

Лекция 11. Компоненты интерфейса и их стандартизация

Цель лекции: в данной лекции рассматриваются все вопросы, касающиеся компонентов интерфейса, а также их стандартизация. Описаны все этапы проектирования интерфейсов. Охарактеризованы все элементы интерфейсов, а также рассмотрены все предпосылки создания единой среды разработки приложений

Вопросы для рассмотрения: Вопросы создания единой разработки приложений. Интерфейс WIMP: окна, значки/иконки, меню, указатели. Другие компоненты: списки, элементы управления, компоненты отображения, компоненты ввода текста, контейнеры инструментов. Описание всех этапов проектирования

Основные термины: юзабилити, обучаемость, эффективность, запоминаемость, атрибуты качества

Введение

Стандартизация пользовательского интерфейса

Ведущие специалисты в области человеко-машинных компьютерных систем уже к середине 70-х годов осознали необходимость формирования единых подходов к реализации пользовательского интерфейса.

Весьма длительное время основной формой общения пользователя с компьютером оставался диалог в форме «вопрос-ответ». Но, возможно, именно потому, что компьютер выступал в роли собеседника, очень быстро возникла необходимость исследования психологических аспектов общения человека с компьютером. В настоящее время уже ни одна серьезная публикация, посвященная пользовательскому интерфейсу, не обходится без ссылок на результаты, полученные в таких областях знаний, как психология, эргономика, математическая лингвистика, кибернетика и т.д. В качестве иллюстрации того, насколько серьезно относятся «законодатели моды» в области компьютерных технологий к проблемам интерфейса, можно отметить следующий факт. Американский Национальный институт стандартов (ANSI) имеет по данному направлению специальную консультативную группу — Комитет по стандартам интерфейса «человек-компьютер» (The Human-Computer Interface Standard Committee). Существуют подобные организации не только в США, но и в

других странах; более того, имеются также международные исследовательские группы, работающие в этом направлении, например, Международный консультативный комитет по телеграфии и телефонии (International Telegraph and Telephone Consultation Committee), изучающий особенности интерактивных элементов интерфейса.

Многими из этих организаций или рабочих групп в свое время были подготовлены проекты документов по стандартизации пользовательских интерфейсов, содержащие принципы их проектирования и реализации. Так, в 1986 году было опубликовано «Руководство по разработке программного пользовательского интерфейса» [1], содержащее 944 принципа, касающихся ввода и отображения данных, поддержки пользователя, защиты данных и т.д. Однако ни один из этих проектов не получил статуса официального документа, поскольку все они имели общий недостаток (тот же, что и первые исследования в этой области): в них не учитывались технологические возможности инструментальных средств, имевшихся в распоряжении разработчиков программного обеспечения.

В 1987 г. корпорация IBM объявила о намерении создать единую среду разработки приложений (Systems Application Architecture — SAA).

Данный проект предусматривал не только разработку единых принципов создания приложений, но и «материализацию» этих принципов на основе соответствующей технологической базы.

Целями проекта являются:

Повышение производительности труда программистов и конечных пользователей.

Облегчение эксплуатации и сопровождения программного обеспечения.

Повышение эффективности распределенной обработки информации.

Увеличение отдачи инвестиций в разработку информационных систем. ***Проект***

SAA содержит 4 компонента:

Соглашения по интерфейсу пользователя (Common User Access — CUA);

Соглашения по программному интерфейсу (Common Programming Interface — CPI);

Соглашения по разработке приложений (Common Applications — CA);

Соглашения по коммуникациям (Common Communications Support — CCS).

В качестве технологической базы для реализации соглашений по пользовательскому интерфейсу было предложено конкретное инструментальное средство — Programming Toolkit для операционной системы OS/2.

Исследованиями и практической реализацией графических интерфейсов в то время уже занимались такие фирмы как Xerox, Apple, Digital Research и Microsoft. В результате их деятельности были определены основные концепции построения графических пользовательских интерфейсов:

- использование единой рабочей среды пользователя в виде так называемого Рабочего стола;

объектно-ориентированный подход к описанию заданий пользователей;

использование графических окон в качестве основной формы отображения данных;

- применение средств неклавиатурного ввода, основанного на выборе и указании с помощью манипулятора «мышь».

Стандартизованный интерфейс (именно стандартизованный, а не стандартный) должен отвечать двум основным требованиям:

обладать перечисленными в предыдущем разделе свойствами (естественности, согласованности и т.д.);

быть узнаваемым (или предсказуемым, что в данном случае одно и то же).

При весьма заметном различии в деталях все известные стандарты используют в качестве исходного положения понятие жизненного цикла программного продукта (или системы).

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

Практически все стандарты 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.

Необходимо подчеркнуть, что положения стандартов остаются справедливыми вне зависимости от предназначения, уровня сложности и состава коллектива разработчиков программного продукта, то есть даже в тех случаях, когда речь идет о небольшой программной утилите, а коллектив разработчиков состоит из одного человека.

ПРОЕКТИРОВАНИЕ

Начальный этап в разработке программного продукта (приложения) является наиболее критичным, поскольку на этой фазе определяется общая концепция создаваемого продукта. Если проект в своей основе неудовлетворителен, впоследствии трудно будет что-либо кардинально изменить в лучшую сторону.

Эта часть процесса разработки включает не только определение цели и характеристик приложения, но и понимание того, кто является его потенциальными пользователями — их задачи, намерения, цели. Это предполагает учет таких показателей, как, например, возраст пользователей, их пол, экспертные знания, уровень опыта, физические ограничения, специальные потребности и т.д. Продумайте структуру приложения и метафоры, которые могут быть применены при ее реализации. Решению указанной проблемы способствует наблюдение за работой пользователей при выполнении ими задач в данной предметной области.

Представьте проект разработки в письменной форме; это не только обеспечивает важную контрольную точку в процессе создания приложения и средство взаимодействия с пользователями, но часто помогает сделать проект более конкретным и выявить нерешенные вопросы.

ПРОТОТИПИРОВАНИЕ

После того, как определены основные концепции проекта, разрабатывается прототип создаваемого приложения, отражающий некоторые основные аспекты его функционирования. В зависимости от уровня вашей подготовки и сложности приложения, его прототип может быть представлен либо в виде иллюстраций интерфейса (внешне они могут напоминать картинки из комиксов), либо в виде специальных схем (в частности, в виде сетей Петри). На последующей стадии может быть создана модель (или макет) проектируемого приложения — действующее программное обеспечение, использующее либо специальные средства макетирования (например, какую-либо CASE-систему), либо обычные инструментальные средства программирования.

Прототип предоставляет хорошую возможность для обсуждения создаваемого приложения как внутри группы разработчиков, так и с потенциальными пользователями. Он может помочь вам определить характер потока заданий и лучше представить себе то, чем вы, собственно, занимаетесь. Это особенно полезно в начале процесса разработки.

Форма представления прототипа зависит от цели разработки. Действующие прототипы обычно наиболее полно позволяют оценить качество механизма взаимодействия пользователя с разрабатываемым приложением, т.е. качество интерфейса.

ИСПЫТАНИЕ ПРОГРАММНОГО ПРОДУКТА

UCD-технология предполагает достаточно активное привлечение пользователя к процессу разработки. Совместное с потенциальным пользователем испытание создаваемого приложения обеспечивает получение

весьма ценной дополнительной информации и является, как правило, залогом успешной последующей реализации продукта. Необходимо подчеркнуть, что испытание программного продукта принципиально отличается от его отладки.

Первое и наиболее важное отличие обусловлено различными целями этих двух процессов: отладка имеет целью выявление дефектов (ошибок) программирования, в то время как в ходе испытаний вы оцениваете, насколько полно разработанное приложения (в частности, его интерфейс) отвечает потребностям и ожиданиям пользователя. Второе принципиальное отличие состоит в том, что отладку выполняет непосредственно его разработчик, а основным действующим лицом при проведении испытаний является потенциальный пользователь (заказчик). Благодаря этому в ходе испытаний могут быть выявлены не только технические погрешности, но и концептуальные проблемы в предлагаемом продукте.

Кроме того, 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.

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.

ОЦЕНКА ПОТРЕБИТЕЛЬСКИХ СВОЙСТВ ПРИЛОЖЕНИЯ В ПРОЦЕССЕ РАЗРАБОТКИ

Как было указано выше, основная цель испытаний — определить, насколько полно разработанное приложение (в первую очередь, его интерфейс) отвечает потребностям и ожиданиям пользователя. В связи с этим основным направлением испытаний приложения является оценка его «потребительских свойств» (Usability). Такая оценка должна проводиться, начиная с самых ранних этапов разработки. Основой для проведения оценки должны служить данные о том, как пользователи обычно выполняют ту работу, которую призвано автоматизировать создаваемое приложение. По мере того, как разработка продвигается, оценка потребительских свойств приложения должна постоянно уточняться. Чем чаще и корректнее будет проводиться оценка, тем выше будет качество разработки.

Нет простого уравнения, позволяющего определить, каким образом вносимые изменения повлияют на первоначальный проект приложения. Однако важную роль в такой оценке могут сыграть следующие соображения:

Каждая дополнительная характеристика потенциально влияет на поведение, сложность, устойчивость, эксплуатацию и издержки по сопровождению создаваемого ПО.

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.

Простота пользовательского интерфейса — это далеко не то же самое, что его упрощенчество. Чтобы сделать нечто простым в использовании, часто требуется приложить много сил и создать весьма сложное изделие с точки зрения его внутренней организации.

Доработки программного кода с целью расширения функциональных возможностей ПО совсем не обязательно будут иметь пропорциональный положительный эффект с точки зрения интерфейса пользователя. Например, если основная задача пользователя состоит в выборе единственного объекта, то, предоставляя ему возможность одновременного выбора нескольких объектов, вы только усложняете ему работу.

WIMP («windows, icons, menus, pointers» — [окна, значки, меню, указатели](#)) — в [человеко-компьютерном взаимодействии](#) означает взаимодействие с компьютером на базе этих элементов. Оно было придумано [Мерзугой Уильбертсом](#) (англ.) в 1980 году.^[1] Хотя его популярность постепенно падает, это слово часто используется в качестве приближённого синонима [«графического интерфейса пользователя»](#). WIMP был разработан в корпорации [Xerox PARC](#) и «популяризирован компьютером Macintosh в 1984 году», в котором были добавлены понятие «строка меню» и концепция расширенного управления окном.^[2]

Этот стиль взаимодействия использует физическое устройство для управления положением курсора и предоставляет пользователю информацию, организованную в виде окон и иконок. Доступные команды собраны в меню и управляются [курсором мыши](#). Это сделано для того, чтобы уменьшить когнитивную нагрузку на пользователя (не нужно помнить все возможности), что сокращает время обучения.

Другое очевидное достоинство этого стиля — это [простота его использования](#) для людей, далёких от компьютеров, как для новичков, так и для опытных пользователей. Кроме того, эта технология может быть легко

перенесена из одного приложения в другое, с учётом высокой согласованности между интерфейсами.

Графический интерфейс имеет два основных элемента: окна и меню. А эти элементы имеют свои различные типы:

Окна	Меню
Рабочий стол; окна папок; окна приложений; окна документов; диалоговые окна.	Главное меню; Контекстное меню.

Рассмотрим каждый элемент графического интерфейса.

1) Рабочий стол.

Рабочий стол — это главная область экрана, которая появляется после включения компьютера и вход в операционную систему.

На рабочем столе можно размещать различные объекты (папки, файлы) и выстраивать их в удобном порядке. Так же на рабочем столе появляются запущенные программы и открытые папки.



Рис. 1. Рабочий стол

Значки — маленькие рисунки, которые обозначают программы, файлы, папки и другие объекты.

Для упрощения доступа с рабочего создают ярлыки файлов и программ.

Ярлык — значок, представляющий ссылку на объект, который может быть расположен в любой папке.

Ярлык можно узнать по стрелке на его значке. Двойной щелчок на ярлыке открывается сам объект. При удалении ярлыка удаляется только сам ярлык, а исходный объект остается. В свойствах ярлыка указывается размещение объекта, на который он указывает.



Рис. 2. Ярлык и размещение объекта, на который он указывает

Панель задач — длинная горизонтальная полоса, расположенная в нижней части экрана.

Панель задач состоит из основных частей:

кнопка «Пуск», открывающая Главное меню;

панель быстрого запуска, позволяющая запустить программу одним нажатием кнопки мыши;

средняя часть, которая отображает открытые программы и документы;

область уведомлений, в которой находятся часы и значки (маленькие картинки).

2) Окна папок.

Окна папок — область экрана, ограниченная прямоугольной рамкой, в которой отображается содержимое папки, работающая программа или документ.

Основные элементы окон папок:

строка заголовка: строка под верхней границей окна, содержащая системный значок, заголовок окна, а также кнопки управления состоянием окна, которые позволяют развернуть, свернуть или закрыть окно;

строка меню: располагается под строкой заголовка и представляет собой перечень тематически сгруппированных команд;

панель инструментов: располагается под строкой меню и представляет собой набор кнопок, которые обеспечивают быстрый доступ к наиболее важным и часто используемым пунктам меню окна;

адресная строка: располагается под панелью инструментов и указывает путь доступа к текущей папке, что удобно для ориентации в файловой структуре;

рабочая область: внутренняя часть окна, в которой производится работа с дисками, файлами и документами;

полосы прокрутки: появляются, если содержимое окна имеет больший размер, чем рабочая область окна, позволяют перемещать содержимое окна по вертикали или горизонтали;

границы окна: рамка, ограничивающая окно с четырех сторон. Размеры окна можно изменять, ухватив и перемещая границу мышью;

строка состояния: строка, в которой выводится дополнительная информация.

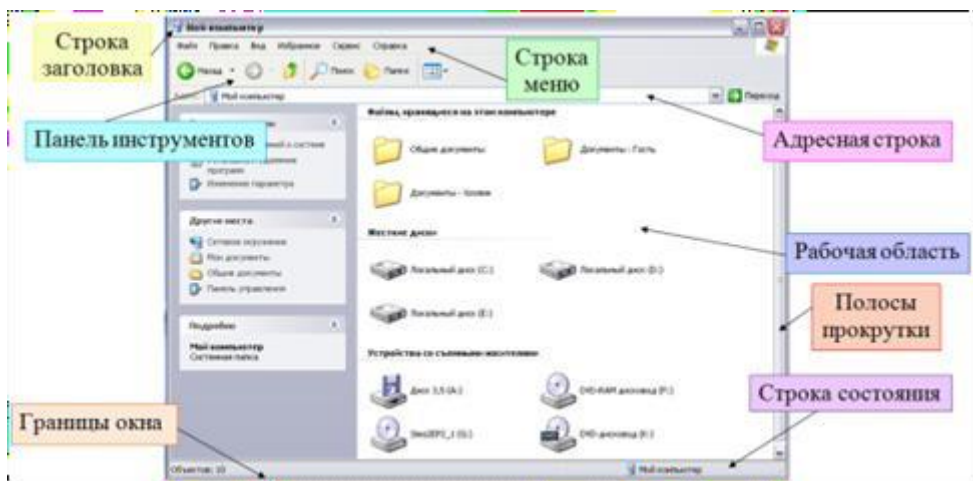


Рис. 3. Вид окна папки и его элементы

3) Окна приложений.

Окно приложения — область экрана, в которой представлено запущенное на выполнение приложение.

Каждое приложение имеет главное окно. В ходе работы с приложением могут открываться дополнительные подчиненные окна.

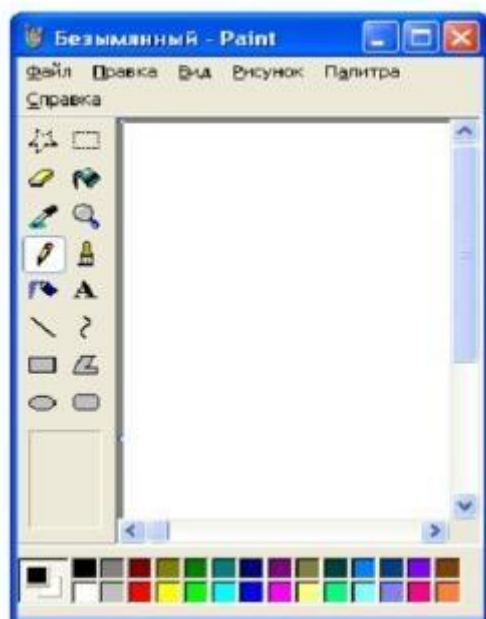


Рис. 4. Окно приложения — графического редактора Paint

4) Окна документов.

Окно документа не может существовать самостоятельно, оно управляется каким-либо приложением. Такие окна размещаются только внутри главного окна приложения и исчезают при закрытии главного окна.

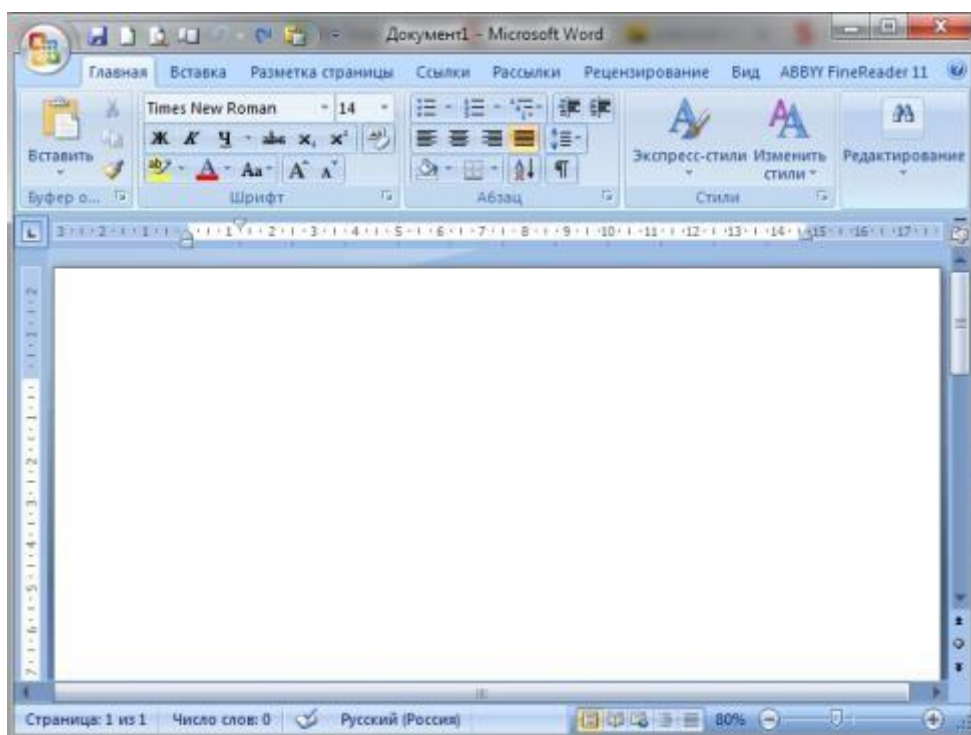


Рис. 5. Окно документа, управляемое приложением MS Word

5) Диалоговые окна.

Диалоговое окно — специальный элемент интерфейса, предназначенный для вывода информации и (или) получения ответа от пользователя. Осуществляет двусторонний «диалог» между пользователем и компьютером.

На диалоговой панели размещаются разнообразные элементы управления:

поля ввода — в него вводится требуемая информация с помощью клавиатуры;

список — представляет собой перечень значений, из которого следует выбрать одно нужное;

раскрывающийся список — представляет собой набор значений и выглядит, как текстовое поле, снабженное кнопкой с направленной вниз стрелкой;

переключатель — служит для выбора одного из взаимоисключающих вариантов, варианты выбора представлены в форме маленьких белых кружков. Выбранный вариант обозначается кружком с точкой внутри;

флажок — обеспечивает присваивание, какому-либо параметру определенного значения; имеет форму квадратика, когда флажок установлен, в нем присутствует «галочка»;

командные кнопки — обеспечивает выполнение того или иного действия, а надпись на кнопке поясняет ее назначение;

счетчики — представляет собой пару стрелок, которые позволяют увеличивать или уменьшать значение в связанном с ними поле;

ползунки — позволяет плавно изменять значение какого-либо параметра.

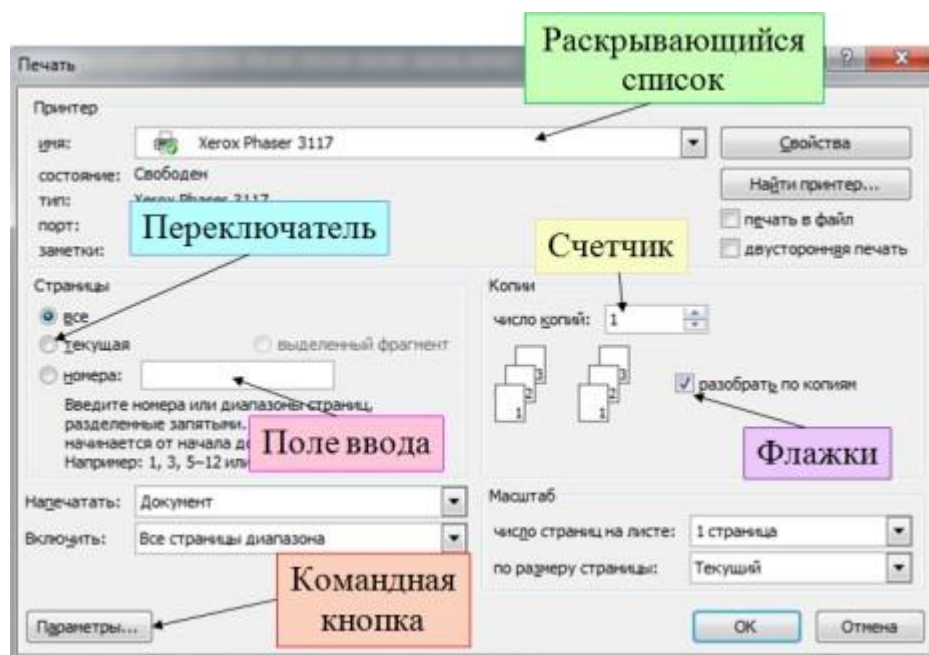


Рис. 6. Пример диалогового окна

Диалоговые окна могут содержать несколько вкладок, переключение между которыми осуществляется щелчком мыши на их названиях.

6) Главное меню.

Главное меню (меню «Пуск») — является основным средством доступа к программам, папкам и параметрам компьютера.

Меню «Пуск» обеспечивает доступ практически ко всем ресурсам системы и содержит команды запуска приложений, настройки системы, поиска файлов и документов, доступа к справочной системе и др.



Рис. 7. Главное меню

7) Контекстное меню.

Контекстное меню — меню, содержимое которого зависит от вида и состояния объекта, с которым оно связано.

Оно отображается при щелчке правой кнопкой мыши по выбранному объекту.

Контрольные вопросы

1. Назовите признаки хорошо спроектированного пользовательского интерфейса.
2. Из каких основных этапов состоит проектирование интерфейса? Перечислите их.
3. С помощью каких средств реализуется пользовательский интерфейс?
4. Назовите основные функции, решаемые с помощью пользовательского интерфейса.
5. С помощью каких систем интерфейсов реализуется взаимодействие человека с компьютером?
6. Что входит в структуру пользовательского интерфейса?
7. Что такое «дружественный интерфейс» и цели его проектирования?
8. Требования к информации в пользовательских интерфейсах.
9. Что такое «функциональность пользовательского интерфейса»?
10. Чем отличается взаимодействие от действия и взаимовоздействия?
11. Что такое «программный интерфейс»? Назовите его основные свойства.
12. WIMP-интерфейс. Опишите полную структуру данного интерфейса. Дайте оценку.

Литература:

1. Купер А. т.б. Об интерфейсе: основы проектирования взаимодействия. 4-е изд. – СПб: Питер, 2018. – 720 б.
2. Норман Д. Дизайн привычных вещей. – М: Манн, Иванов и Фербер, 2019. – 384 б.
3. The Encyclopedia of Human-Computer Interaction, 2nd Ed. [электрондық басылым] – URL: <https://www.interaction-design.org/literature/book/the-encyclopedia-of-human-computer-interaction-2nd-ed> (соңғы алышқан 8/1/2020)

4. Sharp H. Interaction Design: Beyond Human-Computer Interaction. 5th Ed. – Wiley, 2019. – 656 p.
5. Shneiderman B. et.al. Designing the user interface: strategies for effective human-computer interaction. 6th Ed. – Pearson, 2016. – 616 p.
6. Тидвелл Д. т.б. Разработка пользовательских интерфейсов. 2-е изд. – М: Питер, 2011 – 480 б.
7. Круг С. т.б. Веб-дизайн или «не заставляйте меня думать». – СПб.: Символ Плюс, 2008. – 224 .
8. Нильсен Я. т.б. Веб дизайн. СПб: Символ Плюс, 2006. – 512 б.
9. Уильямс Р. т.б. Не дизайнерская книга о дизайне. – СПб: Весь, 2004. – 128 б.
10. <http://appcamp.io/> – Онлайн-курс который дает начальное понимание разработки на HTML и мобильных платформах.
11. <http://phonegap.com/book/> – Список книг по разработке HTML и мобильных приложений с помощью фреймворкаPhoneGap.
12. <http://creator.ionic.io/> – HTML фреймворк пользовательского интерфейса для мобильных приложений.