



# Проектирование человеко- машинных интерфейсов

Лекция 2.

Пользовательский интерфейс.

Часть 2.



## Пользовательский интерфейс (def)

**Интерфейс** - система правил и средств, регламентирующая и обеспечивающая взаимодействие нескольких процессов или объектов.

**Пользовательский интерфейс (ПИ)** - система правил и средств, регламентирующая и обеспечивающая взаимодействие программы с пользователем.



## Эргономика и usability (def)

- **Эргономика** (от греч. ergon работа и nomos закон) — научно-прикладная дисциплина, занимающаяся изучением и созданием эффективных систем, управляемых человеком.
- **Юзабилити (usability)** — степень, в которой продукт может быть использован определенными пользователями для достижения поставленных целей эффективно, экономично и с удовольствием.

(Пункт 3.1 стандарта ISO 9241-11)



## На мировом рынке ПО

- **ПИ** составляет от 47 до 60 процентов кода программы;
- на разработку **ПИ** уходит как минимум 29 процентов проектного бюджета и в среднем 40 процентов всех усилий разработчиков по созданию системы.



## **Типы программ (5 миров)**

- 1. Встроенное программное обеспечение**
- 2. Внутреннее корпоративное ПО**
- 3. Массовое (коробочное) ПО**
- 4. Игры**
- 5. Одноразовые программы**



## **Встроенное ПО**

- **Прочно связано с железом**
- **Редко обновляется – «второго шанса не будет»**
- **Требует оптимизации кода**
- **Средства ввода-вывода ограничены**
- **Нет общепризнанного стандарта элементов интерфейса**



## **Внутреннее корпоративное ПО**

- **Ограниченный круг пользователей**
- **Высокие требования к срокам разработки**
- **Однородное программное окружение**
- **Вне зависимости от качества ПИ, пользователям придется к нему привыкнуть**



## **Массовое (коробочное) ПО**

- **Очень большое количество пользователей**
- **Высокая конкуренция**
- **Множество вариаций программного окружения**
- **Интерфейс должен быть максимально простым**
- **Много версий («в следующей версии сделаем лучше»)**





# Игры

- **Только одна версия! Второго шанса не будет!**
- **Рынок ориентирован на хиты**
- **Дизайн имеет решающее значение**
- **Массовая аудитория очень похожих пользователей**
- **Можно ограничивать требования к аппаратному и программному окружению**



# Одноразовое ПО

- Создается временно для решения конкретной небольшой задачи
- Обычно используется всего один раз
- Интерфейс не имеет значения

## Примеры:

- Скрипты
- Макросы



## Преимущества хорошего ПИ

1. Повышение конкурентоспособности.
2. Снижение стоимости разработки.
3. Увеличение аудитории продукта.
4. Уменьшение затрат на обучение и поддержку пользователей.
5. Уменьшение потерь продуктивности работников при внедрении системы и более быстрое восстановление утраченной продуктивности.
6. Доступность функциональности системы для максимального количества пользователей.
7. Снижение риска ошибок.



## Типы пользовательского интерфейса

1. Пакетный интерфейс.
2. Интерфейс командной строки.
3. Графический
4. Диалоговый
5. Текстовый естественно-языковый
6. Речевой
7. Биометрический (Мимический)
8. Нейрокибернетический



## Командный язык (набор инструкций)

### **Достоинства :**

- Быстрый для экспертов
- Гибкий
- Эффективен

### **Недостатки:**

- Требуется помнить инструкции
- Может быть непонятен
- Не подходит для новичков



## Преимущества графического интерфейса в сравнении с текстовым

### **Графические интерфейсы обладают рядом преимуществ:**

- Их относительно просто изучить и использовать. Пользователи, не имеющие опыта работы с компьютером, могут легко и быстро научиться работать с графическим интерфейсом.
- Каждая программа выполняется в своем окне (экране). Можно переключаться из одной программы в другую, не теряя при этом данные, полученные в ходе выполнения программ.
- Режим полноэкрannого отображения окон дает возможность прямого доступа к любому месту экрана.



## Диалог Вопрос / Ответ

### **Достоинства:**

- Легок для новичков
- Минимальная информационная нагрузка
- Несложный ручной ввод

### **Недостатки:**

- Пользователь теряет управление
- Медленный
- Может быть утомителен



## Выбор из меню

### **Достоинства:**

- Легок для новичков
- Минимальная информационная нагрузка
- Компромисс между командным языком и диалогом вопрос/ответ

### **Недостатки:**

- Медленный для экспертов
- Слишком много альтернатив или уровней могут запутать





## Естественный язык / виртуальная реальность

### **Достоинства:**

- Очень богат
- Использует дополнительный СМЫСЛ - КОНТЕКСТ

### **Недостатки:**

- Нагружает процессор
- Труден для реализации



## Основная аксиома ПИ

**Хороший дизайн  
пользовательского  
интерфейса подразумевает,  
что программа соответствует  
ожиданиям пользователей о  
том, как она должна себя  
вести.**

**(Все остальное – следствия.)**



# Понятность интерфейса

- **Ментальная модель** – представление пользователя о том, как работает программа.
- **Метафора** – пользователь использует модель аналогичной привычной ему деятельности.
- **Приглашение** (аффорданс) – объект подсказывает способ своего использования своим видом.
- **Стандарт** – постоянство в дизайне увеличивает понятность.



# Ментальная модель

- **Модель пользователя** – представление о том, для чего и как программа будет работать.
- **Модель программы** – как на самом деле программа работает.

**Модель программы должна  
соответствовать модели  
пользователя.**



# Метафора интерфейса

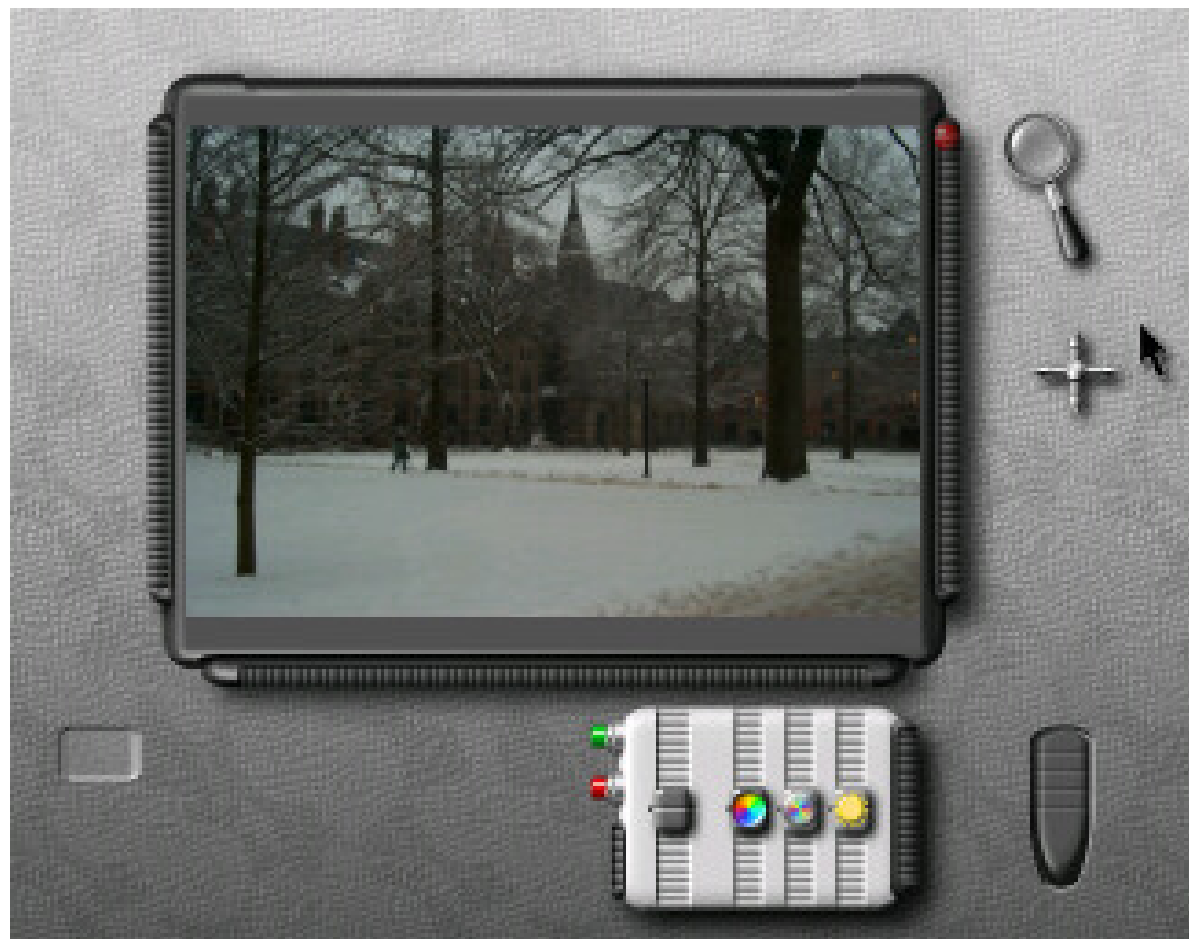
Иногда у пользователей просто **нет** конкретного представления о том, как работает программа и для чего она предназначена.

В таком случае вам придется найти способ подсказать им, как функционирует ваша программа. В графических интерфейсах используется метод **метафор**.

Самая известная метафора GUI –  
«рабочий стол»



# Пример метафоры





# Приглашение

**Объект своим внешним видом  
намекает пользователю на свою  
функцию.**

**Например:**

- **Кнопки - выпуклые, приглашают нажать.**
- **Стрелки.**
- **Пиктограммы.**



# Пример приглашения



А это – не кнопка!

Кнопка

Тоже кнопка

Как ни удивительно,  
НО ЭТО ТОЖЕ КНОПКИ.





# Стандарт

## **Постоянство в дизайне – фундаментальный принцип хорошего UI дизайна.**

**Дизайн элементов управления, выдержанный в едином ключе для различных программ, помогает пользователю обучиться работать с новой программой. До появления графических интерфейсов, каждому разработчику новой программы приходилось придумывать сами основы ее пользовательского интерфейса.**



## Что нужно знать о пользователях

- Пользователи ничего не читают.
- Пользователи плохо видят.
- Пользователи не умеют работать с мышью.
- Пользователи ничего не помнят.



## Пользователи не читают

1. У пользователей нет документации, а если бы она и была, они бы ее не читали.
2. На самом деле, пользователи не умеют читать, а если бы и умели, то не стали бы.

Чем больше слов в диалоговом окне, тем меньшее число пользователей их читают.



# Много слов в диалоговом окне





## Пользователи плохо видят

323 Fillmore Street

323 Fillmore Street

**Удобство использования важнее  
«стильности» и элегантности.**



## Пользователи не умеют работать с мышью

1. Иногда приходится пользоваться не самыми оптимальными манипуляторами.
2. Иногда приходится пользоваться мышью не в самых благоприятных условиях.
3. Иногда человек, сидящий за компьютером, - новичок, не имеющий опыта работы с мышью.
4. Некоторые люди никогда не смогут развить навыки работы с мышью – по медицинским показаниям.
5. Некоторые люди считают, что постоянное применение мыши замедляет работу.



## Пример: выпадающий список



- Выпадающий список нужно сделать таким длинным, чтобы он вмещал все элементы.
- Автоматическая прокрутка.
- Чтобы открыть список, можно щелкнуть в любом месте списка, а не только на стрелочке.



## Пользователи ничего не помнят

**Пользователь не должен  
запоминать то, что может  
запомнить компьютер.**

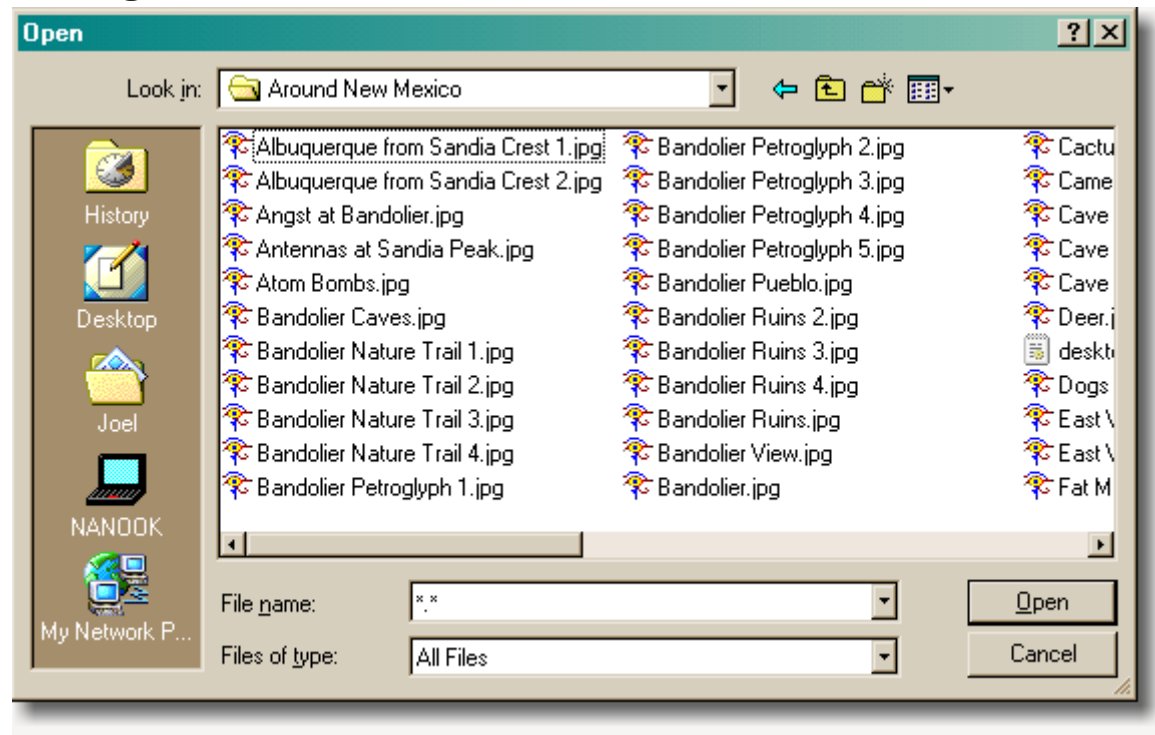
- Меню команд.
- Списки значений.
- «Интеллектуальное завершение ввода» (auto completion).





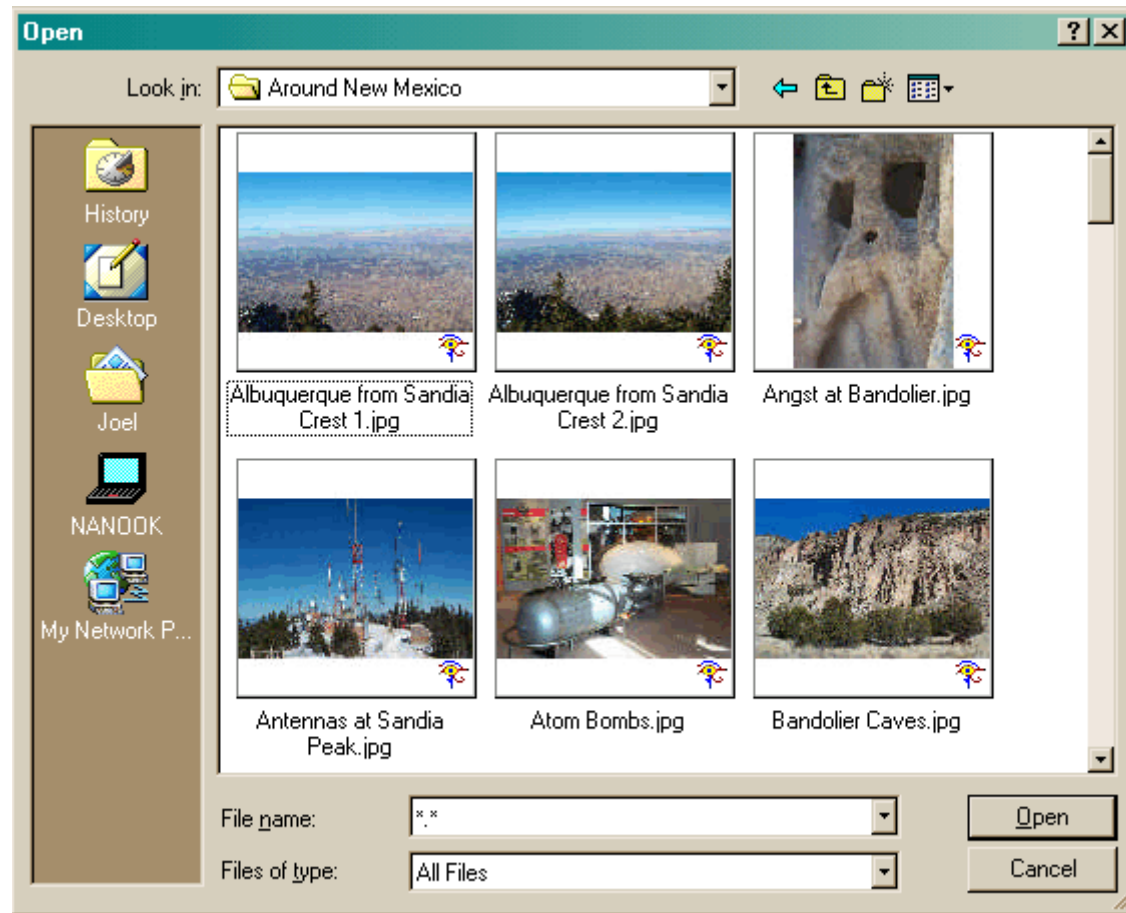
# Пример: выбор файла с картинкой

## Неудобный вариант:





# Пример: выбор файла с картинкой





## Пример: интеллектуальное завершение ввода

	A	B	C	D	E
1	Name	Age	Sex		
2	Joel	23	Male		
3	Jenine	29	Female		
4	Micah	39	Male		
5					
6					
7					
8					



# Как создать интерфейс

## Планирование Деятельности.

**Нужно:** предугадать те виды деятельности, которые будет осуществлять пользователь с вашей программой, и сконцентрироваться на том, чтобы сделать выполнение деятельности простым и удобным.



## **Воображаемые пользователи**

**Перед тем как разрабатывать дизайн интерфейса, нужно придумать и описать воображаемых пользователей (персонажей).**

**Когда вы придумали своих пользователей, понимание того, является ли ваш дизайн соответствующим, приходит почти автоматически.**



# Итог: алгоритм разработки GUI

1. Придумать воображаемых пользователей (персонажей).
2. Продумать виды деятельности пользователей.
3. Узнать модель пользователя – как он будет выполнять деятельность, основываясь на своем опыте.
4. Сделать первый набросок дизайна.
5. Изменять дизайн, все больше и больше делая его простым в использовании, до тех пор, пока продукт не окажется в рамках способностей воображаемых пользователей.
6. Наблюдать за тем, как реальные пользователи работают с вашим продуктом. Отметить области, в которых они испытывают трудности. Эти области, скорее всего, и демонстрируют несоответствия модели программы модели пользователя.

# Процесс проектирования интерфейса

## ■ Анализ

Анализ бизнес-процессов и деятельности пользователей

## Проектирование

- Формализация информации в виде диаграмм бизнес-процессов и сценариев

## ■ Выработка концепции

- Прототипирование интерфейса

## Реализация

- Уточнение прототипов

- Тестирование с пользователями

## ■ Внедрение

Получение обратной связи от пользователей и ее анализ



## Нормативные документы

- ISO 9241** Эргономические требования к офисной работе с визуальными терминалами (VDTs):  
Часть 11 - Руководство по юзабилити
- ISO 9126** Качество программного продукта: Характеристики и подхарактеристики качества. Модель качества. Показатели Качества в использовании.
- CIF** Формат описания юзабилити характеристик продукта и результатов юзабилити тестов.