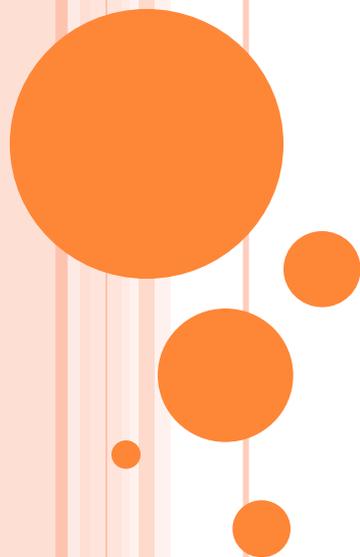


ЧЕЛОВЕКО-МАШИННОЕ ВЗАИМОДЕЙСТВИЕ

ПРОЕКТИРОВАНИЕ ПОЛЬЗОВАТЕЛЬСКОГО ИНТЕРФЕЙСА



ПРЕДПОСЫЛКИ

При проектировании пользовательского интерфейса для сложных проектов существует ряд негативных факторов:

- неизбежность появления ошибок;
- ряд ошибок заставляет пересматривать весь проект;
- картина не меняется при повторных заказах.

- ISO 9241 — содержит требования к эргономике визуальных дисплейных терминалов для офисной работы.
- ISO 9241-210:2010 (13407) — описан процесс проектирования интерактивных систем, ориентированных на пользователей.
- ISO 18529 — эргономика человеко-компьютерного взаимодействия — описание процесса проектирования интерфейсов, ориентированных на пользователей.
- ISO 14915 — эргономика программного обеспечения мультимедийных пользовательских интерфейсов.
- ISO 16071 — эргономика взаимодействия "человек-система". Руководящие указания по доступу к интерфейсам "человек-машина".
- ISO 16982 — эргономика взаимодействия человек-система. Методы, основанные на удобстве применения, для обеспечения проектирования, ориентированного на человека.
- ISO 20282 — юзабилити повседневных вещей.

ISO 11064-7-2010



- ❑ **валидация** (validation): Подтверждение на основе объективных данных, что установленные требования в условиях намеченного использования или применения выполнены.
- ❑ **верификация** (verification): Подтверждение на основе объективных данных, что установленные требования были выполнены.

ЖИЗНЕННЫЙ ЦИКЛ ПРОГРАММНОГО ПРОДУКТА

Под **жизненным циклом** понимается последовательность процессов, действий и задач, которые осуществляются в ходе разработки, эксплуатации (использования) и сопровождении программного продукта в течение всей его жизни, от определения требований до завершения использования.

ISO 15288:2002(8) – Информационная технология. Системная инженерия. Процессы жизненного цикла систем.

ВОЗМОЖНОСТЬ АДАПТАЦИИ СТАНДАРТОВ

- Практически все стандарты (даже военные) предусматривают возможность их адаптации к особенностям конкретного проекта при условии соблюдения основных требований к технологии и показателям качества продукта.
- Например, на этапе формирования требований к системе должны учитываться:
 - • область применения системы;
 - • требования пользователя (заказчика) к функциональным возможностям системы, к уровню ее безопасности и защищенности;
 - • эргономические требования и требования к уровню квалификации пользователей;
 - • степень документированности системы;
 - • организация сопровождения и т.д.

ТРЕБОВАНИЯ К СИСТЕМЕ

на этапе формирования требований к системе должны учитываться:

- область применения системы;
- требования пользователя (заказчика) к функциональным возможностям системы, к уровню ее безопасности и защищенности;
- эргономические требования и требования к уровню квалификации пользователей;
- степень документированности системы;
- организация сопровождения и т.д.

ПРОЕКТИРОВАНИЕ

- На этой фазе определяется общая концепция создаваемого продукта.
- Проектирование включает: определение цели и характеристик приложения, определение пользователей (потенциальных) и их задачи, намерения, цели.
- Это предполагает **учет таких показателей**: возраст пользователей, их пол, экспертные знания, уровень опыта, физические ограничения, специальные потребности и т.д.
- Решению указанной проблемы способствует наблюдение за работой пользователей при выполнении ими задач в данной предметной области.

ПРОТОТИПИРОВАНИЕ

- В зависимости от уровня подготовки и сложности приложения его прототип может быть представлен либо **в виде иллюстраций** интерфейса (комиксы), либо **в виде специальных схем** (сети Петри).
- Прототип играет важную роль:
 - возможность для обсуждения создаваемого приложения как внутри группы разработчиков, так и с потенциальными пользователями;
 - может помочь определить характер потока заданий и лучше представить себе то, чем вы занимаетесь, что особенно полезно в начале процесса разработки.
- Форма представления прототипа зависит от цели разработки.

ИСПЫТАНИЕ ПРОГРАММНОГО ПРОДУКТА

- Ряд технологий предполагает достаточно активное **привлечение пользователя к процессу разработки.**
- Испытание программного продукта отличается от его отладки:
 - различны цели этих двух процессов: отладка имеет целью выявление дефектов (ошибок) программирования, а испытания оценивает насколько полно разработанное приложения (в частности, его интерфейс) отвечает потребностям и ожиданиям пользователя;
 - отладку выполняет непосредственно его разработчик, а основным действующим лицом при проведении испытаний является потенциальный пользователь (заказчик).

ИСПЫТАНИЕ ПРОГРАММНОГО ПРОДУКТА

- Испытания могут проводиться для двух или более альтернативных вариантов реализации создаваемого приложения с целью выявления наиболее удачного именно с точки зрения пользователя и решаемых им задач.
- **Оценки испытаний:**
 - **количественные (объективные)** - данные о работе приложения в тех или иных ситуациях;
 - **качественные** - информация, отражающая субъективное восприятие пользователем предлагаемого варианта приложения, его удовлетворенность, а также перечень проблем, которые на его взгляд могут иметь место при реальной эксплуатации программного продукта.

ПОВТОРНОЕ ВЫПОЛНЕНИЕ ЭТАПОВ РАЗРАБОТКИ

- Поскольку испытания часто обнаруживают те или иные слабости проекта, или, по крайней мере, обеспечивают получение дополнительной информации, которую вы захотите использовать, почти всегда оказывается необходимым возврат к одному из предыдущих этапов разработки (а иногда и в начальную точку) и проведение повторных испытаний.
- Так может продолжаться до тех пор, пока и разработчик, и потенциальные пользователи не будут полностью удовлетворены полученными результатами.

ПОТРЕБИТЕЛЬСКИЕ СВОЙСТВА ПРИЛОЖЕНИЯ

- Основная цель испытаний — определить, насколько полно разработанный интерфейс отвечает потребностям и ожиданиям пользователя. Основным направлением испытаний приложения является оценка его «потребительских свойств» (**Usability**).
- Основой для проведения оценки должны служить данные о том, как пользователи обычно выполняют ту работу, которую призвано автоматизировать создаваемое приложение.
- Чем чаще и корректнее будет проводиться оценка, тем выше будет качество разработки.

ТЕХНИКА ПРОВЕДЕНИЯ ИСПЫТАНИЙ USABILITY

- Проведение испытаний потребительских свойств приложения требует привлечения значительных дополнительных сил и средств, в том числе специалистов тестовых лабораторий, имеющих в своем распоряжении соответствующее оборудование для регистрации результатов испытаний.
- Тем не менее, даже обычные «офисные инструменты» (магнитофон, секундомер и записная книжка) могут принести существенную пользу.
- Используемые тесты не должны быть «всеохватывающими». Значительно более полезны быстрые итеративные тесты, ориентированные на исследование конкретных проблем.

НЕОЖИДАЕМЫЕ ИЗМЕНЕНИЯ В ПРОЕКТЕ

- На разработку продукта оказывают влияние многие дополнительные факторы. Например, **маркетинговые** соображения могут потребовать сокращения сроков разработки, или сравнительные оценки с аналогичными продуктами могут заставить реализовать в создаваемом ПО **дополнительные** характеристики.
- После выхода в свет официальной версии продукта **труднее** устранить проблемы, оставшиеся нерешенными на стадии разработки, поскольку пользователи могут приспособиться, или даже «подчиниться» имеющимся недостаткам вашего ПО.
- **Чтобы сделать нечто простым** в использовании, часто **требуется приложить много сил** и создать весьма сложное изделие с точки зрения его внутренней организации.

ПРАВИЛА ВЕДЕНИЯ ДИАЛОГА

- Интерфейсы взаимодействия пользователя с компьютером – диалоговые интерфейсы.
- Правила ведения диалога:
 - участники диалога должны понимать язык друг друга;
 - они не должны говорить одновременно;
 - очередное высказывание должно учитывать как общий контекст диалога, так и последнюю информацию, полученную от собеседника;
 - участники должны придерживаться единой терминологии;
 - применение дополнительных выразительных средств способствует лучшему взаимопониманию.

ЭТАПЫ ПРОЕКТИРОВАНИЯ ПОЛЬЗОВАТЕЛЬСКОГО ИНТЕРФЕЙСА

- При проектировании пользовательского интерфейса необходимо определить:
 - структуру диалога;
 - возможный сценарий развития диалога;
 - содержание управляющих сообщений и данных, которыми могут обмениваться человек и приложение (семантику сообщений);
 - визуальные атрибуты отображаемой информации (синтаксис сообщений).

ВЫБОР СТРУКТУРЫ ДИАЛОГА

- Выбор структуры диалога — это первый из этапов, который должен быть выполнен при разработке интерфейса:
 - диалог типа «вопрос - ответ»;
 - диалог на основе меню;
 - диалог на основе экранных форм;
 - диалог на основе командного языка.

ДИАЛОГ ТИПА «ВОПРОС - ОТВЕТ»

- Структура диалога типа «вопрос-ответ» (Q&A) основана на аналогии с обычным интервью. Система берет на себя роль интервьюера и получает информацию от пользователя в виде ответов на вопросы.
- Структура Q&A предоставляет естественный механизм ввода как управляющих сообщений (команд), так и данных.
- Нет никаких ограничений на диапазон или тип входных данных, которые могут обрабатываться.
- Структура Q&A не гарантирует минимального объема ввода, оцениваемого по количеству нажатий клавиш, однако при подходящем подборе сокращений можно уменьшить любую избыточность.
- Существуют системы, ответы в которых даются на естественном языке, но чаще используются предложения из одного слова с ограниченной грамматикой.

Q&A

- **Недостаток:** даже если ввод происходит достаточно быстро, для человека, который уже знает, какие вопросы задает система и какие ответы нужно на них давать, отвечать на всю серию вопросов довольно утомительно.
- С появлением графического интерфейса структура Q&A несколько устарела.
- **Достоинства:**
 - может удовлетворить требования различных пользователей и типов данных;
 - особенно уместна при реализации диалога с множеством «ответвлений», т.е. в тех случаях, когда на каждый вопрос предусматривается большое число ответов, каждый из которых влияет на то, какой вопрос будет задан следующим.

ДИАЛОГ НА ОСНОВЕ МЕНЮ

- Меню является наиболее популярным вариантом организации запросов на ввод данных во время диалога, управляемого компьютером.
- Несколько основных форматов представления меню на экране:
 - список объектов, выбираемых прямым указанием, либо указанием номера (или мнемонического кода);
 - меню в виде блока данных;
 - меню в виде строки данных;
 - меню в виде пиктограмм.

Традиционная структура меню недостаточно гибка и не в полной мере согласуется с методами адаптации диалога, такими, например, как опережающий ввод, с помощью которого можно ускорить темп работы подготовленного пользователя.

ДИАЛОГ НА ОСНОВЕ ЭКРАННЫХ ФОРМ

- Как структура типа «вопрос — ответ», так и структура типа меню предполагают обработку на каждом шаге диалога единственного ответа.
- Диалог на основе экранных форм допускает обработку **на одном шаге диалога нескольких ответов**.
- В некоторых системах информация, вводимая пользователем, становится доступной только после нажатия клавиши «ввод» по окончании заполнения формы. Вопрос о том, надо ли проверять ответ непосредственно или отложить проверку до окончания ввода всех ответов, решить непросто: сообщения об ошибках, выводимые непосредственно после ответа, могут отвлечь внимание, но могут оказать и положительное влияние.
- В тех случаях, когда информация для ввода выбирается из некоторого целостного документа, проверку лучше отложить до конца заполнения формы, чтобы не прерывать процесс ввода; если же такой целостности нет, то проверку следует выполнять сразу после ввода ответа (после заполнения очередного поля).
- Часто все необходимые единицы ввода нельзя отобразить одновременно в пределах одного экрана (или окна), и их необходимо разделить на группы, которые отображаются на последовательности экранов (окон).

ДИАЛОГ НА ОСНОВЕ КОМАНДНОГО ЯЗЫКА

- Структура диалога на основе командного языка столь же распространена, что и структура типа меню. Она очень часто используется в операционных системах.
- Обеспечивает **более широкие возможности** выбора в любой точке диалога и не требует иерархической организации обслуживающих его программ.
- Программная система **может поддерживать достаточно большое количество команд**, но на практике следует ограничивать их число, чтобы не перегружать память пользователя.
- Не отличается хорошей поддержкой пользователя и пригодна в основном для подготовленных специалистов.

УПРАВЛЕНИЕ КОМАНДАМИ

- В интерфейсах на основе языков команд управление данными достигается с помощью **составных командных строк**, где ключевое слово для обозначения команды (что делать) предшествует списку параметров (входным данным).
- Параметры в списке можно задавать в одной из двух форм:
 - **позиционной** – назначение параметра определяется по его месту в командной строке, уменьшают объем вводимой информации, но должны указываться в строго определенном порядке ;
 - **ключевой** – каждое значение предваряется определенным идентификатором, который определяет его назначение, уменьшают нагрузку на память пользователя в том отношении, что отпадает необходимость в запоминании порядка их следования, но пользователю необходимо запомнить множество ключевых слов.
- Многие командные языки поддерживают **макросы**, которые расширяют функциональные возможности диалога без увеличения количества команд.

ДОСТОИНСТВА ДИАЛОГА НА ОСНОВЕ КОМАНД

- Структура на основе языка команд по своим возможностям **самая быстрая и гибкая** из всех структур диалога.
- Большинство пользовательских интерфейсов на базе «естественного» языка реализуется с помощью языков команд с очень большим набором ключевых слов.
- Подготовленный пользователь испытывает удовольствие от ощущения того, что он управляет системой, а не наоборот.

ТАБЛИЦА ВЫБОРА ТИПА ДИАЛОГА

Критерии	Выбор пользова- теля	Тип диалога			
		меню	воп рос — отве т	язык кома нд	заполнен ие экранны х форм
Цель: Запрос Вычисления Сложный выбор ввод данных ввод данных (большой объем)		+ + + +	+ + + + +	+ + + + +	+ + +
Тип пользователя: Программист Непрограммист Имеет опыт работы нет опыта работы		 + +	 + +	+ + *	+ + *
Время обучения: очень малое менее 1 дня более 1 дня		+ +	+ +	 ** +	 ** +
Результат оценки					

* — использование этого типа диалога данной категорией пользователей требует наличия системы помощи;

** — использование средств системы возможно только в ограниченном объеме.

ВЫБОР ДИАЛОГА

- Основная ценность таблицы состоит в том, что ее можно использовать как исходный вариант выбора типа диалога, либо как средство окончательной проверки соответствия выбранного типа диалога рассматриваемым критериям.
- Если предполагается, что одни пункты более важны, чем другие, можно брать их с разными весовыми коэффициентами.
- В таблицу могут быть включены и другие критерии выбора (наличие инструментальных средств разработки интерфейса, характер пользователя, ограничения по имеющимся ресурсам и т.д.).



Данные: new_sta 10n * 10n

Чис зна	1 VAR1	2 VAR2	3 VAR3	4 VAR4
1	4.000	20.000	4.000	18.000
2	18.000	9.000	14.000	
3	15.000	3.000	10.000	
4	11.000	1.000	8.000	
5	7.000	1.000	2.000	
6	1.000	8.000	13.000	
7	2.000	11.000	16.000	
8	8.000	12.000	17.000	
9	3.000	13.000	11.000	
10	4.000	13.000	7.000	

- Быстрые стат. графики...
- Пользовательские графики
- Быстрые основные статистики...
- Спецификации переменных...
- Текстовые значения...
- Пересчитать...
- Изменить переменные
- Изменить наблюдения
- Имена наблюдений...
- Заполнить/стандартизировать блок
- Статистики блока по столбцам
- Статистики блока по строкам**
 - Средние
 - Медианы
 - Ст. откл.
 - Число набл. N
 - Суммы**
 - Мин. знач.
 - Макс. знач.
 - 25%
 - 75%
 - Все
 - Графики
- Вырезать Ctrl+X
- Копировать Ctrl+C
- Копировать содержание
- Вставить Ctrl+V
- Очистить Del

Внешний вид

Разбиение панелей

- Вертикальное
- Горизонтальное
- Равные размеры

040[<] = [>]041

Цветовыделение

- Права доступа
- Типы файлов

Прочие настройки

- Линейка меню
- Мини-статус
- Командная строка
- Метки клавиш
- Строка подсказки
- Подсказка в xterm

[< Дальше >] [Сохранить] [Прервать]

Album Photo **Transition & Music** Album Theme Album Preview

Add Trim Ripper Record More... **установка**

Background music:

Name	Length	Time in Track	Path
✓ default.mp3	40' of 40'	00:00 - 00:40	C:\Program Files\DVD Photo Slideshow Professional\music

Нажимаем кнопку "ADD" чтобы добавить музыку в проект

Fade in/out Loop audio Transitions:

Transition effects:

(S)Dbl. Sweep - Top Left (2)Dbl. Sweep - Top Left Dbl. Sweep - Bottom Left (S)Dbl. Sweep - Bottom Left (2)Dbl. Sweep - Bottom Left Door Wipe - Top (S)Door Wipe - Top

Door Wipe - Left (S)Door Wipe - Left Door Wipe - Bottom (S)Door Wipe - Bottom Door Wipe - Right (S)Door Wipe - Right Push - L. to R.

Push - R. to L. Push - Up to Down Push - Down to Up Expand From Left **Expand From Right** Expand From Top Expand From Bottom

Expand From Horizon Center Expand From Vertical Center Expand From Center Explode Wand Wipe Hor. Cross Wipe Wipe Center

Wipe Block Wipe Snow Wipe Grid Ball Wipe Diamond Wipe Twirl Twirl Center Expand

перетаскиваем мышкой нужный эффект

6 00:27 7 00:33 8 00:38 9 00:39 10 00:40 11 00:40 Album Credits 10'

00:00 - 00:40 (default.mp3) 62:39

Web Application Starter Project

Back Forward Refresh Stop Compile/Browse Google Web Toolkit

http://localhost:8080/MyApplication.html Go

Предупреждение о выходе

Хотите записать изменения перед выходом?



РАЗРАБОТКА СЦЕНАРИЯ ДИАЛОГА

- Диалог во времени можно рассматривать как последовательность переходов системы из одного состояния в другое.
- **Сценарий диалога** – это все возможные состояния диалога и пути перехода из одного состояния в другое.
- Цели разработки сценария диалога:
 - выявление и устранение возможных тупиковых ситуаций в ходе развития диалога;
 - выбор рациональных путей перехода из одного состояния диалога в другое (из текущего в требуемое);
 - выявление неоднозначных ситуаций, требующих оказания дополнительной помощи пользователю.

СЛОЖНОСТЬ РАЗРАБОТКИ СЦЕНАРИЯ

- Сложность разработки сценария определяется двумя факторами:
 - функциональные возможности создаваемого приложения (т.е. числом и сложностью реализуемых функций обработки информации);
 - степенью неопределенности возможных действий пользователя.
- В свою очередь, степень неопределенности действий пользователя зависит от выбранной структуры диалога.
- Наибольшей детерминированностью обладает диалог на основе меню, наименьшей — диалог типа «вопрос-ответ», управляемый пользователем.

ШАГ ДИАЛОГА

Независимо от способа описания сценария его основной структурной единицей является **шаг диалога**, соответствующий одному акту взаимодействия пользователя с системой.



ТЕМП ВЕДЕНИЯ ДИАЛОГА

- При разработке сценария диалога должны учитываться такие **психофизиологические особенности** потенциальных пользователей, как моторные навыки, время реакции, восприимчивость цветовой гаммы и т.д.
- Темп ведения диалога зависит от характеристик аппаратных и программных средств ЭВМ, а также от специфики решаемых задач.
- **Время ответа (отклика) системы** – это интервал между событием и реакцией системы на него. Данная характеристика интерфейса определяет задержку в работе пользователя при переходе к выполнению следующего шага задания.

ВРЕМЯ ОТВЕТА ДИАЛОГА

- Время ответа должно соответствовать естественному ритму работы пользователей. В обычном разговоре люди ожидают ответа около 2 секунд и ждут того же при работе с компьютером.
- Время ожидания зависит от их состояния и намерений пользователя.
- На представления пользователя оказывает влияние его предшествующий опыт работы с системой.
- Обычно человек может одновременно запомнить сведения о пяти-девяти предметах.
- Хранение данных в кратковременной памяти ограничено по времени: около 2 секунд для речевой информации и 30 секунд для сенсорной.

КЛАУЗА И ЗАКРЫТИЕ

- Хранение данных в кратковременной памяти ограничено по времени: около 2 секунд для речевой информации и 30 секунд для сенсорной.
- Люди имеют склонность разбивать свою деятельность на этапы, соответствующие порциям информации, которые они могут хранить одновременно в памяти.
- Завершение очередного этапа называется **клаузой**. Задержки, препятствующие наступлению клаузы, очень вредны и неприятны, так как содержимое кратковременной памяти требует постоянного обновления и легко стирается под влиянием внешних факторов.
- После **клаузы** задержки вполне приемлемы и даже необходимы. Завершение задачи, ведущее к отдыху, называют **закрытием**. В этот момент исчезает необходимость дальнейшего хранения информации и человек получает существенное **психологическое облегчение**.
- Так как пользователи интуитивно стремятся к закрытию в своей работе, следует **делить диалоги на фрагменты**, чтобы пользователь мог вовремя забывать промежуточную информацию.

ИНТЕРАКТИВНАЯ СИСТЕМА

Имеющиеся результаты исследований позволили выработать следующие рекомендации по допустимому времени ответа интерактивной системы:

- 0,1 ... 0,2 с — для подтверждения физических действий (нажатие клавиши, работа со световым пером, «мышью»);
- 0,5 ... 1,0 с — для ответа на простые команды (например, от момента ввода команды, выбора альтернативы из меню до появления нового изображения на экране);
- 1 ... 2 с — при ведении связного диалога (когда пользователь воспринимает серию взаимосвязанных вопросов как одну порцию информации для формирования одного или нескольких ответов, задержка между следующими друг за другом вопросами не должна превышать указанную длительность);
- 2 ... 4 с — для ответа на сложный запрос, состоящий в заполнении некоторой формы. Если задержка не влияет на другую работу пользователя, связанную с первой, могут быть приемлемы задержки до 10 с;
- более 10 с — при работе в мультизадачном режиме, когда пользователь воспринимает данную задачу как фоновый процесс.

МЕТОДЫ РАЗРАБОТКИ ГИБКОГО ИНТЕРФЕЙСА

- **Гибкость интерфейса** – способность приложения адаптироваться (пользователем или автоматически) к любому возможному уровню подготовки пользователя.
- Три вида адаптации:
 - **фиксированная** – пользователь явно выбирает уровень диалоговой поддержки.
 - **полная** – диалоговая система стремится построить модель пользователя, которая по мере обучения последнего и определяет стиль диалога.
 - **косметическая** – призвана обеспечить гибкость диалога без учета поведения пользователя, но и без однозначного выбора им конкретного стиля диалога.

ФИКСИРОВАННАЯ АДАПТАЦИЯ

Фиксированная адаптация – это такое поведение диалоговой системы, при котором пользователь явно выбирает уровень диалоговой поддержки.

Например, используется правила двух уровней:

- **подробный диалог** (для начинающего пользователя);
- **краткий диалог** (для подготовленного пользователя).

Правило двух уровней может быть расширено до правила N уровней диалога.

Недостатки:

- не учитывается факт, что навыки накапливаются постепенно;
- пользователь может хорошо знать одну часть системы и совсем не знать другую;
- пользователь сам определяет уровень своей подготовки, что снижает объективность оценки.

ПОЛНАЯ АДАПТАЦИЯ

- **Полная адаптация** – это поведение диалоговой системы, при котором система сама стремится построить модель пользователя, в соответствии с которой по мере обучения последнего и определяет стиль диалога.
- Одна из основных проблем – это распознавание характеристик пользователя. Для ее решения необходимо определить, что использовать в качестве таких характеристик: время, затрачиваемое пользователем на ответ, количество его обращений за помощью или характер ошибок и тип запрашиваемой помощи.
- В настоящее время полная (автоматическая) адаптация практически ни в одной диалоговой системе не реализована.

КОСМЕТИЧЕСКАЯ АДАПТАЦИЯ

- **Косметическая адаптация** – это такое поведение диалоговой системы, которое обеспечивает гибкость диалога без однозначного выбора пользователем конкретного стиля диалога и без учета поведения пользователя.
- **Методы косметической адаптации:**
 - • использование умолчаний;
 - • использование сокращений;
 - • опережающий ввод ответов;
 - • многоуровневая помощь;
 - • многоязычность.

ВИЗУАЛЬНЫЕ АТТРИБУТЫ ОТОБРАЖАЕМОЙ ИНФОРМАЦИИ

- К визуальным атрибутам отображаемой информации относятся:
 - • взаимное расположение и размер отображаемых объектов;
 - • цветовая палитра;
 - • средства привлечения внимания пользователя.
- Проектирование размещения данных на экране предполагает выполнение следующих действий:
 - 1) Определение состава информации, которая должна появляться на экране;
 - 2) Выбор формата представления этой информации;
 - 3) Определение взаимного расположения данных (или объектов) на экране;
 - 4) Выбор средств привлечения внимания пользователя;
 - 5) Разработка макета размещения данных на экране;
 - 6) Оценка эффективности размещения информации.

ПРИНЦИПЫ РАСПОЛОЖЕНИЯ ИНФОРМАЦИИ

- Общие принципы расположения информации на экране должны обеспечивать для пользователя:
 - возможность просмотра экрана в логической последовательности;
 - простоту выбора нужной информации;
 - возможность идентификации связанных групп информации;
 - различимость исключительных ситуаций (сообщений об ошибках или предупреждений);
 - возможность определить, какое действие со стороны пользователя требуется (и требуется ли вообще) для продолжения выполнения задания.

КАКУЮ ИНФОРМАЦИЮ РАЗМЕЩАТЬ?

- Вопрос о том, какая информация подлежит отображению, решается в зависимости от специфики выполняемого пользователем задания.
- Здесь существенную роль играет правильное разбиение задания на операции (этапы), не требующие одновременного присутствия большого объема данных на экране.
- Ограниченность кратковременной памяти, способной хранить одновременно не более пяти — девяти объектов.
- Если вся информация исходного документа не помещается на одном экране, некоторые элементы данных могут повторяться на других экранах для сохранения целостности и последовательности обработки.
- Если в выделении логических групп есть сомнения, необходим тщательный учет пожеланий заказчика или предоставление ему возможности самостоятельного формирования таких групп.

РАЦИОНАЛЬНОЕ РАЗМЕЩЕНИЕ ДАННЫХ НА ЭКРАНЕ

- Существуют некоторые правила, регулирующие плотность расположения данных на экране (или в пределах окна):
 - оставлять пустым приблизительно половину экрана (окна);
 - оставлять пустую строку после каждой пятой строки таблицы;
 - оставлять четыре-пять пробелов между столбцами таблицы;
 - фрагменты текста должны располагаться на экране так, чтобы взгляд пользователя сам перемещался в нужном направлении;
 - содержимое полей не должно «прижиматься» к краю экрана, а располагаться около его горизонтальных или вертикальных осей;
 - меню, содержащее относительно небольшой объем информации, должно смещаться в левую верхнюю часть экрана;
 - чтобы подчеркнуть симметрию, содержимое и наименования полей, относящихся к одной группе, должны выравниваться по вертикали;
 - по возможности необходимо выравнивать все логически связанные группы данных.

МОЗГ



Левое полушарие: обработка вербальной информации, аналитическое мышление, последовательная обработка информации, т. е. логика и рациональное мышление.

Правое полушарие: обработка невербальной информации, воображение, образное мышление и интуиция.

- В связи с этим можно рекомендовать текстовые сообщения группировать справа, а изображения — слева.
- У некоторых людей это распределение функций полушарий противоположно, у женщин асимметрия выражена слабее, чем у мужчин.

ВЫДЕЛЕНИЕ ИНФОРМАЦИИ

- **Выделение информации** — это использование таких атрибутов, которые позволяют привлечь внимание пользователя к некоторой области экрана.
- В качестве подобных атрибутов могут выступать:
 - **цвет** символов, цвет фона, уровень **яркости**, мерцание и применение различных **шрифтов** для выводимых **символов**, **подчеркивание**, вывод в инверсном виде, различные **рамки** и «тени».
- **Следует стараться использовать минимально необходимое число атрибутов** (чтобы привлечь внимание человека, достаточно лишь легонько его «коснуться»).

МЕТОДЫ ОЦЕНКИ ФОРМАТИРОВАНИЯ

- Один из возможных подходов к решению этой проблемы — отделить содержание от формы. Для этого применяются два метода:
 - **Метод прямоугольников** предполагает разбиение экрана на поля, каждое из которых заполняется произвольным текстом и отделяется от других по всему периметру, по крайней мере, одним пробелом. Через центр экрана мысленно проводятся оси, позволяющие оценить сбалансированность размещения полей.
 - **Метод выделенных точек** позволяет определить число и размещение областей экрана, к которым будет привлечено внимание пользователя (из-за повышенной яркости, цвета или мерцания символов). Для этого каждая область, требующая повышенного внимания, моделируется группой символов, отличных от пробела.
- Рассмотренные методы позволяют устранить грубые ошибки в форматировании экрана, однако лучший способ оценить его качество — дать возможность потенциальному пользователю поработать с системой.

GUI — GRAPHICAL USER INTERFACE

- В основу разработки практически любого графического пользовательского интерфейса положены три метафоры:
 - **«рабочий стол»**: для пользователя доступны как определенные источники информации, так и средства обработки этой информации (документы, представленные в различной форме: текст, таблицы, графики, рисунки и т.д., относящиеся либо к различным задачам, либо к разным аспектам одной и той же задачи);
 - **«работаешь с тем, что видишь»**,
 - **«видишь, что получил»**.

ПРЯМОЕ МАНИПУЛИРОВАНИЕ ОБЪЕКТАМИ

- **Прямое манипулирование объектами** обладает следующими достоинствами:
 - Обеспечивает визуальный контроль за выполняемыми операциями.
 - Позволяет легко восстановить предшествующее состояние «рабочего стола».
 - Позволяет решать различные задачи, используя ограниченный набор стандартных операции (открытие/закрытие окна, буксировка объекта, изменение атрибутов окна или объекта и т.п.).

- Прямое манипулирование снижает гибкость диалога, поскольку изначально графический интерфейс базируется на структуре меню.

МНОГООКОННОСТЬ

- **Многооконность** -- обеспечивает пользователю доступ к большему объему информации, чем при использовании одного экрана.
- Имея через окна доступ к нескольким источникам информации одновременно, пользователь может:
 - Объединять имеющуюся в них информацию. Например, изображения, полученные с помощью графического редактора, можно включить в текстовый документ.
 - Анализировать информацию представленную на различных уровнях детализации.

КОНЦЕПЦИЯ ИНТЕРФЕЙСА, УПРАВЛЯЕМОГО ДАННЫМИ

- ❑ **Разработка, управляемая данными** (сокращенно DCD — Data-Centered Design) означает, что проектирование интерфейса поддерживает такую модель взаимодействия пользователя с системой, при которой **первичными являются обрабатываемые данные**, а не требуемые для этого программные средства.
- ❑ При использовании DCD-технологии основным объектом является **документ**, который представляет собой некоторое абстрактное устройство хранения данных, используемых для выполнения заданий пользователей.
- ❑ Документ должен быть доступен как различным приложениям, используемым для его обработки, так и всем взаимодействующим пользователям.

ОБЪЕКТНО-ОРИЕНТИРОВАННЫЙ ПОДХОД

- В основу реализации DCD-технология обуславливает необходимость применения для проектирования GUI объектно-ориентированного подхода.
- С точки зрения пользовательского интерфейса, объектами являются любые устройства для хранения и обработки информации, включая ячейки, параграфы, символы, и т.д., а также документы, в которых они находятся.
- Объекты обладают определенными характеристиками, которые помогают нам понимать, что они собой представляют, и как они ведут себя в тех или иных ситуациях.

ХАРАКТЕРИСТИКИ ОБЪЕКТОВ, ИМЕЮЩИХ КОМПЬЮТЕРНОЕ ВОПЛОЩЕНИЕ

- ▣ **Свойства объектов.** Объекты имеют определенные характеристики или атрибуты, которые определяют их представление или возможные состояния (например, цвет, размер, дату модификации).
- ▣ **Операции над объектами.** Все действия, которые могут быть выполнены с (или над) объектом, считаются допустимыми операциями.
- ▣ **Связь (отношения) между объектами.** Любой объект тем или иным образом взаимодействует с другими объектами.

НАБОР, ОБЪЕДИНЕНИЕ, КОМПОЗИЦИЯ

- **Набор** – тип отношения, которое отражает наличие у объектов некоторых общих свойств.
- **Объединение** отражает более «тесное» отношение между объектами, при котором изменение объекта влияет на некоторый другой объект.
- **Композиция** имеет место в том случае, когда агрегация нескольких объектов может рассматриваться как новый объект со своим собственным множеством свойств и допустимых операций.
- **Контейнер** – объект, который содержит другие объекты (например, рисунок в документе или документ в папке могут рассматриваться как часть содержимого соответствующего контейнера).

КЛАССЫ

- Объекты одного типа имеют аналогичные свойства и поведение, что позволяет относить их к некоторому классу объектов.
- Совокупность объектов (возможно, различных типов) образует некоторую среду пользователя. Исходя из этого, большинство заданий пользователя могут быть представлены как определенная комбинация взаимосвязанных объектов.
- Например, обработка текстового документа может быть описана как композиция операций, выполняемых над его элементами (отдельными словами, параграфами и т.д.).

- При всех достоинствах объектного подхода к разработке интерфейса, его использование само по себе **не гарантирует требуемого качества интерфейса.**
- Для создания эффективного пользовательского интерфейса необходимо дополнить объектный подход тщательным проектированием всех компонентов интерфейса с ориентацией на потребности потенциального пользователя.

- При объектно-ориентированном проектировании интерфейса необходимо предусмотреть следующие работы:
 - **анализ целей пользователей** и особенностей выполняемых ими заданий,
 - определить **основные компоненты** или **объекты**, с которыми взаимодействует пользователь, а также характерные особенности объектов каждого типа.
 - выявить **перечень операций**, выполняемых над объектами, их влияние на состояние и свойства объектов.

- Следующим этапом проектирования GUI является компоновка и пространственное размещение на экране визуальных элементов интерфейса.
- Именно на этом этапе должны быть решены такие проблемы, как выбор цвета, размера и других атрибутов этих элементов, а также выбор средств и методов привлечения внимания пользователя к наиболее важной информации, отображаемой на экране.