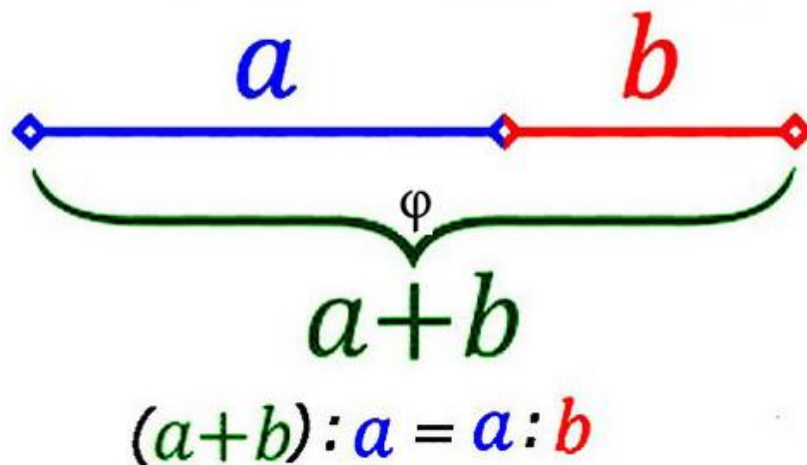
Эвристические правила Якоба

- Помощь и техническая документация - хорошо, если систему можно использовать, не прибегая к технической документации, но даже в этом случае может возникнуть необходимость получить какую-либо помощь. Должна быть возможность легко найти информацию такого рода в соответствии с проблемой пользователя и получить список конкретных шагов для решения задачи; при этом информации должно быть не слишком много.

Принципы построения интерфейсов

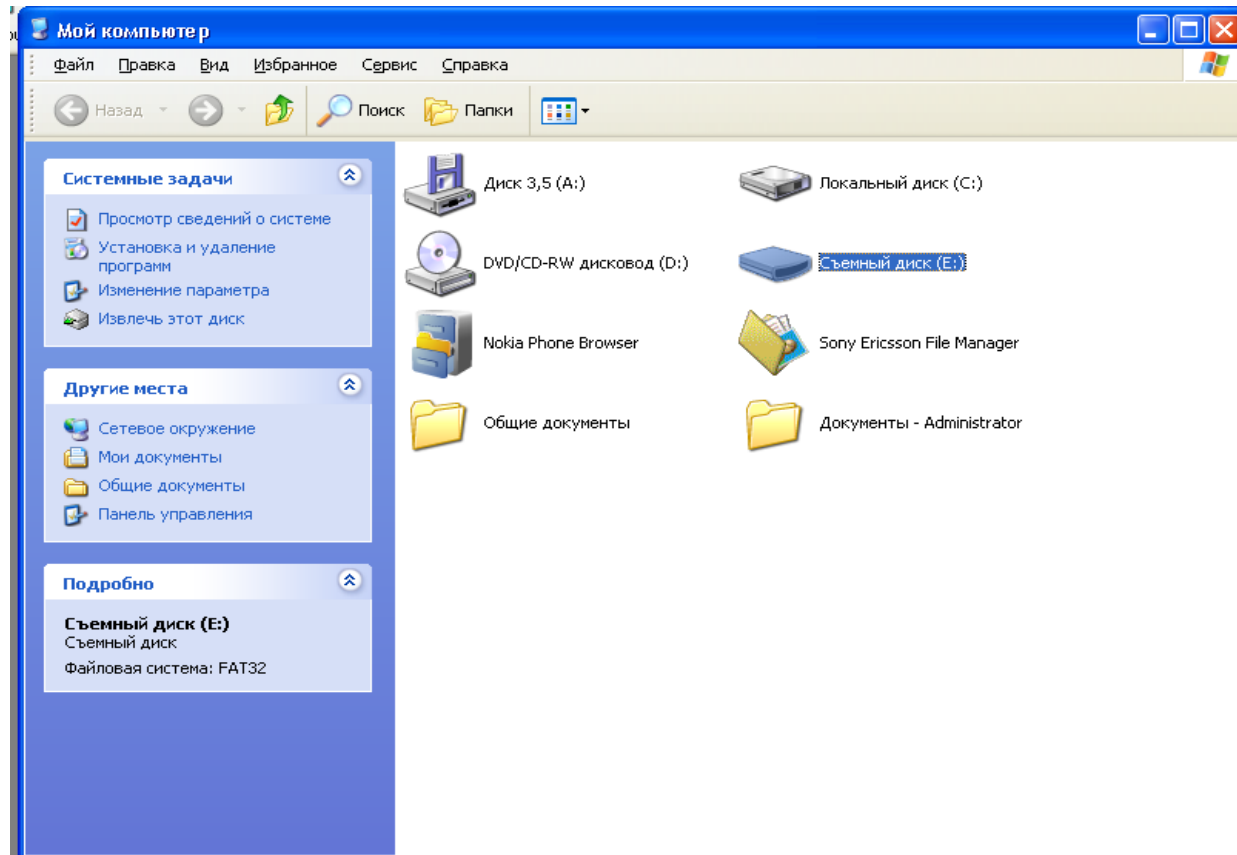
- *Золотое сечение*
- *Кошелек Миллера*
- *Принцип группировки*
- *Бритва Оккама или KISS*
- *Видимость отражает полезность*
- *Умное заимствование*

ЗОЛОТОЕ СЕЧЕНИЕ



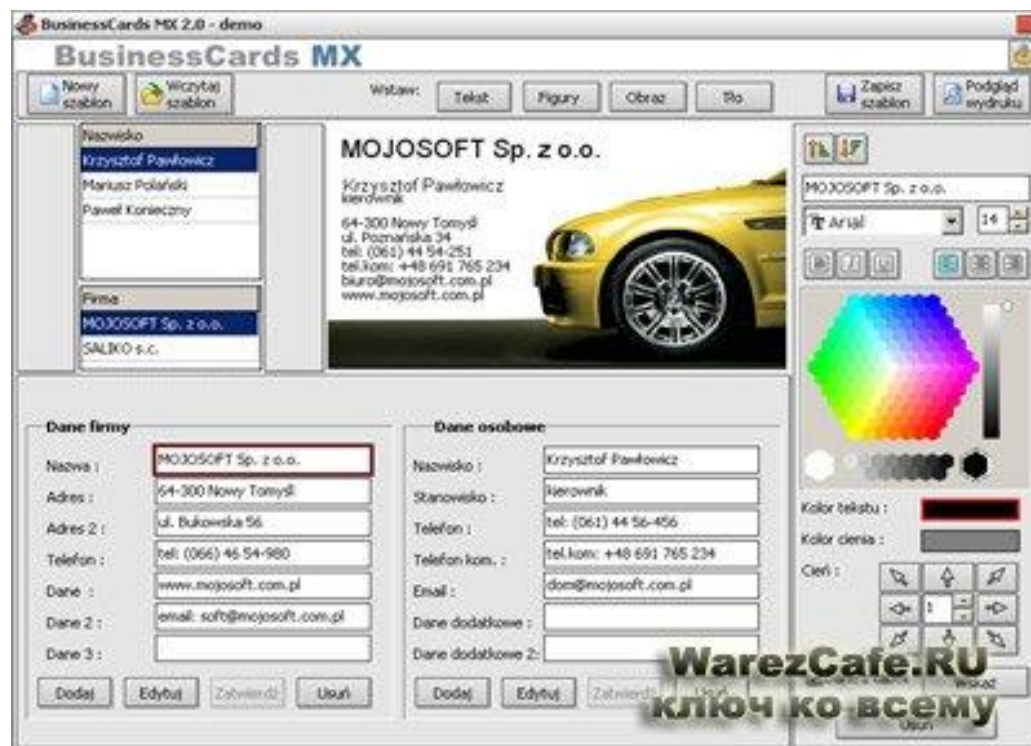
Золотое сечение

- **Золотое сечение** — это самая комфортная для глаза пропорция, форма, в основе построения которой лежит сочетание симметрии и золотого сечения, способствует наилучшему зрительному восприятию и появлению ощущения красоты и гармонии.



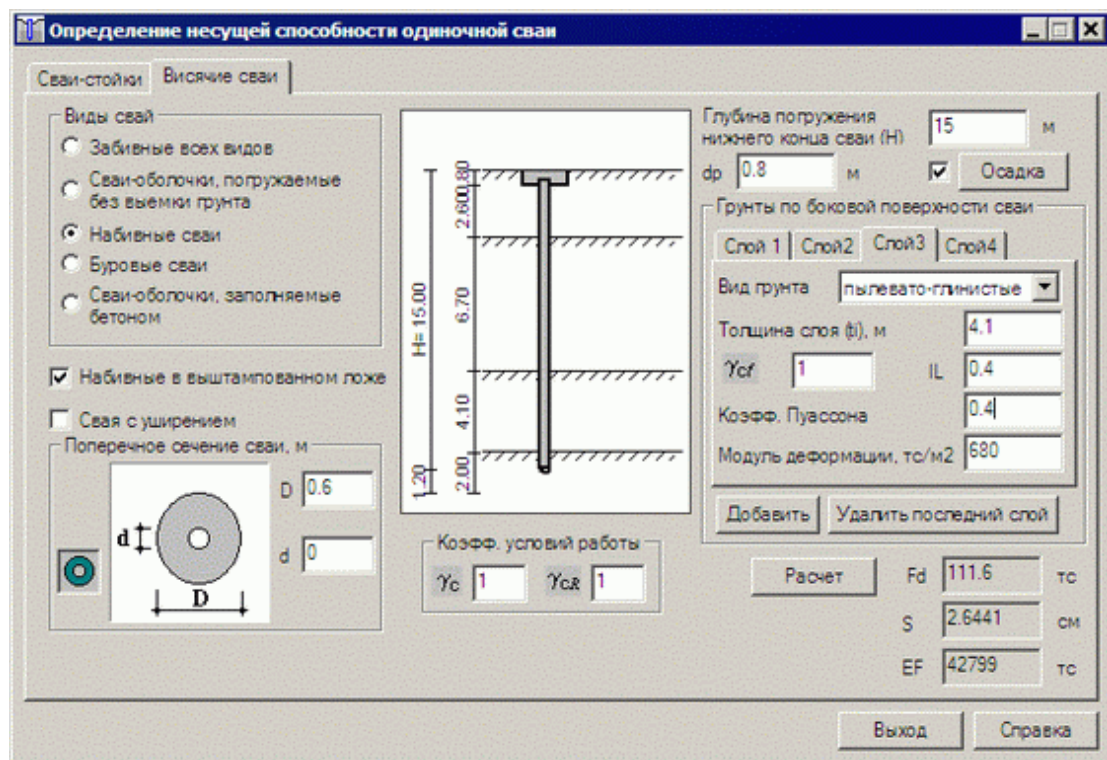
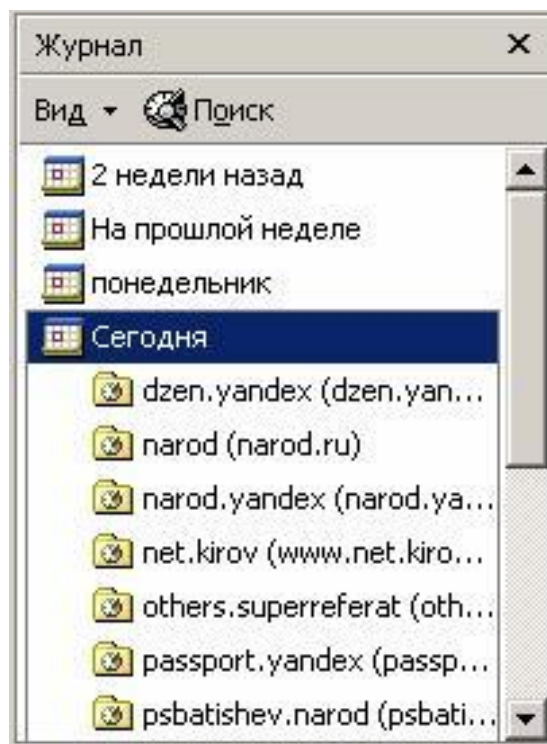
Кошелек Миллера

- Следует группировать элементы в программе (кнопки на панелях инструментов, пункты меню, закладки, опции на этих закладках и т. п.)
- В группе не более семи элементов, в крайнем случае — девяти.



Принцип группировки

- Экран программы должен быть разбит на ясно очерченные блоки элементов, может быть, даже с заголовком для каждого блока.
- Группировка должна быть осмысленной: как расположение элементов в группах, так и расположение самих групп друг от друга должны быть продуманы.

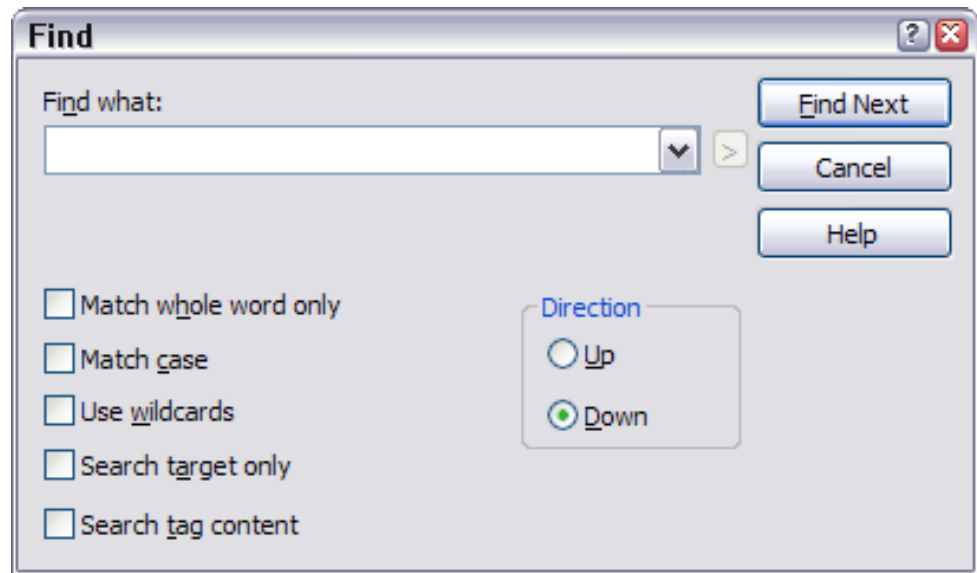


"Бритва Оккама"

Философский принцип гласит: "Не множить сущности без надобности". Или, как говорят американцы, KISS ("Keep It Simple, Stupid" — "Не усложняй, болван").

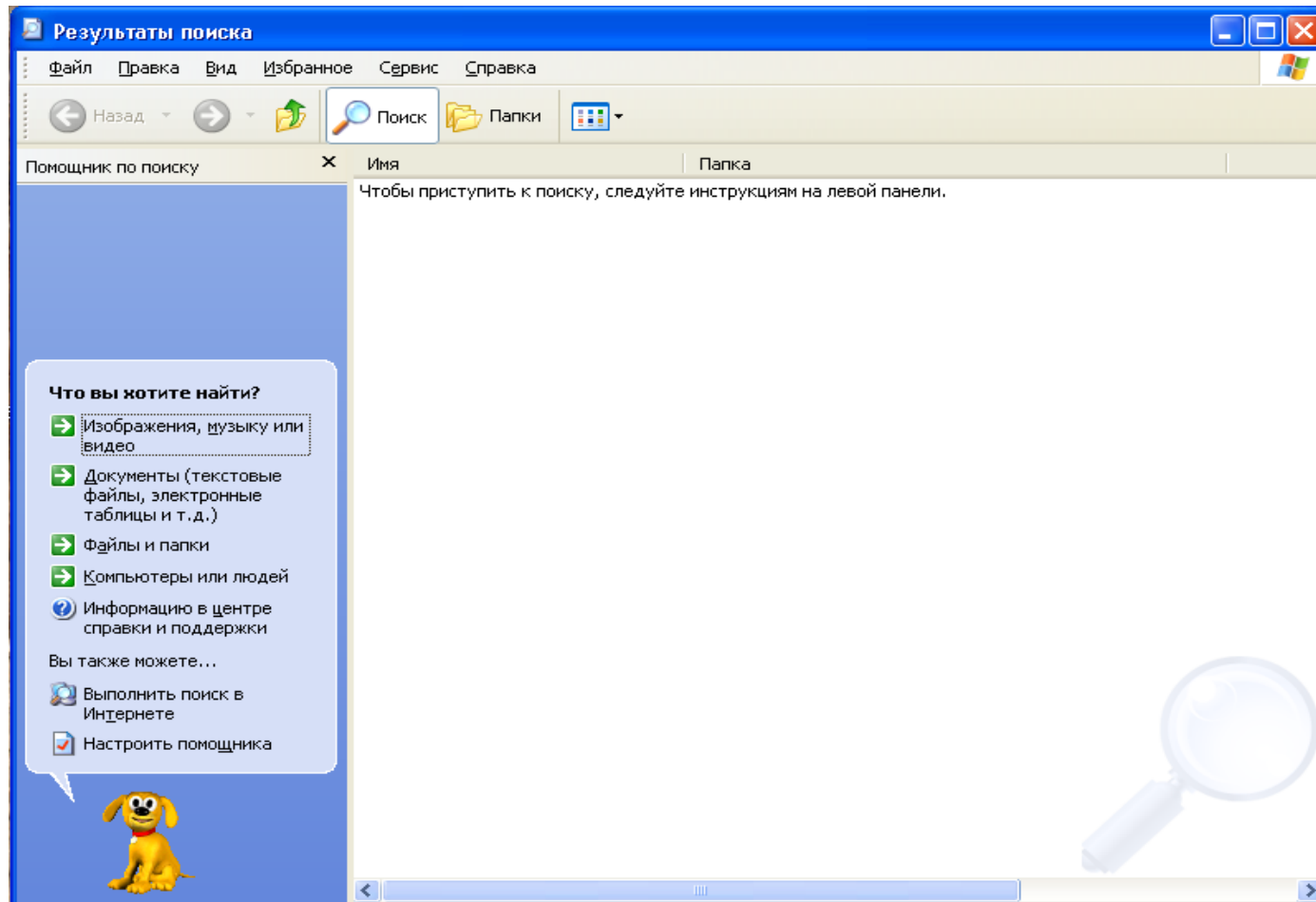
На языке интерфейсов это означает, что:

- любая задача должна решаться минимальным числом действий;
- логика этих действий должна быть очевидной для пользователя;
- движения курсора и даже глаз пользователя должны быть оптимизированы.



Видимость отражает полезность

- Вынести самую важную информацию и элементы управления на первый план и сделать их легкодоступными пользователю, а менее важную — переместить, например, в меню

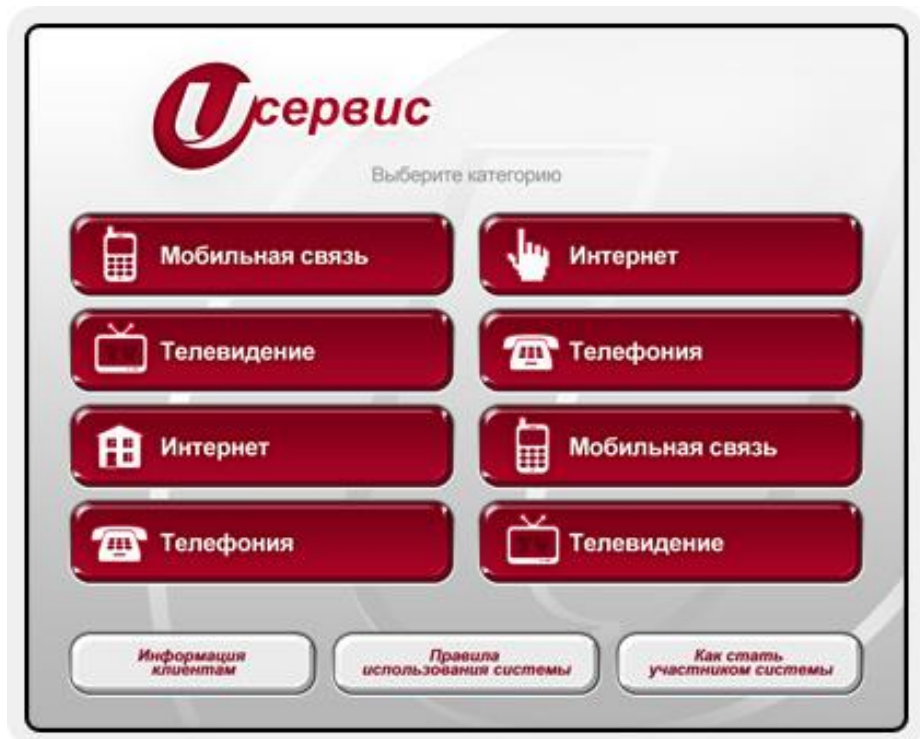
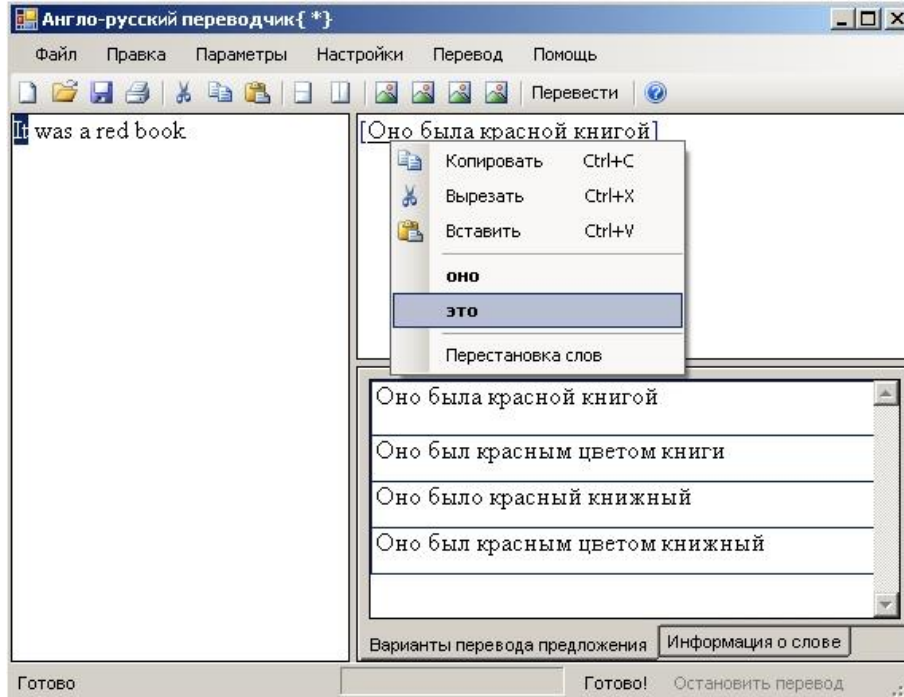


Умное заимствование

- Заимствование широко распространенных приемов дизайна интерфейсов и удачных находок авторов конкурирующих программ позволяет резко сократить время обучения и повысить комфорт пользователя. При работе он будет использовать уже приобретенные навыки — этот вопрос затрагивает и принцип равенства между системой и реальным миром.



Меню



Меню

- Следуйте стандартным соглашениям о расположении пунктов меню принятым в Windows File, Edit, View, и т.д.
- Группируйте пункты меню в логическом порядке и по содержанию.
- Для группировки пунктов в раскрывающихся меню используйте разделительные линии
- Избегайте избыточных меню.
- Избегайте пунктов меню верхнего уровня, не содержащих раскрывающихся меню
- Не забывайте использовать символ троеточия для обозначения пунктов меню, активизирующих диалоговые окна.
- Обязательно используйте клавиатурные эквиваленты команд и "горячие" клавиши.
- Помещайте на панель инструментов часто используемые команды меню.

Форма

Пенсионный платеж юридических лиц

Платежное поручение N на сумму

Дата валютирования Дата документа ЕКНП

Отправитель

Счет Наименование
 РНН
 Банк Г.АЛМАТЫ ФИЛИАЛ ОАО "БАНК ТУРАНАЛЕМ"
 Сектор экономики
 Руководитель
 Гл. бухгалтер
☐ Признак нерезидентства

Получатель (пенсионный фонд)

Банк Г.АЛМАТЫ ФИЛИАЛ ОАО "БАНК ТУРАНАЛЕМ"
 Счет
 Наименование
 РНН
 Сектор экономики
☐ Признак нерезидентства

Посредник

Банк Счет

Назначение платежа

ТЕСТ

Список вкладчиков пенсионных взносов

Сохранить Отмена

Бронирование – шаг 1: заполнение параметров бронирования

Источник Город
 Туроператор Категория
 Отправление Отель
 Страна Питание
 Дата заезда по Прод.
 Взрослых детей
 Покупатель

☒ все

Тип	Номер	Питание	Стоим.	Партнер	Услуги	Прод.
<input checked="" type="checkbox"/> Special tour1	DBL Sea View Adult	BB	1005	Партнер1	7	8
<input checked="" type="checkbox"/> Special tour2	DBL STD 2+1 Mountain View	HB	1155	Партнер 2	6	8

Фамилия	Имя	Дата рожд.	Серия	Номер пасп.	Окончание	Номер в отеле
1. Mr	Sidorov	Andrey	15.09.06	3333	55556677	15.09.06
2. Mr	Sidorova	Marina	15.09.06	3333	55556677	15.09.06
3. Mr						
4.						
5.						
6.						

Перелет

Рейсы	Вылет	Дата	Прилет	Вылет	Дата	Прилет	Тариф
SU246 – SU247	Москва	MOW	26.10.2006	13:10	EGP	EGP	05.11.2006 14:00
SU246 – SU247	Москва	MOW	27.10.2006	13:20	EGP	EGP	06.11.2006 14:15

Дополнительная информация

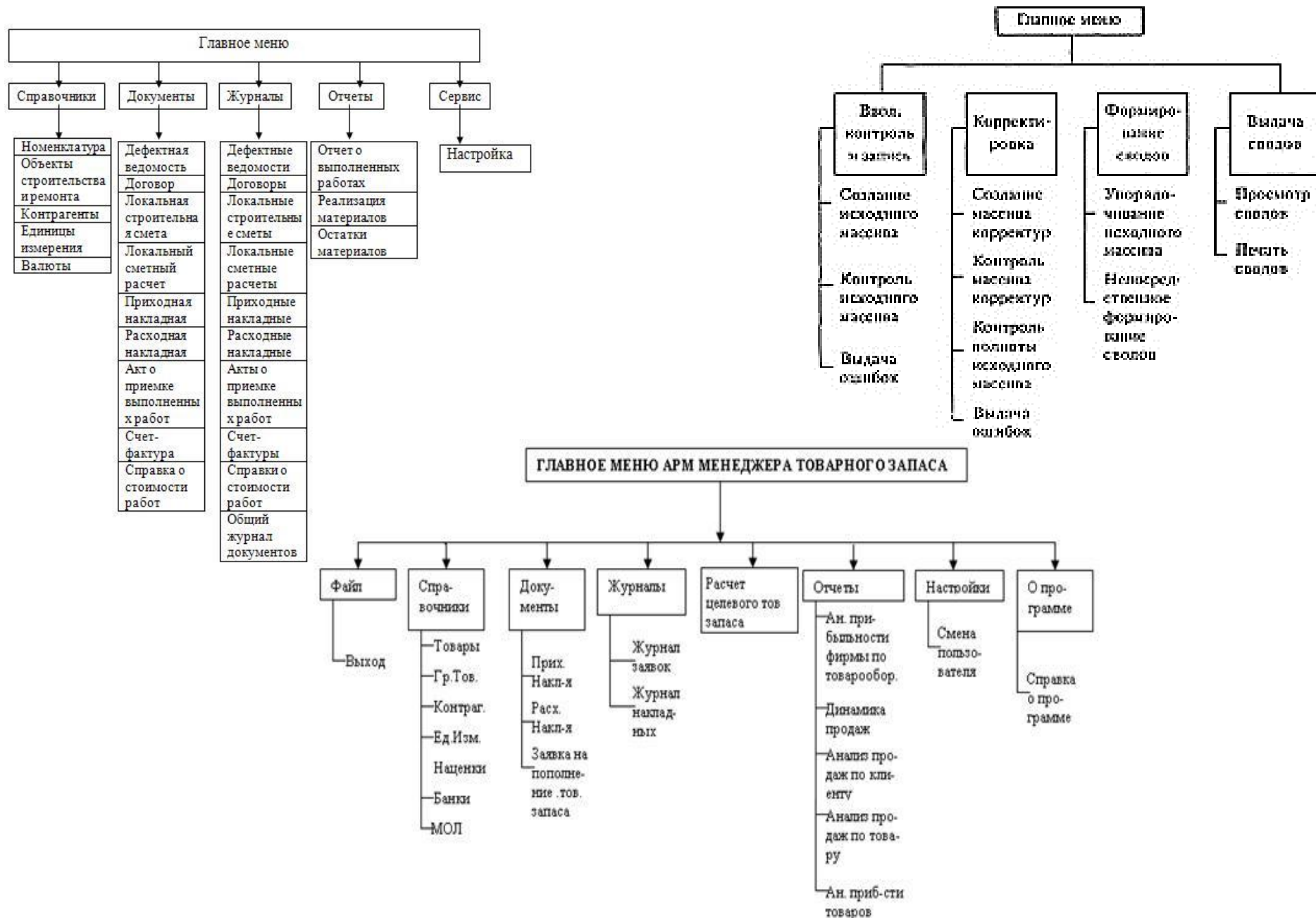
Менеджер

Итого: 790 Eu
 Комиссия: 74 Eu
 К оплате: 716 Eu

Приемы ускорения ввода данных

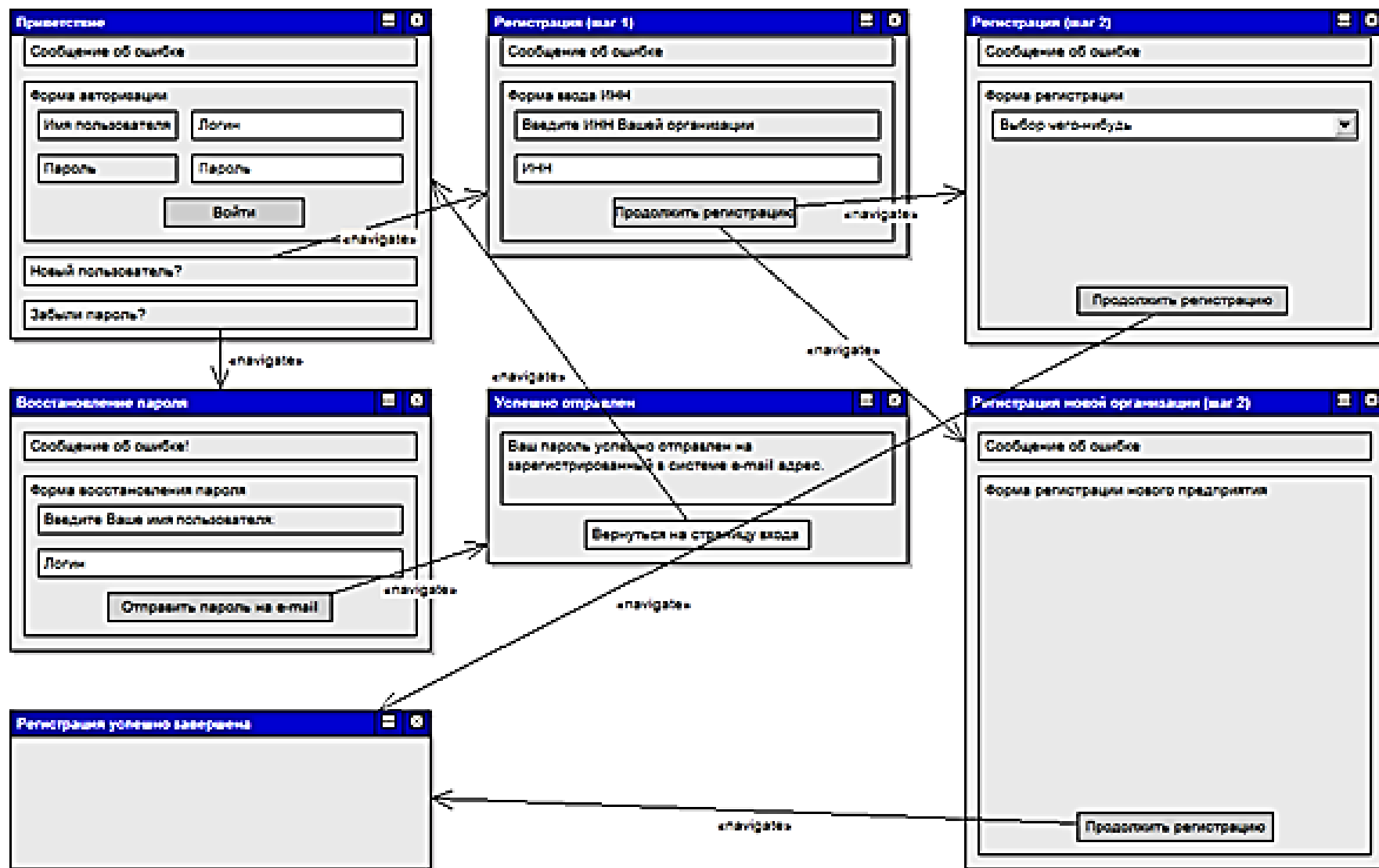
- Всегда назначайте клавиатурные эквиваленты команд;
- Расположение элементов должно быть согласовано с задачами пользователя.
- Не заставляйте пользователя выполнять лишнюю работу.
- Не ставьте работу формы в зависимость от содержимого отдельных полей.
- Используйте заметную, но ненавязчивую обратную связь с пользователем.
- Если возможно, выполняйте добавление и редактирование записей в одной и той же форме.

Дерево диалога



Навигация по окнам

system Авторизация и аутентификация



Проектирование и разработка пользовательского интерфейса

Общие принципы проектирования пользовательских интерфейсов

- Программа должна помогать выполнить задачу, а не становиться этой задачей.
- Прозрачность интерфейсов
 - Интерфейс должен быть лёгким для освоения
 - Интерфейс должен быть интуитивно понятным
- Не считать пользователя дураком
 - Интерфейс не должен быть перегружен «предупреждениями» и «подсказками»
 - Если действие не противоречит логике программного обеспечения, оно должно быть выполнимо (пример – блокнот из Windows)
- Не давать пользователю считать компьютер и программный продукт дураком
 - Интерфейс не должен быть перегружен «пустыми» вопросами
 - Интерфейс должен быть обучаемым

Стратегия разработки интерфейса

- Разрабатывать интерфейс как отдельный компонент системы
- Учитывать возможности аппаратных и программных средств
- Последовательность в разработке
- Использовать принятые принципы разработки интерфейса
- Понять задачу и пользователя
- Привлекать пользователя
- Предусмотреть средства адаптации в рамках интерфейса

Рекомендации к проектированию интерфейсов

- Особенности человеческого внимания

- Локус внимания

При периодическом переключении внимания (например, с рабочей области на область вывода информации о ходе работы), снижается эффективность (теряется оперативная информация)

- Формирование привычек

При выполнении постоянных действий, у пользователя формируются привычки
=> необходима возможность обратимости команд и действий.

- Проблема модальности

Одно и то же действие может вызывать различный результат, что замедляет работу.

- Отсутствие модальности

Исключение ошибок во взаимодействии с программой и развитие «полезных» привычек

- Сохранность пользовательских данных

- Автоматическое сохранение изменений
 - Бэкап
 - Обучение и автоматическая подстановка
 - Запоминание состояния

Рекомендации к проектированию интерфейсов

- **Формирование команд по принципу «объект-действие»**

При таком формировании команд, локус внимания уже сосредоточен на объекте, что упрощает понимание действия

- **Монотонность**

Ограниченное число возможностей выполнения одного и того же действия

- **Видимость**

Своевременное и эффективное информирование пользователя о состоянии системы и способах управления и воздействия на систему

- **Состоятельность**

По внешнему виду элемента управления, пользователь должен однозначно определить, каким образом с ним необходимо взаимодействовать

- **Поясняющие сообщения и тексты**

Во всем интерфейсе используйте **понятные** для пользователя термины

Выбирайте **правильный тон** в сообщениях и приглашениях

Страхуйте от **проблем и ошибок**

Этапы разработки ПИ

Этапы разработки пользовательского интерфейса:

- I. Сбор и анализ информации, поступающей от пользователей
- II. Разработка пользовательского интерфейса
- III. Построение пользовательского интерфейса
- IV. Подтверждение качества ПИ, тестирование.

I. Сбор и анализ информации, поступающей от пользователей

1. Определение профилей(моделей) пользователей
2. Анализ задач, стоящих перед пользователями
3. Сбор требований
4. Анализ рабочей среды пользователей
5. Соответствие требований пользователей стоящим перед ним задачам

Определение профиля пользователя

1. Психофизиологические характеристики.

Влияют на:

- возможность использования определенных каналов восприятия
- масштабирование элементов интерфейса и пр.

2. Социально-демографические сведения.

Влияют на:

- ❑ глубину и уровень детализации программы
- ❑ набор используемых метафор
- ❑ язык представления информации
- ❑ стиль взаимодействия с программой
- ❑ лексику элементов интерфейса программы

3. Опыт работы с компьютерами и прикладными программами.

Влияет на:

- ❑ содержание и время обучения
- ❑ объем и структуру справочной системы
- ❑ организацию протокола взаимодействия (привычность)

II. Разработка пользовательского интерфейса

1. Определение цели с точки зрения удобства применения продукта
2. Разработка задач и сценария действий пользователей
3. Определение объектов и операций интерфейса
4. Определение визуального представления
5. Разработка меню объектов и окон
6. Оптимизация визуальной разработки

III. Построение пользовательского интерфейса

1. выделить объекты, данные и действия из сценариев и задач, стоящих перед пользователями;
2. просмотреть и уточнить список объектов и действий совместно с пользователями;
3. начертить диаграмму взаимодействия между объектами;
4. заполнить матрицу прямого манипулирования объектами.

Этапы проектирования интерфейса

- Анализ трудовой деятельности, объединение функционала в роли
- Формулировка требований к функционалу пользователя
- Разработка обобщённого сценария взаимодействия пользователя с программным модулем
- Разработка макетов и прототипов интерфейса, их оценка в деловой игре и выбор окончательного варианта
- Реализация интерфейса в программном коде, создание тестовой версии
- Разработка средств поддержки пользователя и их встраивание в программный код
- Тестирование
- Подготовка пользовательской документации и разработка программы обучения

Проектирование иерархического меню ПИ

Порядок проектирования меню предусматривает следующую последовательность работ:

- проектирование содержания меню;
- проектирование формы меню;
- программное обеспечение меню.

При проектировании меню необходимо руководствоваться этими рекомендациями:

1. Количество уровней в меню должно быть не более 2-3.
2. Пользователь должен знать, в какой точке иерархического меню он находится.
3. Пункты меню не нумеруются.
4. Название пунктов горизонтального меню должно быть коротким – из одного слова.
5. Заглавной должна быть только первая буква названия пункта.
6. Для выбора пункта всплывающего меню должна быть предназначена «горячая» клавиша, поскольку путь к нему через главное меню может быть долгим.
7. Пункты, к которым часто обращаются, должны быть расположены в начале меню. Если присутствует пункт «Помощь», то он располагается в начале главного меню, а пункт «Выход» – в конце.
8. Логически взаимосвязанные пункты всплывающего меню объединяются в группы сплошной горизонтальной линией и могут получить свои подзаголовки.

Проектирование экранных форм ПИ

Порядок проектирования экранной формы подразумевает следующие этапы:

- проектирование содержания экранной формы;
- проектирование ее формы представления (формы экрана);
- программное обеспечение экранной формы.

По назначению можно выделить 4 класса экранных форм:

1. Для ввода информации в БД, т.е. для формирования и ведения БД.
2. Для ввода параметров обработки информации по задаче и идентификаторов запросов (условия выборки).
3. Для вывода результатов решения задачи (отчетов) и справочной информации.
4. Комбинированные формы, предусматривающие многоцелевое назначение (диалоговые окна).

Таблица 1 – Контроль количественных реквизитов

Наименование	Машинное имя (Name)	Диапазон (Range)	
		min	max

Таблица 2 – Реквизитный состав экранной формы

Наименование реквизита	Машинное имя	Тип	Ширина	Число дробных позиций	Метод контроля	Имя файла	Примечание

Зоны экранных форм

Информация на экране может размещаться **в 4 зонах**:

- 1. Заголовок экранной формы** содержит ее наименование и характеризует путь к ней по иерархическому меню и располагается в верхней строке экрана.
- 2. Предметная часть экранной формы** предназначена для размещения вводимой информации, идентификаторов объектов, значений показателей, параметров обработки и выборки и т.д. Предметная часть строится по анкетной, зональной или табличной форме.
- 3. Зона управляющих элементов** включает в себя различные виды меню. Это зона действий (транзакций) конечного пользователя.
- 4. Зона сообщений** содержит подсказки и сообщения об ошибках. Обычно она занимает нижнюю строку экрана.

Типовые формы документов

Наименование типов реквизитов	$X(1)$...	$X(i)$...	$X(n)$
Значения реквизитов	$X(1,1)$...	$X(i,1)$...	$X(n,1)$

Рисунок 1 – Линейная форма

Наименование типов реквизитов	Значения реквизитов
$X(1)$	$X(1,1)$
...	...
$X(i)$	$X(i,1)$
...	...
$X(n)$	$X(n,1)$

Рисунок 2 – Анкетная форма

	Наименование типов реквизитов				
Значения реквизитов	$X(1)$...	$X(i)$...	$X(n)$
		
	$X(1,j)$...	$X(i,j)$...	$X(n,j)$
		
	$X(1,m)$...	$X(i,m)$...	$X(n,m)$

Рисунок 3 – Табличная форма

Проектирование отчетов ПИ

Проектирование отчетов (машинограмм) состоит из следующих этапов:

- проектирование содержания отчета.
- проектирование формы отчета.
- программное обеспечение формирования отчета.

Таблица 3 – Реквизитный состав отчёта

Наименование реквизита	Машин ное имя	Тип	Шир ина	Число дробных позиций	Имя файла	Примеч ание
					Выражение для вычисления	

Структура формы отчета содержит заголовок, предметную часть и основание.

Заголовок и **основание** печатаются иногда в виде отдельного титульного листа, который содержит следующую информацию: наименование организации, название задачи, подписи и дату решения задачи.

Предметная часть печатается в табличной форме, часто на нескольких страницах. Предметная часть не разлиновывается, а иногда и не разграфляется. Горизонтальными линиями выделяются лишь шапка таблицы и итоговые строки.

Контрольные вопросы:

- 1) Понятие пользовательского интерфейса с точки зрения организации взаимодействия с ИС.
- 2) Перечислите требования, предъявляемые к ПИ, для обеспечения максимального удобства пользователю при работе с программой.
- 3) Опишите 3 правила разработки пользовательского интерфейса.
- 4) Перечислите принципы дают пользователю контроль над системой.
- 5) Перечислите принципы, позволяющие снизить загрузку памяти пользователя.
- 6) Состав пользовательского интерфейса: компоненты, понятие процесса диалога, сообщения, типы сообщений, типы диалога.
- 7) Опишите критерии хорошего диалога, виды справок.
- 8) Эвристические правила Якоба
- 9) Опишите какие принципы построения интерфейсов существуют и их суть.
- 10) Какие требования предъявляются к созданию меню?
- 11) Перечислите общие принципы проектирования пользовательских интерфейсов
- 12) Каких приемов следует придерживаться при проектировании интерфейсов?
- 13) Перечислите этапы разработки пользовательского интерфейса
- 14) Опишите этапы проектирования интерфейса
- 15) При проектировании меню необходимо...
- 16) 4 класса экранных форм, зоны экранных форм
- 17) Структура формы отчета