**Лекция 4 ПрЧМИ 2019**

**3. Проектирование интерфейса**

***Сведения из теории***

При проектировании интерфейса, кроме словесного описания, используют различные виды диаграмм, приспособленные к пользовательскому интерфейсу.

**Практическая часть (Пример)**

При разработке данной легенды предполагается в основном использование текстового описания и изображения структур экранов.

**3.1.** **Проектирование структуры экранов системы**

***Сведения из теории***

Основываясь на сценариях работы и ролях пользователей, формируется структура экранов системы, т.е. определяется количество экранов, функциональность каждого из них, навигационные связи между ними, формируется структура меню и других навигационных элементов.

По сути на этом этапе выделяются отдельные функциональные блоки и определяется, как именно эти блоки связываются между собой. Под функциональными блоками будем подразумевать функцию или группу функций, связанных по назначению или области применения в случае программы, и группу функций/фрагментов информационного наполнения в случае сайта.



Рис. 1. Структура экранов системы

На рис.1 приведены типичные структуры сайта (слева) и программы. Если сайты обычно разветвлены, т.к. функции обычно размещаются в отдельных экранах, то программы обычно имеют только один изменяющийся экран, в котором и вызываются почти все функции.

Проектирование общей структуры состоит из двух параллельно происходящих процессов: выделение независимых блоков и определение связи между ними. Если проектируется сайт, в завершении необходимо также создать схему навигации.

При выделении независимых блоков рекомендуется избегать помещения в один блок более трех функций, поскольку каждый блок в результирующей системе будет заключен в отдельный экран или группу управляющих элементов. Перегружать же интерфейс опасно. Результатом этой работы должен быть список блоков с необходимыми пояснениями.

Существует три основных вида связи между блоками. Это логическая связь, связь по представлению пользователей и процессуальная связь.

***Логическая связь*** определяет взаимодействие между фрагментами системы с точки зрения разработчика. Полученные связи очень существенно влияют на навигацию в пределах системы (особенно, когда система многооконная). Поэтому чтобы не перегружать интерфейс, стоит избегать как слишком уж отдельных блоков (их трудно найти), так и блоков, связанных с большим количеством других.

***Связь по представлению пользователей.*** В информационных системах, когда необходимо гарантировать, что пользователь найдет всю нужную ему информацию, необходимо устанавливать связи между блоками, основываясь не только на точке зрения разработчика, но и на представлениях пользователей. Дело в том, что самый распространенный способ поиска (поиск по классификации признаков) работает только в том случае, когда пользователи согласны с принципами этой классификации. Большинство же понятий однозначно классифицировать нельзя из-за наличия слишком большого количества значимых признаков. Проблема также состоит в том, что реальный классификационный признак может отличаться от широко распространенного. Например, помидор, который все считают овощем, на самом деле ягода. Это значит, что классификация, приемлемая для ботаника, не будет работать для всех остальных, причем обратное не менее справедливо.

***Процессуальная связь*** описывает не вполне логичное, но естественное для имеющегося процесса взаимодействие. Жестко заданная связь упрощает процесс разработки системы и позволяет уменьшить количество ошибок пользователя. Классическим примером жестко заданной процессуальной связи является устройство мастеров, при котором пользователя заставляют нажимать кнопку «Далее».

**Практическая часть (Пример)**

На рис. 2 показана структура экранов для приложения.

5

4

2

1

3

Рис. 2. Структура экранов программы

На рисунке цифрами обозначены следующие экраны:

1 – основной экран;

2 – создание теста;

3 – прохождение теста;

4 – справка о приложении;

5 – справка о компоненте.

**3.2.** **Проектирование навигационной системы**

*Сведения из теории*

На основе разработанной ранее структуры экранов на этом этапе выбирается наиболее адекватная навигационная система и разрабатывается её детальный интерфейс.

Навигационная система показывает механизм распределения функций и задач между окнами программы. Навигационная система определяет, каким образом пользователи смогут перемещаться как между различными задачами, так и внутри отдельной задачи. Например, можно ли будет оставить частично завершенную задачу и начать другую.

Как правило, на этом этапе не создается отдельного отчета; разработанный интерфейс в дальнейшем описывается в отчете, посвященном низкоуровневому проектированию.

**Практическая часть (Пример)**

Согласно рис.2 связи между экранами изображены линиями без стрелок. Это означает, что возможен переход между экранами в обе стороны. Выход из приложения возможен только из основного экрана.

**3.3. Низкоуровневое проектирование**

*Сведения из теории*

Разрабатываются интерфейсы конкретных экранов системы (состав, взаимное расположение и поддерживающие тексты интерфейсных элементов).

При проектировании основных экранов разрабатываются их интерфейсы. Используя информацию предыдущих этапов, производится полное описание интерфейса главных экранов системы (без обработки исключительных ситуаций).

К отчету прилагаются макеты экранов с описаниями функциональности каждого интерфейсного элемента.

При проектировании второстепенных экранов разрабатываются их интерфейсы. К ним относятся диалоговые окно и всевозможные сообщения.

**Практическая часть (Пример)**

Согласно легенды программа имеет два режима работы: создание тестового файла из текстового файла; создание тестового файла в режиме диалога.

Формирование и тестирование реализуется с помощью применения формирования элементов теста. Элемент теста представляет собой вопрос и набор вариантов ответов. Правильных ответов может быть один или несколько. Вариантов ответов может быть не больше пяти.

В процессе диалога вводится вопрос и варианты ответов, которые записываются в тестовый файл.

Если элемент теста вводится из файла, то структура файла следующая: несколько строк может занимать вопрос, признаком окончания вопроса является символ точка с запятой; затем в каждой из пяти строк содержится вариант ответа, признаком окончания текста ответа является символ точка с запятой; после этого следуют номера правильных ответов, перечисленные через запятую и ограниченные символом точка с запятой. Если вариантов ответов меньше, чем пять, то формируются пустые строки.

Тестовые файлы и названия тем тестов должны храниться в каталоге с именем “Test”, в котором хранятся текстовые файлы с расширением \*.tst и один файл с расширением \*.tema. Файлы с расширением \*.tst, должны иметь имена, совпадающие с именами тем.

***Проектирование основных экранов***

На рис. 3, 4, 5, 6 показаны основной экран, экраны создания теста, экран тестирования.

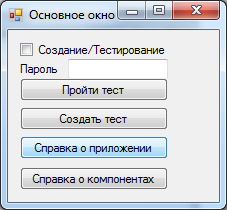


Рис. 3. Основной экран

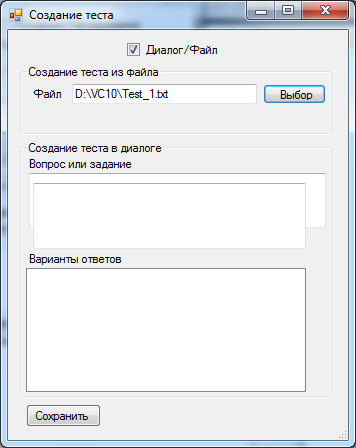


Рис.4. Экран создания теста из файла

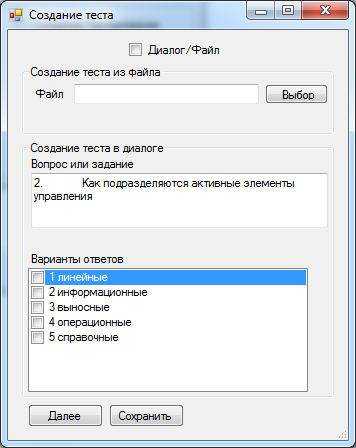


Рис. 5. Экран создания теста с помощью диалога

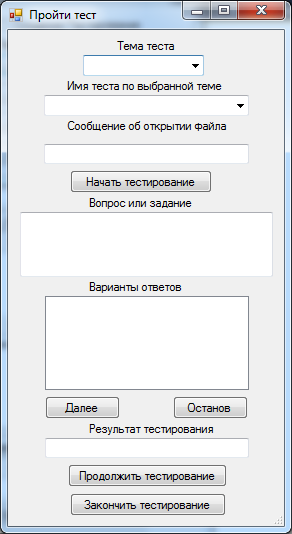
******

Рис. 6. Экран прохождения теста

***Проектирование второстепенных экранов***

На рис. 7, 8 показаны вспомогательные экраны, которые выдают информацию по программе и используемых компонентах, представленных в задании.

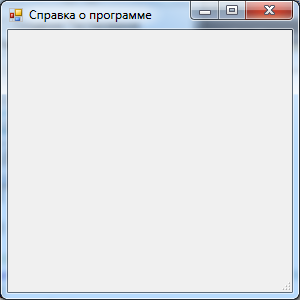


Рис. 7. Экран справки о программе

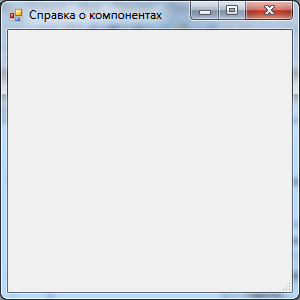


Рис. 8. Экран справки о компонентах

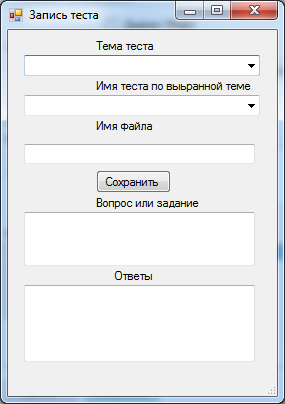


Рис. 9. Экран записи теста в файл