

# Проектирование и дизайн пользовательского интерфейса

# Основные принципы создания интерфейса

1. Естественность (интуитивность)
2. Непротиворечивость
3. Неизбыточность
4. Непосредственный доступ к  
системе помощи
5. Гибкость

# **Естественность (интуитивность)**

**Работа с системой не должна вызывать у пользователя сложностей в поиске необходимых директив (элементов интерфейса) для управления процессом решения поставленной задачи.**

## **Непротиворечивость**

**Если в процессе работы с системой пользователем были использованы некоторые приемы работы с некоторой частью системы, то в другой части системы приемы работы должны быть идентичны. Также работа с системой через интерфейс должна соответствовать установленным, привычным нормам (например, использование клавиши Enter).**

# Неизбыточность

Это означает, что пользователь должен вводить только минимальную информацию для работы или управления системой. Например, пользователь не должен вводить незначимые цифры (00010 вместо 10). Аналогично, нельзя требовать от пользователя ввести информацию, которая была предварительно введена или которая может быть автоматически получена из системы. Желательно использовать значения по умолчанию где только возможно, чтобы минимизировать процесс ввода информации.

# **Непосредственный доступ к системе ПОМОЩИ**

**В процессе работы необходимо, чтобы система обеспечивала пользователя необходимыми инструкциями. Система помощи отвечает трем основным аспектам - качество и количество обеспечиваемых команд; характер сообщений об ошибках и подтверждения того, что система делает. Сообщения об ошибках должны быть полезны и понятны пользователю.**

# Гибкость

Насколько хорошо интерфейс системы может обслуживать пользователя с различными уровнями подготовки. Для неопытных пользователей интерфейс может быть организован как иерархическая структура меню, а для опытных пользователей как команды, комбинации нажатий клавиш и параметры.

# **Современные компоненты интерфейса пользователя**



# Размещение информации на экране

Количество информации, отображаемой на экране, называется экранной плотностью.

Чем меньше экранная плотность, тем отображаемая информация наиболее доступна и понятна для пользователя и наоборот, если экранная плотность большая, это может вызвать затруднения в усвоении информации и ее ясном понимании.

# Размещение информации на экране

Информация на экране может быть сгруппирована и упорядочена в значимые части. Это может быть достигнуто с использованием кадров, методов типа цветового кодирования, рамок, негативного изображения или других методов для привлечения внимания.

## Выделение элементов интерфейса яркостью

Для привлечения внимания к каким-либо элементам интерфейса можно воспользоваться выделением этих элементов большей яркостью на фоне других – более темных.

# Выделение элементов интерфейса яркостью

**Существует несколько способов выделения яркостью:**

- **Движение** (мигание или изменение позиции). Очень эффективный метод, поскольку глаз имеет специальный детектор для движущихся элементов;
- **Яркость**. Не очень эффективный метод, так как люди могут обнаружить всего лишь несколько уровней яркости;

# Выделение элементов интерфейса яркостью

- Цвет - использование цвета может быть чрезвычайно эффективно;
- Форма (символ, шрифт, форма символа).  
Используется для того, чтобы отличить различные категории данных;
- Использование различных алфавитов (шрифтов) в различных формах;

# Выделение элементов интерфейса яркостью

- Размер (текста, символов). Обычно применяют увеличение выделенного объекта в 1.5 раза;
- Оттенение (различная текстура объектов). Эффективный метод для привлечения внимания к какой-либо части экрана;
- Окружение (подчеркивание, рамки, инвертированное изображение).

# Использование цвета при проектировании эргономичного интерфейса

**Это наиболее эффективно когда цвет используется для:**

- группировки информации;
- выделения различий между информацией;
- выделения простых сообщений (ошибки, состояния и т.д.)

# Использование цвета при проектировании эргономичного интерфейса

## Принципы использования цвета:

- необходимо ограничить число цветов до 4 на экране и до 7 для последовательности экранов; для неактивных элементов нужно использовать бледные цвета;
- если цвет используется для кодировки информации, необходимо удостовериться, что пользователь правильно понимает код, например, просроченные счета выделяются красным цветом, а непросроченные – зеленым;



# Использование цвета при проектировании эргономичного интерфейса

- необходимо использовать цвета согласно представлениям пользователя, например, для картографа зеленый - лес, желтый - пустыня, синий - вода. Для химика, красный - горячий, синий - холодный;
- для отображения состояния: красный = опасность/стоп, зеленый = нормально/продолжение работы, желтый = предостережение;
- для привлечения внимания наиболее эффективны белый, желтый и красный цвета;

# Использование цвета при проектировании эргономичного интерфейса

- для упорядочения данных можно использовать спектр 7 цветов (радуга);
- для разделения данных необходимо выбрать цвета из различных частей спектра (красный / зеленый, синий / желтый, любой цвет / белый);
- для группировки данных, объединения и подобия нужно использовать цвета, которые являются соседями в спектре (оранжевые / желтые, синие / фиолетовые);

# Использование цвета при проектировании эргономичного интерфейса

Важно отметить, что около 9% людей не различают цвета, (обычно красно - зеленые сочетания). Однако, эти люди могут отличать черно-белые оттенки, поэтому проектировщики автоматизированных систем должны проверять, не нарушает ли восприятие пользователей этой категории использование различных цветов в интерфейсах программных продуктов

# Непротиворечивость и стандартизация

- Данные на экране следует располагать таким образом, чтобы пользователь знал, где найти и где ожидать вывода необходимой информации.
- Информация, на которую следует немедленно обратить внимание, должна всегда отображаться в видном месте, чтобы захватить внимание пользователя (например предупреждающие сообщения и сообщения об ошибках);

# Непротиворечивость и стандартизация

- Менее срочная или менее необходимая информация не должна все время находиться перед пользователем, но должна быть доступна, когда понадобится;
- Отчеты и ссылки должны быть сгруппированы

## Тексты и диалоги

- Текст в нижнем регистре читается приблизительно на 13% быстрее чем текст, который напечатан полностью в верхнем регистре;
- Символы верхнего регистра наиболее эффективны для информации, которая должна привлечь внимание.
- НЕ ИСПОЛЬЗУЙТЕ ВЕРХНИЙ РЕГИСТР, ЕСЛИ ВЫ НЕ ХОТИТЕ ВЫДЕЛЯТЬ КАКУЮ-ЛИБО ИНФОРМАЦИЮ;

## Тексты и диалоги

- Выровненный по правому краю текст труднее читать, чем равномерно распределенный текст с невыровненным правым полем;
- Оптимальный интервал между строками равен или немного больше, чем высота СИМВОЛОВ

# Средства управления графического интерфейса пользователя

'Управление' - общий термин для компонентов интерфейса типа слайдеров, кнопок, кадров(фреймов), переключателей и т.д., которые служат, чтобы заместить объекты, являющимися знакомыми пользователям из реального мира.



# Средства управления графического интерфейса пользователя

- **Кнопки** используются, чтобы выбрать опцию или вызвать событие (например, запуск подпрограммы).
- **Переключатели** подобны кнопкам выбора, в которых пользователь выбирает значение из фиксированного списка, но в данном случае, пользователь может выбрать более чем одно значение из списка.

# Средства управления графического интерфейса пользователя

- Элемент **полоса прокрутки** может быть помещен или в горизонтальную или вертикальную линейку на экране.
- **Метки и текстовые блоки** используются для текстовой информации. Различие между ними - **текстовые поля**, позволяют пользователю вводить текстовые данные в поля, в то время как **метки** - поля не редактируемые, используемые только для отображения текста, типа подсказок, команд пользователя и т.д.

# Средства управления графического интерфейса пользователя

- **Списки** - специализированные средства управления, которые отображают раскрывающиеся списки значений (часто с присоединенной полосой прокрутки, чтобы перемещаться вверх или вниз по списку) и позволяют пользователю выбирать значение из списка, или вводить другое значение в присоединенное текстовое поле.
- **Списки – удобный и компактный элемент интерфейса, который занимает минимум места на экране и в то же время несет большую информационную нагрузку.**

# Изображения (Иконки)

Все иконки можно классифицировать согласно тому, насколько точно они отображают несущую функцию:

**Иконки Подобия** - иконки похожи на объекты, которые они отображают (типа ножниц, чтобы отобразить операцию 'вырезки');



**Иконки по образцу** – представляют пример типа объекта (например иконкой, показывающей линию, чтобы представить средство рисования);



# Изображения (Иконки)

**Символические иконки** - используются, чтобы представить действие или состояние в символической форме (например, разорванная линия между двумя компьютерами для того, чтобы показать разорванное сетевое соединение);

**Произвольные иконки** - не несут никакой информации по поводу их представления, поэтому их назначение должно быть описано (например, обратная круговая стрелка, чтобы представить действие 'отмена последней команды')



# Меню

Необходимый элемент автоматизированной системы - меню, позволяющее пользователю выполнять задачи внутри приложения и управлять процессом решения.

**Достоинство** меню в том, что пользователи не должны помнить название элемента или действия, которое они хотят выполнить - они должны только распознать его среди пунктов меню.

Меню может занимать много экранного места, но есть решение для этой проблемы - использование всплывающего или ниспадающего меню.

# Основные принципы создания меню

Обычно команды меню упорядочены некоторым иерархическим способом. Основная проблема состоит в том, чтобы правильно распределить различные пункты меню по различным уровням и правильно их сгруппировать.

Исследования показывают, что имеются четыре варианта для организации меню:

- ✓ Алфавитный
- ✓ Категорийный
- ✓ В соответствии с нормальными соглашениями
- ✓ В соответствии с частотой использования

# Основные принципы создания меню

## Принципы проектирования меню:

- ✓ Структура меню должна соответствовать структуре решаемой системой задачи, организация меню должна отразить наиболее эффективную последовательность шагов, чтобы достичь решения поставленной задачи;
- ✓ Пункты Меню должны быть краткими, грамматически правильными и соответствовать своему заголовку в меню. Порядок пунктов меню выбирается согласно соглашению, частоте использования, порядку использования, в зависимости от потребностей задачи или пользователя.



# Основные принципы создания меню

## Принципы проектирования меню:

- ✓ Выбор пунктов меню должен быть обеспечен несколькими способами – с помощью клавиатуры, с помощью мыши, а также через другие объекты пользовательского интерфейса. Необходимо использовать легко запоминаемые сочетания клавиш для более быстрого доступа к пунктам меню, поскольку это очень экономит время.

# Формы

Формы – основной элемент интерфейса. Назначение форм – удобный ввод и просмотр данных, состояния, сообщений автоматизированной системы.

## Основные принципы проектирования форм:

- ✓ Форма проектируется для более удобного, более понятного и скорейшего достижения решения поставленной задачи. Если форма переносится из бумажной формы, то передвижение по смежным полям не должно вызывать затруднений у пользователя.

# Формы

- ✓ Размещение информационных единиц на пространстве формы должно соответствовать логике ее будущего использования: это зависит от необходимой последовательности доступа к информационным единицам, частотой их использования, а также от относительной важности элементов.

# Формы

- ✓ Важно использовать незаполненное пространство, чтобы создать равновесие и симметрию среди информационных элементов формы, для фиксации внимания пользователя в нужном направлении.
- ✓ Логические группы элементов необходимо отделять пробелами, строками, цветовыми или другими визуальными средствами.
- ✓ Взаимозависимые или связанные элементы должны отображаться в одной форме.

## Дизайн заголовков и полей

- ✓ Для отдельных полей заголовков должен быть выровнен по левому краю; для полей списков, заголовков должен быть выше и левее по отношению к основному полю, числовые поля выравниваются по правому полю.
- ✓ Длинные колоночные поля или длинные столбцы информационных единиц с одиночными полями необходимо объединять в группы пять элементов, разделяемых пустой строкой - это помогает пользователю мысленно обрабатывать информацию по выделенным группам.

# Дизайн заголовков и полей

- ✓ В формах с большим количеством информации необходимо использовать названия разделов, которые однозначно свидетельствуют о характере принадлежащей им информации.
- ✓ Необходимо четко разделить отображение заголовков и непосредственно полей ввода, поскольку такая путаница может вызвать дискомфорт у пользователя.
- ✓ Заголовки должны быть краткими, знакомыми и содержательными.
- ✓ Поля, необязательные для заполнения, либо не имеющие особой важности должны отличаться визуально (цветом или другими эффектами) от полей важных и обязательных для заполнения.

# Форматы ввода

- ✓ Необходимо обеспечить ввод значений по умолчанию во все поля, которые это допускают и такая функция не будет раздражать пользователя. Можно назначить клавиши или коды для ввода часто повторяющихся значений.
- ✓ Входные данные должны быть значимыми и общепринятыми. Данные или коды на международном Английском языке будут проще для узнавания и понимания, чем произвольные коды.
- ✓ Не объедините поля ввода чисел и символов, поскольку числовые и алфавитные клавиши находятся неудобно относительно друг друга на клавиатуре.

# Форматы ввода

- ✓ Моделируйте поля ввода короткими, насколько это возможно.
- ✓ Необходимо исключить частое переключение между верхним и нижним регистрами для ускорения ввода данных.
- ✓ Нельзя требовать от пользователя ввода незначимых цифр (например, вместо 00000010 пользователь должен ввести только 10).



# Организация системы навигации и системы отображения состояний

Навигация обеспечивает пользователю способность перемещаться между различными экранами, информационными единицами и подпрограммами в автоматизированной системе. В полноценной системе пользователь всегда может получить информацию о состоянии системы, процесса выполнения или активной подпрограмме.

# Общие принципы проектирования

Существует ряд навигационных средств и приемов, которые помогают пользователю ориентироваться в системе.

Они включают:

- ✓ использование заголовков страниц для каждого экрана;
- ✓ использование номеров страниц;
- ✓ номеров строк и столбцов;
- ✓ отображение текущего имени файла вверху экрана.

# Общие принципы проектирования

Для интерфейсов **языка команд** очень мало способов обеспечения полноценной навигации.

В интерфейсах **с меню** можно использовать иерархически-структурированное меню. Для выхода из подменю нужно применять не сложные действия.

**Диалоговые интерфейсы** сами по себе защищают пользователя от ошибочных действий.

**Информация Состояния** обычно отображается внизу экрана и содержит в себе данные количества записей, числе обработанных единиц, процессе печати, очереди печати и т.д.

# Проектирование сообщений

**Сообщения необходимы для направления пользователя в нужную сторону, подсказок и предупреждений, для выполнения необходимых действий на пути решения задачи.**

Они также включают подтверждения действий со стороны пользователя и подтверждения, что задачи были выполнены системой успешно либо по каким-то причинам не выполнены. Сообщения могут быть обеспечены в форме диалога, экранных заставок и т. п.

# Проектирование сообщений

Сообщения могут предложить пользователю:

- ✓ Выбрать из предложенных альтернатив некую опцию или набор опций;
- ✓ Ввести некоторую информацию;
- ✓ Выбрать опцию из набора опций, которые могут изменяться в зависимости от текущего контекста;
- ✓ Подтвердите фрагмент введенной информации перед продолжением ввода;

# Проектирование сообщений

Сообщения могут быть помещены в модальные диалоговые окна, которые вынуждают пользователя ответить на вопрос прежде, чем может быть предпринято любое другое действие, потому что все другие средства управления заморожены.

Это может быть полезно, когда система должна вынудить пользователя принять решение перед продолжением работы.

Немодальные диалоговые окна позволяют работать с другими элементами интерфейса, в то время, как само окно может игнорироваться.

# Предотвращение, обнаружение и исправление ошибок

Ошибки могут быть классифицированы как:

- ✓ Ошибки, которые основаны на неправильном понимании действия или порядка действий;
- ✓ Ошибки, которые возникли случайно, непреднамеренно, например опечатка при вводе текста;

# Предотвращение, обнаружение и исправление ошибок

Также ошибки второго вида могут быть разделены еще на шесть видов:

- ✓ Ошибки неточности выбора опции (например, пользователь случайно нажал кнопку "Выход" и программа закрылась);
- ✓ Ошибки управления данными (например, присвоение ошибочного имени файла из-за неточности отображения последнего);
- ✓ Ошибки ассоциативного характера (например, сохранение файла с именем какого-либо человека, так как пользователь думал о нем в момент сохранения);



# Предотвращение, обнаружение и исправление ошибок

- ✓ Ошибка потери активности, когда пользователь забывает необходимую последовательность действий для продолжения работы;
- ✓ Ошибка режима или состояния - когда пользователь думает, что он находится в одном состоянии, но - фактически в другом, например режим вставки взамен режима печати поверх текста в текстовом процессоре;

# Предотвращение, обнаружение и исправление ошибок

Пользователь всегда будет делать ошибки, даже в отличной программной системе, поэтому в разрабатываемой системе всегда должна быть предусмотрена защита от ошибок.

# Предотвращение, обнаружение и исправление ошибок

**Техника защиты от ошибок включает в себя следующие аспекты:**

- ✓ Принудительные действия в системе, которые предотвращают или затрудняют появление ошибок;
- ✓ Обеспечение хороших и информативных сообщений об ошибках;
- ✓ Использование обратимых действий, которые позволяют пользователям исправлять их собственные ошибки;
- ✓ Обеспечение нормальной диагностики системы, в процессе которой пользователю объясняется, в чем суть ошибки и пути ее исправления;

# Обработка ошибок в формах ввода

## Основные принципы:

- ✓ Обеспечить возможность посимвольного редактирования введенных записей для исправления ошибок ввода (опечаток);
- ✓ Если ошибка была обнаружена системой, желательно вернуть курсор в поле с ошибочными данными и каким-либо образом выделить это поле визуально;
- ✓ Обеспечить значимые сообщения об ошибках, использующие стиль языка пользователя и соответствующую терминологию;
- ✓ Обеспечьте сообщения об ошибках, которые объясняют и предлагают пути ее устранения.

Спасибо за внимание!