# Лабораторная работа №4

# Создание прототипа интерфейса и его тестирование

# 1 Цель лабораторной работы

# 1.1. Приобретение умений по формированию электронного прототипа – демонстрационного ролика интерфейса.

# 1.2. Приобретение практических навыков по созданию тестовых заданий.

# 1.3. Закрепление теоретических знаний и приобретение практических навыков по проведению тестирования интерфейса.

# 2 Задание на лабораторную работу

# 2.1) Собрать полную функциональную схему приложения в виде диаграммы последовательности действий.

# 2.2) Выполнить проверку соответствия структуры полной схемы и последовательностей действий, то есть проверить, нет ли лишних или тупиковых состояний.

# 2.3) При выявлении несоответствий внести коррективы в содержание экранных форм и/или схему навигации по приложению.

# 2.4) Повторить предыдущие шаги для откорректированной схемы. Если используется несколько экранных форм, показать их все. Создать наброски в виде диаграмм состояний, в которых могут находиться формы, со стрелками от состояния 1 к состоянию 2, и т.д.

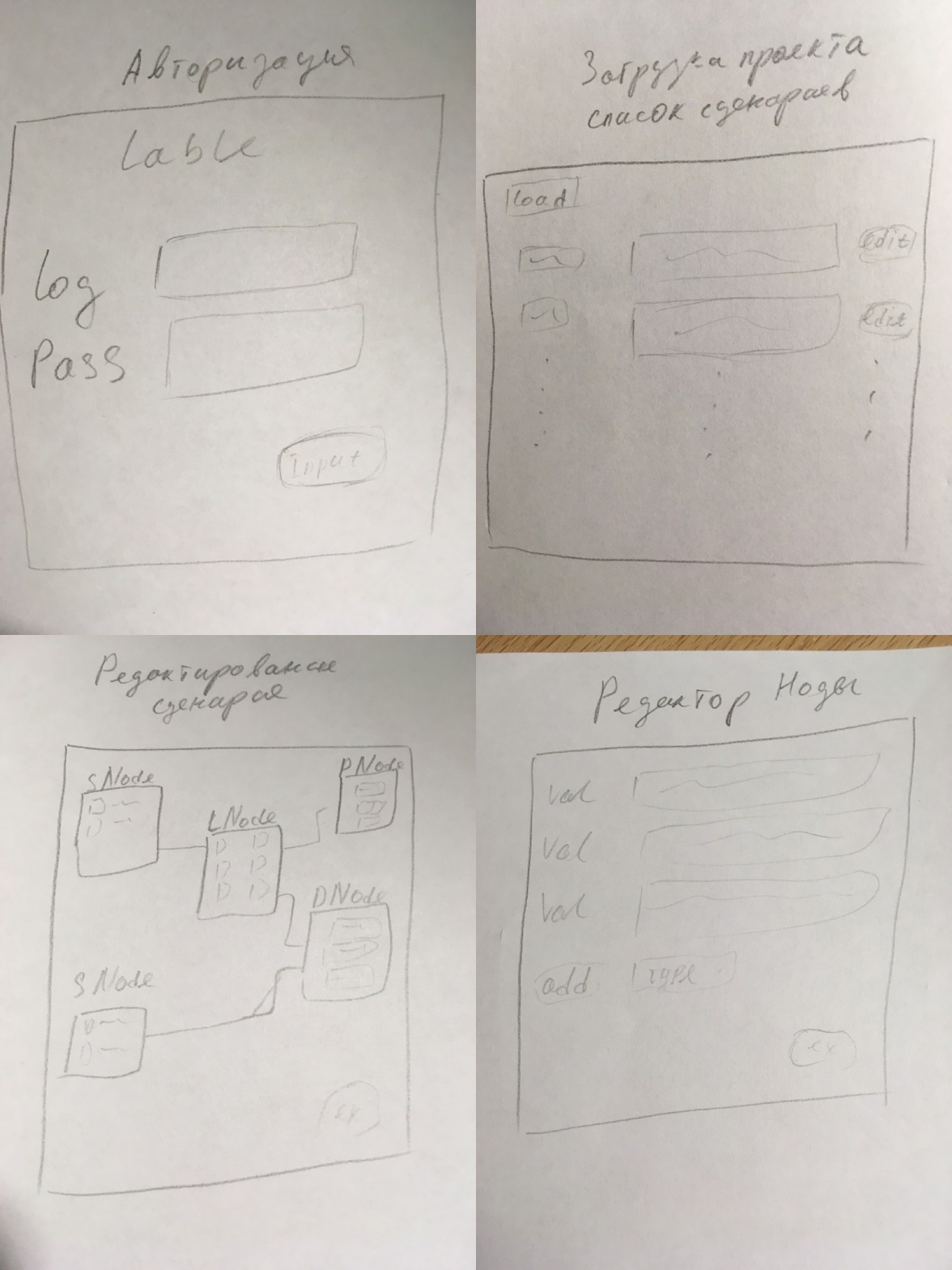
# 2.5) Сформировать слайды со скриншотами формы или набросками для создания демонстрационного ролика. Каждый слайд соответствует определенному состоянию отдельной экранной формы.

# 2.6) Согласно полной схеме приложения собрать демонстрационной ролик. Для организации переходов между слайдами использовать гиперссылки (на кнопках смены состояния).

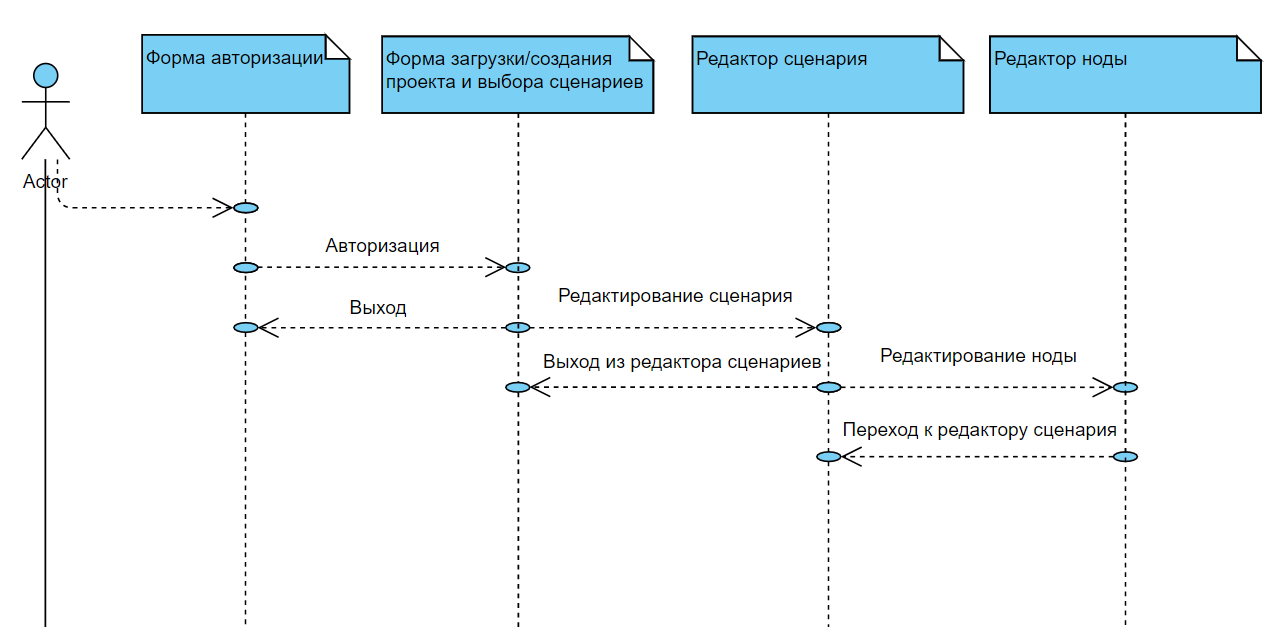
# 2.7) Для оценки работоспособности системы исходя из пользовательских сценариев подготовить тестовые задания, которые определят, правильно ли работает система. Зафиксировать количественные оценки качества разрабатываемого интерфейса.

# 2.8) Составить отчет с пояснениями и выводом.

Прототип



Диаграммы последовательности действий



Презентация в файле ЛР.pptx

Лабораторная работа № 5

Разработка пользовательского интерфейса: этапы предварительного и высокоуровневого проектирования

1 Цели работы

Целями лабораторной работы являются:

1.1 Закрепить теоретические знания по разработке пользовательского интерфейса.

1.2 Получить практические навыки по проведению этапов предварительного и высокоуровневого проектирования интерфейса пользователя.

Задание

2.1)Разработать главное меню своего приложения, составить его схему.

2.2) Выполнить этапы предварительного и высокоуровневого проектирования при разработке пользовательского интерфейса приложения для предметной области, соответствующей варианту задания. Создать программу.

2.3)Предусмотреть использование профилей потенциальных пользователей программного обеспечения, т.е. различные состояния программы для различных пользователей.

2.4) Составить отчет с пояснениями и выводом.

Предметная область представлена информационной системой, отображающую деятельность построения сценариев, проекты которой привязывается к разрабатываемому продукту.

Потенциальными пользователями приложения являются, сценаристы.

Примерные профили некоторых могут выглядеть следующим образом

|  |  |
| --- | --- |
| Пользователи | Сценарист |
| Социальные характеристики | Мужчины, женщины  Взрослые  Русскоязычные  Средний уровень владения компьютером |
| Мотивационно целевая среда | Прямая производственная необходимость, удобство  Мотивация к обучению высокая |
| Навыки и умения | Должны иметь значительный тренинг работы с программой |
| Требования к ПО ИС | Создание логических блоков ветвления сюжета  Построение диалогов |
| Задачи пользователя | Проектирование логики сценария  Создание диалогов |
| Рабочая среда | Стандартизированные ПК, локальная сеть |

Пользовательские сценарии для выделенных профилей пользователей.

Примером могут служить приведенные ниже сценарии действий пользователей.

1. Пользователь запускает приложение. Приложение предоставляет ему возможность «Регистрация»
2. Приложение позволяет авторизоваться в приложении (аккаунт уже имеется).
3. Приложение позволяет восстановить пароль (Забывчивость пользователей).
4. Пользователь загружает или создает проект со сценариями.
5. Пользователь создает новый сценарий.
6. Проектирует логику сценария.
7. Редактирование узлов сценария

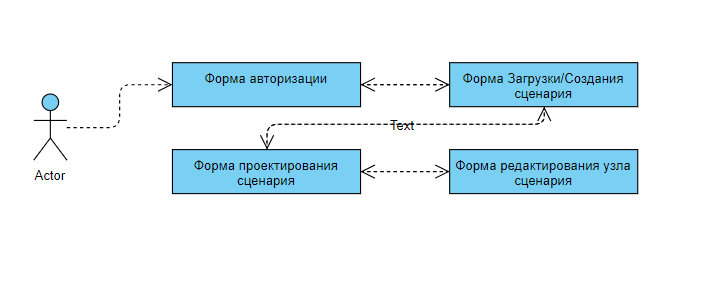
По функциональным блокам можно отнести действия пользователя:

* По авторизации в приложении (1,2,3).
* По работе со списком сценариев (4,5).
* По работе построения сценарной логики (6).
* По работе с отдельными узлами (7)

Получается 4 функционирующие формы:

* Форма авторизации;
* Форма Загрузки/Создания сценария;
* Форма проектирования сценария;
* Форма редактирования узла сценария.

Схема навигации



Низкоуровневое проектирование интерфейса:

количественная оценка и построение прототипа

1 Цели работы

1.1 Закрепить теоретические знания по разработке пользовательского интерфейса.

1.2 Развить навыки создания вариантов прототипов интерфейса пользователя.

1.3 Получить практические навыки, по количественной оценке, интерфейса на этапе низкоуровневого проектирования.

1.4 Закрепить принципы обоснования выбора прототипа интерфейса по его количественной оценке.

2 Задание

2.1 Используя разработанное приложение, провести количественную оценку элементов интерфейса. Метод количественной оценки – GOMS, информационная производительность, символьная эффективность.

2.2 По результатам количественной оценки сделать выводы о возможности усовершенствования интерфейса.

2.3 При возможности внести необходимые усовершенствования в модели форм и реализовать их в среде разработки приложения. Каждую форму следует снабдить описанием навигации по ней. После улучшения получить те же количественные оценки для усовершенствованного варианта.

2.4 Составить отчет со скриншотами, пояснениями и выводом.

Для анализа качества интерфейсов используется множество количественных и эвристических методов. Одним из лучших подходов к количественному анализу моделей интерфейсов является классическая модель GOMS (goals, objects, methods and selection rules).

Метод, использующий модель GOMS, основан на разбиении всех действий пользователя на отдельные составляющие. Для каждой из них с помощью тщательных лабораторных исследований получен набор временных интервалов, необходимых для ее выполнения. В таблице 1 приведена номенклатура элементарных действий и соответствующие временные интервалы.

|  |  |  |
| --- | --- | --- |
| Нажатие клавиши клавиатуры, включая клавиши Alt, Ctrl, Shift | 0,28 с | К |
| Нажатие клавиши мыши | 0,1 с | М |
| Указание – перемещение курсора мыши, чтобы указать какую-либо позицию на экране монитора | 1,1 с | П |
| Перемещение- перенос руки пользователя с клавиатуры на мышь или обратно | 0,4 с | В |
| Ментальная подготовка – мысленный выбор пользователем своего следующего элементарного действия | 1,2 | Д |
| Ответ – реакция системы на элементарное действие пользователя | - | Р |

Широкая изменяемость каждой из представленных мер объясняет, почему эта модель не может использоваться для получения абсолютных временных значений с высокой степенью точности. Но этот метод вполне пригоден для проведения сравнительной оценки между какими-либо двумя моделями интерфейса по уровню эффективности их использования.

Расчет времени, необходимого для выполнения некоторого действия начинают с разбиения его на элементарные действия, которые соответствуют номенклатуре приведенной в таблице 1. Проще всего выделить движения К, М, П, В. Проблему составляет определение моментов, когда пользователь должен остановиться, чтобы выполнить бессознательную ментальную операцию.

Основные правила по выявлению этих моментов

|  |  |
| --- | --- |
| **Правило 0**  Начальная расстановка операторов Д | Операторы Д следует устанавливать перед всеми операторами К и М (нажатие клавиши), также перед всеми операторами П, предназначенными для выбора команд. Но перед операторами П, предназначенными для указания на аргументы этих команд, ставить оператор Д не следует. |
| **Правило 1**  Удаление ожидаемых операторов Д | Если оператор, следующий за оператором Д, является полностью ожидаемым с точки зрения оператора, предшествующего Д, то этот оператор Д может быть удален.  Если пользователь перемещает мышь с намерением нажать на ее кнопку по достижении цели движения, то в соответствии с этим правилом следует удалить оператор Д, установленный по правилу 0. Так последовательность действий П Д К преобразуется в П К. |
| **Правило 2**  Удаление операторов Д внутри когнитивных единиц | Если строка Д К Д К Д К… принадлежит когнитивной единице, то следует удалить все операторы Д, кроме первого. Когнитивной единицей является непрерывная последовательность вводимых символов, которые образуют название команды или аргумент. Например *Y*, *перемещать*, *4564.23 –* это когнитивные единицы. |
| **Правило 3**  Удаление операторов Д перед последовательными разделителями | Если оператор К означает лишний разделитель, стоящий в конце когнитивной единицы (например, разделитель команды, следующий сразу за разделителем аргумента этой команды), то следует удалить оператор Д , стоящий перед ним. |
| **Правило 4** Удаление операторов Д, которые являются прерывателями команд | Если оператор К является разделителем, стоящим после постоянной строки (например, название команды или любая последовательность символов, которая каждый раз вводится в неизменном виде), то следует удалить оператор Д, стоящий перед ним. (Добавление разделителя станет привычным действием, и поэтому разделитель станет частью строки и не будет требовать специального оператора Д.) Но если оператор К является разделителем строки аргументов или любой другой изменяемой строки, то оператор Д следует сохранить перед ним. |
| **Правило 5**  Удаление перекрывающих операторов Д | Любую часть оператора Д, которая перекрывает оператор Р, означающий задержку, связанную с ожиданием ответа компьютера, учитывать не следует. |

В этих правилах под ***строкой*** понимается некоторая последовательность символов.

***Разделителем*** считается символ, которым обозначено начало или конец значимого фрагмента текста, такого как, например, слово естественного языка или телефонный номер. Так, пробел является разделителем для большинства слов, а точка используется в конце предложений для разделения. В качестве разделителей могут выступать скобки для ограничений пояснений или замечаний и т.д.

Если для выполнения команды требуется дополнительная информация, она называется здесь ***аргументом*** данной команды.

Правила GOMS позволяют определит время, необходимое пользователю для выполнения любой. четко сформулированной задачи, для которой данный интерфейс предусмотрен. Однако этого метода недостаточно,

чтобы оценить, насколько быстро должен работать интерфейс – его производительность.

Рассмотри следующий сценарий поведения пользователя: Вход в систему – Загрузка проекта – Открытие отельного элемента проекта.

* + - 1. Перемещение руки и мыши (B)
      2. Указание на поле логина (П)
      3. Нажатие клавиши мыши (М)
      4. Перемещение рук клавиатуре (В)
      5. Ввод логина (K\*n)
      6. Перемещение руки и мыши (B)
      7. Указание на поле пароль (П)
      8. Нажатие клавиши мыши (М)
      9. Перемещение рук клавиатуре (В)
      10. Ввод пароля (K\*n)
      11. Перемещение руки и мыши (B)
      12. Указание на кнопку входа (П)
      13. Нажатие клавиши мыши (М)
      14. Перемещение руки и мыши (B)
      15. Указание на кнопку загрузки (П)
      16. Нажатие клавиши мыши (М)
      17. Перемещение руки и мыши (B)
      18. Указание на проект (П)
      19. Нажатие клавиши мыши (М)

В П М В К\*n В П М В К\*n В П М В П М В П М

По правилу 0: В П Д М В Д К\*n В П Д М В Д К\*n В П Д М В П Д М В П Д М

По правилу 1: В П Д М В Д К\*n В П Д М В Д К\*n В П М В П М В П М

Правила 2-5 оставят прежний результат для данной последовательности действий.

N логина возьмем 15, а пароля 8

0.4 + 1.1 + 1.2 + 0.1 + 0.4 + 1.2 + 0.28\*15(4.2) + 0.4 + 1.1 + 1.2 + 0.1 + 0.4 + 1.2 + 0.28\*8(2.24) + 0.4 + 1.1 + 0.1 + 0.4 + 1.1 + 0.1+ 0.4 + 1.1 + 0.1 = 20.04

Лабораторная работа № 7

Проектирование справочной системы

Цель работы: изучение методики создания файлов справочной системы Windows (\*.hlp) при разработке приложений.

Задание:

1. Создать глоссарий – перечень уникальных понятий.

2. Спроектировать справочную систему для приложения.

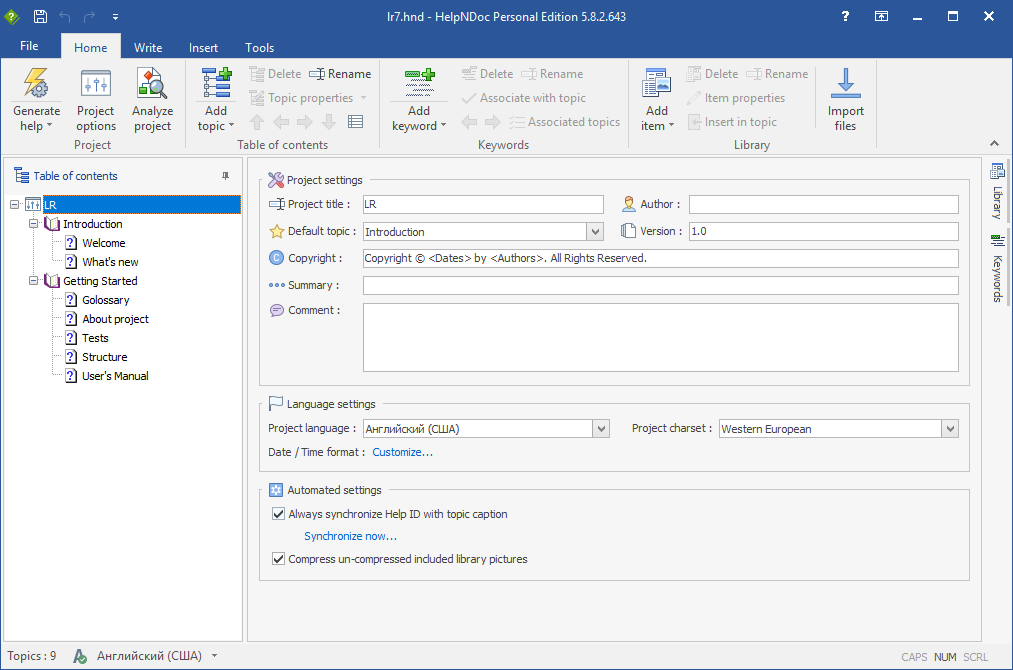
3. Обеспечить возможность поиска по ключевым словам.

4. Решить такие вопросы проектирования справочной системы как вид основного окна, введение изображений, наличие горячих областей и ссылок.

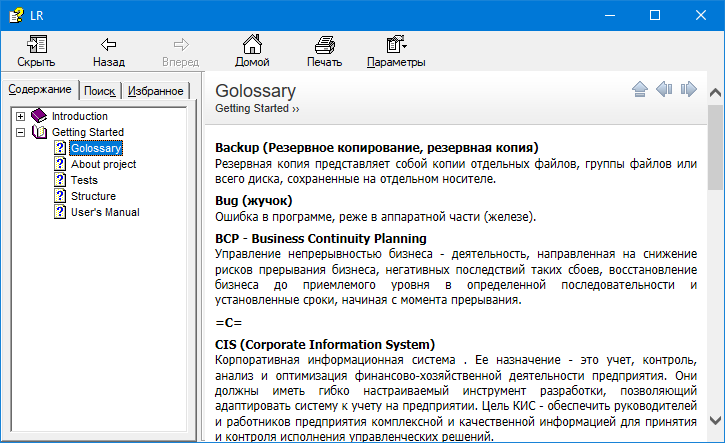
5. В результате выполнения работы получить программу с функциональным справочником, поиском, указателем. Возможно также и создание контекстной справки.

6. Составить отчет со скриншотами, пояснениями и выводом.

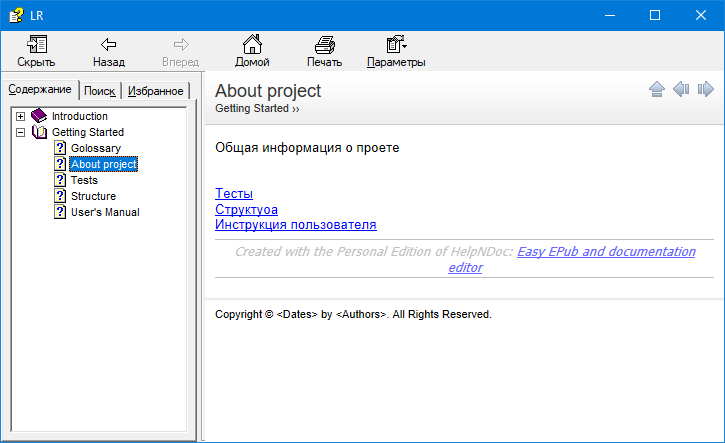
Построение структуры (самого проекта) справочника.



Глоссарий



Ссылки в справочнике



## Лабораторная работа № 8

**Тема: Диалог виды (проектирование диалога,диалог на основе экранных форм, выбор структуры диалога, диалог типа Вопрос – Ответ…)**

**Цель**: работы: изучение методики создания диалогов либо другой интерактивной документации.

**Назначение:** овладение навыками и приемами создания пользовательского интерфейса в среде Windows. Эти же приемы будут применяться в последующем во время создания прикладных приложений и других дополнений к Windows. Навыки работы с интерфейсами позволят кастомизировать и упростить операции целевого пользователя.

**Задания**

1. Описать вид диалога, который используется в приложении, разработанном в предыдущих лабораторных работах.
2. Чем характеризуется выбор данного типа диалога?
3. Опишите другие варианты диалога для вашего приложения, какие элементы будут использоваться для организации того же функционала? Изобразить иной вариант диалога в виде эскиза.
4. Структура диалога типа вопрос-ответ?
5. Диалог на основе меню?
6. Диалог на основе экранного меню?
7. Составить отчет со скриншотами, пояснениями и выводом.
   * + 1. Диалог в приложении

Диалог с пользователем осуществляется по типу Диалог на основе меню.

* + - 1. Выбор был вызван в связи с тем, что редактирование структуры сценариев необходимо выбирать различные узлы взаимодействия и настраивать их состояния в проекте.
      2. Другим вариантом будет

Диалог типа «Вопрос - Ответ». В данном варианте пользователю будут предлагаться последовательный выбор разных нод далее их настройка и связь со следующей. Но данный тип общения с пользователя будет не удобен в связи сложностью построения больших сценарных проектов в приложении

* + - 1. Структура диалога типа вопрос-ответ

Структура диалога типа «вопрос-ответ» основана на аналогии с обычным интервью. Система берет на себя роль интервьюера и получает информацию от пользователя в виде ответов на вопросы. Это наиболее известная структура диалога; все диалоги, управляемые компьютером, в той или иной степени состоят из вопросов, на которые пользователь отвечает. Однако в данной структуре этот процесс выражен явно. В каждой точке диалога система выводит в качестве подсказки один вопрос, на который пользователь дает один ответ. В зависимости от полученного ответа система может решить, какой следующий вопрос задавать. Структура предоставляет естественный механизм ввода как управляющих сообщений (команд), так и данных. Никаких ограничений на диапазон или тип входных данных, которые могут обрабатываться, не накладывается. Существуют системы, ответы в которых даются на естественном языке, но чаще используются предложения из одного слова с ограниченной грамматикой.

Диалог в виде вопросов и ответов в достаточной степени обеспечивает поддержку пользователя, так как даже краткий наводящий вопрос при разумном построении может быть самопоясняющим. Эта структура не гарантирует минимального объема ввода, оцениваемого по количеству нажатий клавиш, однако при подходящем подборе сокращений можно уменьшить любую избыточность. Вместе с тем структура обладает одним существенным недостатком. 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.

* + - 1. Диалог на основе меню

*Меню* — это наиболее удобная структура диалога для неподготовленных пользователей; жесткая очередность открытия и иерархическая вложенность меню может вызывать раздражение профессионала, замедлять его работу. 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

• 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.

* + - 1. Диалог на основе экранного меню

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. На практике формы используются в основном там, где учет какой-либо деятельности требует ввода достаточно стандартного набора данных. Человек, заполняющий форму, может выбирать последовательность ответов, временно пропускать некоторый вопрос, возвращаться назад для коррекции предыдущего ответа и даже «порвать бланк» и начать заполнять новый. Он работает с формой до тех пор, пока не заполнит ее полностью и не передаст системе. Программная система может проверять каждый ответ непосредственно после ввода или выждать и вывести список ошибок только после заполнения формы целиком. В некоторых системах информация, вводимая пользователем, становится доступной только после нажатия клавиши «ввод» по окончании заполнения формы.