

Министерство науки и высшего образования Российской
Федерации
Федеральное государственное бюджетное образовательное
учреждение высшего профессионального образования
«Ивановский государственный энергетический
университет имени В.И. Ленина»
Кафедра «Программное обеспечение компьютерных систем»

Проектирование интерфейса компьютерных приложений

Методические указания

Иваново 2019

Составитель В.П. Зубков
Редактор Е.Р. Пантелеев

Методические указания содержат задания для выполнения лабораторных работ по курсу «Проектирование интерфейса компьютерных приложений», а также методический материал, необходимый для выполнения работ, примеры выполнения.

Методические указания предназначены для выполнения лабораторных работ бакалаврами, обучающимися по направлению 231000.62 «Программная инженерия» (профиль «Разработка программно-информационных систем») и изучающие курс «Проектирование интерфейса компьютерных систем».

Утверждено цикловой методической комиссией ИВТФ

РЕЦЕНЗЕНТ

д-р техн. наук, проф. И.Д. Ратманова (ФГБОУВПО «Ивановский
государственный энергетический университет имени
В.И.Ленина»)

2019

ВВЕДЕНИЕ

Основной целью преподавания дисциплины «Проектирование интерфейса компьютерных приложений» является развитие профессиональной компетенции ПК-22 «Способность создавать программные интерфейсы» студентов в области проектирования, реализации и тестирования человеко-машинных интерфейсов в процессе разработки программной продукции.

Дисциплина «Проектирование интерфейса компьютерных приложений» является составляющей базовой части учебного плана в структуре подготовки бакалавров по направлению «Программная инженерия». В методических указаниях представлены варианты заданий, способствующие приобретению навыков составления проекта человеко-машинного интерфейса, проектирования экранов, тестирования, составления помощи.

Целью практических занятий по курсу «Проектирование интерфейса компьютерных приложений» является знакомство с технологией разработки и тестирования интерфейсных элементов. При этом выполняется анализ требований, предъявляемых к разрабатываемым компонентам, реализуются основные этапы проектирования интерфейса, его тестирования, создается система обучения и документации. В качестве средства разработки компонентов используются различные среды программирования.

Для указанного визуального компонента нужно сформулировать легенду, т.е. небольшую формулировку задачи. Например, для визуального элемента «ProgressBar» формулировка может быть следующей: *«Разработать приложение копирования файлов, соответствующих именам или шаблонам имен из заданного каталога в указанный каталог или в указанные каталоги».*

При реализации легенды могут использоваться другие визуальные элементы. Надо реализовывать требования только для указанных визуальных элементов в задании.

Практикум включает в себя текстовый отчет и разработанную программу. Программа должна быть разработана при завершении практикума.

Все пункты и подпункты отчета при проектировании рассматриваются с точки зрения легенды. При разработке приложения

необходимо полностью выполнить требования для визуальных элементов, указанных в задании, и учесть требования или пожелания преподавателя при создании кода для легенды.

Структура отчета по лабораторным работам приведена в прил. 1.

Лабораторная работа 1. Постановка задачи (первая часть)

Цель работы

Практическое освоение методов разработки проектирования человеко-машинного интерфейса.

Студент должен знать:

- о формализации контекста использования;
- формализации объективных критериев успеха;
- определении необходимой функциональности системы;
- способах проведения анализа целей;
- способах проведения анализа действий пользователей.

Студент должен приобрести навыки:

- описания формализации контекста использования;
- описания формализации объективных критериев успеха;
- описания определения необходимой функциональности системы;
- описания проведения анализа целей;
- описания проведения анализа действий пользователей.

Руководство по лабораторной работе 1

1. Общие сведения

Лабораторная работа 1 является первой частью лабораторного практикума. Она начинается с описания постановки задачи, которая является вторым разделом отчета. Титул и описание предметной области создаются в конце лабораторного практикума. В разделе «Процесс разработки» по каждому пункту приводятся сведения из теории и практическая часть.

Введение и первый пункт лабораторного практикума описываются позже, после создания четырех разделов, начиная со второго раздела. Поэтому процесс разработки начинается со второго раздела «Постановка задачи». Это объясняется тем, что в самом начале студент не очень ясно представляет, как надо

описывать предметную область и как сжато и понятно описать всю работу во введении.

Процесс разработки

2. Постановка задачи

Методические указания

Приводится формулировка задания и формулировка легенды (формулировки задачи для некоторой предметной области). Легенда должна представлять собой задание на разработку приложения, в котором используется наряду с другими визуальными элементами визуальный компонент задания.

Практическая часть(пример)

Задание

Разработать визуальные компоненты, реализующие элементы управления CheckBox и RadioButton и позволяющие изменять свой внешний вид путем задания их цвета или вида элементов из файла формата BMP. Компонент должен поддерживать темы ОС Windows.

Легенда

Создать приложение, реализующее тестирование 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. Элемент теста может формироваться в диалоге или вводиться из текстового файла. В процессе диалога вводится вопрос и варианты ответов. 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.

2.1. Формализация контекста использования

Методические указания:

Описать характеристики пользователей вашего создаваемого приложения:

- опыт работы с компьютером, знание предметной области, мотивы, размер/важность групп пользователей, образцы (типовые ситуации) использования;
- цели и задачи пользователей;
- задачи проекта (причина создания проекта, его этапы, результаты, которые должны быть получены, необходимая информация);
- технология разработки и платформа, на которой будут работать пользователи;
- среда, в которой будет создаваться и использоваться проект.

Практическая часть (Пример)

Сведения о пользователях

Пользователей разрабатываемых компонентов можно разделить на две основные группы:

1. Конечные пользователи систем, в которых будут использоваться разрабатываемые компоненты. К этой группе относятся люди, имеющие минимальные навыки работы с компьютером (основные приемы работы с Windows, программами MS Office, почтовыми системами и т.п.), либо не имеющие этого опыта вообще.
2. Программисты, которые будут использовать компоненты CheckBox и RadioButton и элемент теста в своих разработках.

Цели и задачи пользователей

Целью первой группы пользователей является создание приложения, представляющего собой некоторый тест.

Целью второй группы пользователей является удобное и простое подключение компонентов к разрабатываемой системе и обеспечение их корректной работы в рамках проекта в целом.

Задачи проекта

Компонент элемент теста является составной частью большинства разрабатываемых тестовых систем. Значительное упрощение процесса составления тестов является задачей проекта.

Такие компоненты, как CheckBox и RadioButton, применяются практически на каждой интерфейсной форме в любой из компьютерных систем. Однако, стандартные компоненты имеют единообразное практически ненастраиваемое исполнение, что снижает удобство и наглядность их использования. В связи с этим задачей проекта является добавление к стандартным компонентам функций, позволяющих изменить внешний вид этих компонентов (задание их цвета или вида элементов из файла формата BMP).

Технология разработки и платформа

Компоненты разрабатываются в среде Visual Studio с использованием языка C#. Требуемая операционная среда – Windows 7 и более поздние версии Windows.

2.2. Формализация объективных критериев успеха

Методические указания

1. Выделить объективные критерии оценки эргономичности интерфейса (показатели эффективности, продуктивности, удовлетворенности пользователей).

2. Создать реальное задание на проектирование интерфейса.

Практическая часть (Пример)

Основными критериями успеха у пользователей являются простота, удобство и наглядность использования компонентов.

Интерфейс должен обладать свойством быстрого освоения работы с ним пользователей-непрофессионалов.

Для программистов разработанные компоненты интерфейса должны обладать свойством удобного подключения компонентов к проектам.

2.3. Определение необходимой функциональности системы

Методические указания

1. Использовать анализ целей для определения функциональности.
2. Использовать анализ действий пользователей для определения функциональности.
3. Используя результаты анализа, описать функциональность.

Практическая часть (Пример)

Компонент элемент теста формируется в диалоговом режиме или путем ввода информации из файла. Для установления режима ввода предназначен компонент CheckBox. Если галочка не установлена, то происходит ввод в диалоге с помощью текстовых полей, если галочка установлена, то происходит ввод информации из файла.

Компонент CheckBox используется для того, чтобы пользователь с помощью мыши или клавиатуры мог указать свое решение типа да/нет или да/нет/не уверен (в последнем случае флажок устанавливается, но выглядит блеклым). Это решение отражается в специальном свойстве компонента, доступном как для чтения, так и для записи. В составе диалогового окна может быть несколько компонентов CheckBox, причём состояние любого из них никак не зависит от состояния остальных. Также в компоненте должна существовать возможность определения положения текста надписи (левое или правое выравнивание). В соответствии с заданием допускается изменять цвет и внешний вид элементов.

В отличие от компонентов CheckBox компоненты RadioButton представляют собой переключатели, предназначенные для выбора одного из нескольких взаимоисключающих решений. Выбор осуществляется также с помощью мыши или клавиатуры. На форму (точнее, в компонент-контейнер) помещается, по меньшей мере, два таких компонента. Они могут иметь только два состояния, определяемых соответствующим свойством. Если в одном компоненте это свойство принимает значение True, во всех других компонентах, расположенных в том же контейнере, эти свой-

ства принимают значения False. Аналогично компоненту CheckBox задаётся свойство выравнивания и свойства, позволяющие изменять цвет и внешний вид элементов.

2.4. Анализ целей разработки

Методические указания

1. Используя результаты анализа целей пользователя проанализировать формулировку легенды и задания и выявить цели разработки.

2. Описать цели разработки, учитывая интересы пользователей и программистов.

Практическая часть (Пример)

Как выше было отмечено, целью первой группы пользователей является создание приложения, представляющего собой некоторый тест. Было упомянуто, что с помощью элемента CheckBox, условное имя которому «Указатель источника», устанавливается, каким образом формируется элемент теста (формирование с помощью диалога либо формирование из файла). С помощью другого элемента CheckBox, условное имя которого «Режим работы», устанавливается, как работает приложение (создание теста или функционирование теста). Для защиты от исправления теста во время функционирования необходимо предусмотреть ввод пароля, который защищает тест от вмешательства в процесс его разработки, т.е. если введенный пароль верен, то приложение работает в режиме формирования теста, если не верен или не введен, то приложение работает в режиме тестирования.

Таким образом, цель первой группы пользователей содержит две подцели: указание режима работы и указание источника формирования элемента теста.

Целью второй группы пользователей является удобное и простое подключение компонентов к разрабатываемой системе и обеспечение их корректной работы в рамках проекта в целом.

Эта цель может быть реализована с помощью решения следующих задач:

- создание класса CheckBox1, включающего класс CheckBox;

- создание класса RadioButton1, включающего класс RadioButton;
- создание класса CreateTest, реализующего создание элемента теста.

2.5. Анализ действий пользователей

Методические указания

1. Для типовых сценариев проанализировать действия пользователей при работе с визуальными элементами.
2. Описать типичные действия пользователей для типовых сценариев.

Практическая часть (Пример)

Первая группа пользователей для выбора альтернатив может использовать как клавиатуру, так и мышь. При этом у клавиатуры скорее вспомогательная роль, так как обычно при использовании подобных компонентов для пользователя более обычной и удобной является мышь.

Клавиатура в этих случаях используется достаточно редко, только для некоторых крайних случаев. Так, клавиша табуляции применяется для перемещения между компонентами, а функцию, аналогичную функции левой кнопки мыши, здесь играет пробел.

Вторая группа пользователей использует созданные классы в привычной среде разработки приложения.

Лабораторная работа 2. Постановка задачи (вторая часть)

Цель работы

Практическое освоение методов разработки проектирования человеко-машинного интерфейса.

Студент должен знать:

- об определении низкоуровневых и высокоуровневых функций;
- формализации бизнес-ролей пользователей;
- формализации функциональности;
- формализации сценариев действий пользователей;
- данные обзора интерфейсов конкурирующих систем;
- формализации привычек и ожиданий пользователей.

Студент должен приобрести навыки:

- описания низкоуровневых и высокоуровневых функций;
- описания формализации бизнес-ролей пользователей;
- описания формализации функциональности;
- описания формализации сценариев действий пользователей;
- описания обзора интерфейсов конкурирующих систем;
- описания формализации привычек и ожиданий пользователей.

Руководство по разработке лабораторной работы 2

Общие сведения

Лабораторная работа 2 является второй частью лабораторного практикума. В первой лабораторной работе были приведены цели обеих групп пользователей, проведен их анализ. При анализе целей цели групп пользователей могут делиться на подцели или представлять собой совокупность действий или задач для достижения цели.

Множество функций, соответствующих заданию, содержат низкоуровневые функции, позволяющие пользователю изменять характеристики свойств визуальных элементов задания. К высокоуровневым функциям задания относится выдача различных справок о визуальных элементах задания.

Множество функций, соответствующих легенде, содержат высокоуровневые функции, как для пользователя, так и для разработчика.

Процесс разработки

2.6. Постановка задачи

Методические указания

1. Определить высокоуровневые функции приложения.
2. Определить низкоуровневые функции приложения.
3. Определить низкоуровневые функции визуального компонента.
4. Описать высокоуровневые и низкоуровневые функции.

Практическая часть (Пример)

К высокоуровневым функциям легенды относятся следующие:

- указание источника формирования элемента теста;
- указание режима работы;
- ввод пароля;
- создание теста;
- прохождение теста;
- выдача справки о приложении.

Низкоуровневыми функциями легенды являются выдача контекстной помощи о визуальных элементах легенды и выдача сообщений об ошибках.

К низкоуровневым функциям задания относятся:

- изменение цвета элементов (текста и фона);
- изменение вида элементов (шрифта и вида).

2.7. Формализация бизнес-ролей пользователей

Методические указания

1. Определить роли различных групп пользователей.
2. Определить роли программистов.
3. Описать роли.

Практическая часть (Пример)

В данном приложении предусмотрены три роли:

- пользователь, который проходит тест (первая группа);
- пользователь, который разрабатывает тест (вторая группа);
- пользователь-программист, который использует разработанные классы компонент в своей разработке (третья группа).

Для пользователей первой группы разрабатываются различные виды тестов, которые они проходят. Пользователи второй группы создают тесты для различных направлений.

2.8. Формализация функциональности

Методические указания

1. Окончательно сформировать список функциональных возможностей системы.
2. Проанализировать необходимость и достаточность списка функций.

Практическая часть (Пример)

Для пользователей второй группы при создании теста нужно указать источник. Если нужно указать файл, то его либо нужно выбрать в директории, либо задать путь.

Для пользователей первой группы для прохождения теста нужно указать местонахождение файла с тестом, для этого надо выполнить функции для файла, аналогичные для пользователей второй группы.

Функциональность разрабатываемых компонентов состоит в задании определенных параметров путём выбора некоторого набора или только одного параметра из заданного множества (каждый параметр определяется одним компонентом в активном или неактивном состоянии). Выбор осуществляется с помощью мыши или клавиатуры.

Дополнительная функциональность компонентов CheckBox и RadioButton состоит в изменяемости внешнего вида для разработчиков; для пользователей же – в изменённом внешнем виде, что придает компоненту дополнительную гибкость в его использовании и позволяет получить соответствие с внешним видом системы-хозяина.

2.9. Формализация сценариев действий пользователей

Методические указания

1. Составить все типовые сценарии для приложения.
2. Проанализировать достижение целей пользователей с помощью типовых сценариев.

Практическая часть (Пример)

Первый типовой сценарий: «Пользователь решил создать тест по математике по разделу «Решение простых дифференциальных уравнений». Для этого он предварительно создал в текстовом файле тест».

Второй типовой сценарий: «Пользователь решил пройти тест. Для этого он должен указать расположение файла с тестом и получить оценку в некоторой шкале за этот тест».

Для компонента *CheckBox* возможен следующий типовой сценарий: «Допустим, пользователь нажал левой клавишей мыши на некотором компоненте *CheckBox*. Если до этого компонент был неактивен, он переходит в активное состояние. Если он был в активном состоянии, он может перейти в неактивное состояние либо в промежуточное состояние («не уверен») – в зависимости от настроек компонента».

Для компонента *RadioButton* возможен следующий типовой сценарий: «Допустим, пользователь нажал левой клавишей мыши на некотором компоненте *RadioButton*. Если до этого компонент был неактивен, он переходит в активное состояние. Если он был в активном состоянии, он в нём и остаётся. Далее пользователь может продолжать аналогичные действия с данным компонентом или другими, ему аналогичными, в этой группе или другой. При этом в группе может быть только один активный компонент».

2.10. Обзор интерфейса конкурирующих систем

Методические указания

1. Найти в Интернете один или два прототипа разрабатываемого приложения.
2. Отметить недостатки и достоинства в интерфейсе прототипов.
3. Описать, какие элементы интерфейса хотите позаимствовать.

Практическая часть (Пример)

На рис. 1, 2 приведены интерфейсы приложений тестирования по учебным дисциплинам.

The screenshot shows a window titled "Выбор теста" (Test Selection) with a blue background. It contains the following elements:

- Field "Ваша фамилия и имя (в Род. падеже):" with the value "Билл Гейтс".
- Field "Выберите свой класс:" with the value "11".
- Field "Выберите предмет:" with the value "история Каз-на".
- Field "Вид тестирования:" with the value "Тестирование".
- Field "Выберите тему:" with a list of variants and scores:

Вариант	Оценка
Вариант 9	25
Вариант 10	25
Вариант 11	25
Вариант 12	25
Вариант 13	25
Вариант 14	25
Вариант 15	24
Вариант 16	25

Below the list is a button labeled "Начать" (Start).

Рис.1. Интерфейс системы тестирования

На рис.1 показан интерфейс системы тестирования. Приведенный интерфейс неудачно оформлен с точки зрения дизайна. Не учтены следующие критерии эргономики: На рис. 2 показан интерфейс системы создания тестов и прохождения тестов.

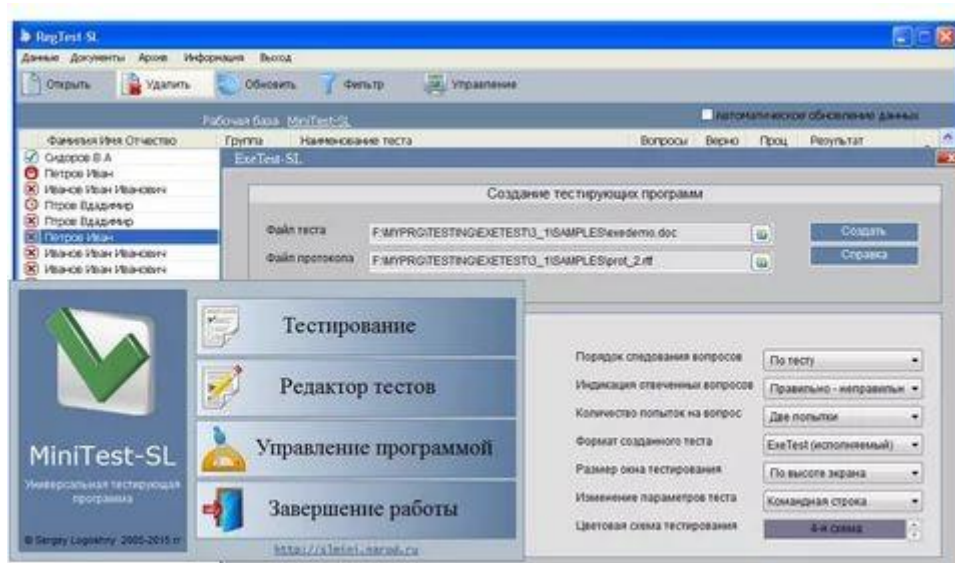


Рис.2. Интерфейс создания прохождения тестов

В приведенных примерах систем тестирования нет ничего, заслуживающего внимания для заимствования. Интерфейс перегружен, гармоничное расположение элементов интерфейса отсутствует, используется больше 5 цветов.

2.11. Формализация привычек и ожиданий пользователей

Методические указания

1. Попытаться отметить наиболее типичные действия пользователей при работе с приложением.
2. Отметить наиболее часто используемые клавиши и комбинации клавиш пользователями.

Практическая часть (Пример)

Работа с указанными выше компонентами является обычной для пользователя, привыкшего работать в среде Windows, так как она осуществляется с помощью стандартных, наиболее часто используемых клавиш «Пробел» и левой кнопки мыши, а также клавиши Tab для перемещения.

Лабораторная работа 3. Проектирование интерфейса

Цель работы

Получить представление о технологии проектирования интерфейса.

Студент должен знать:

- основы проектирования структуры экранов системы;
- основы проектирования навигационной системы;
- основы построения прототипа пользовательского интерфейса;
- методику низкоуровневого проектирования.

Студент должен приобрести навыки:

- проектирования структуры экранов системы;
- проектирования навигационной системы;
- построения прототипа пользовательского интерфейса;
- низкоуровневого проектирования.

Руководство по разработке лабораторной работы 3

Общие сведения

Лабораторная работа 3 является третьей частью лабораторного практикума. Студенты проектируют интерфейс, используя информацию первых двух лабораторных работ. На этом этапе начинается визуализация окон и визуальных элементов. Вначале показывается взаимосвязь окон и соответственно описывается взаимосвязь групп функций. Затем каждое окно наполняется визуальными элементами и описанием взаимосвязи между ними. После этого описывается навигация между окнами или между частями окон.

Процесс разработки

3. Проектирование интерфейса

Методические указания

При проектировании интерфейса кроме словесного описания можно использовать различные виды диаграмм, приспособлен-

ные к пользовательскому интерфейсу. На схеме экранов можно использовать изображения экранов в виде прямоугольников или в виде скругленных прямоугольников. Прямоугольное изображение означает независимый отдельный экран. Изображение в виде скругленного прямоугольника означает часть экрана. На схеме экраны нумеровать, их названия помещать ниже. Типичные структуры сайта (слева) и программы (справа) приведены на рис. 3.

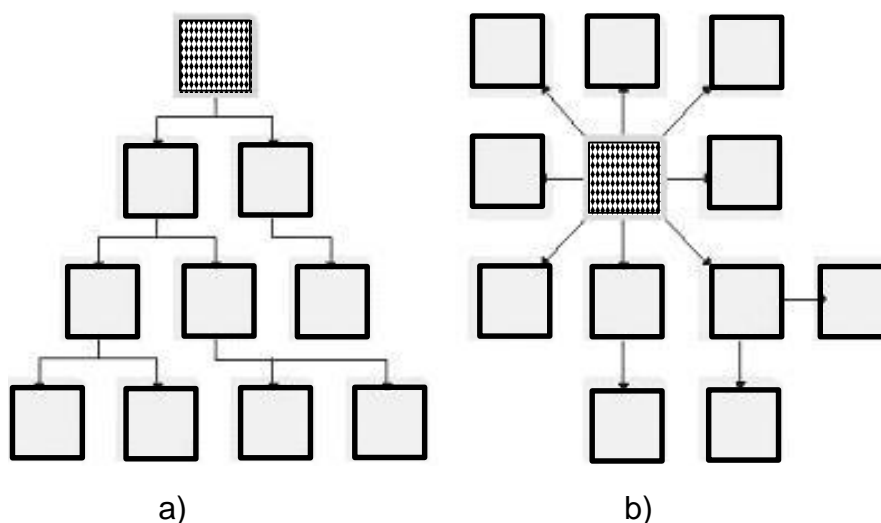


Рис. 3. Типичные структуры экранов: а – для сайтов; б – для программ

Практическая часть (Пример)

Рассматривается следующая легенда: «Создать приложение, реализующее тестирование 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. Элемент теста может формироваться в диалоге или вводиться из текстового файла. В процессе диалога вводится вопрос и варианты ответов. 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.» В ходе ее разработки в основном будут использоваться текстовые описания и различные изображения структур экранов.

3.1. Проектирование структуры экранов системы

Методические указания

1. Определить количество экранов.
2. Определить функциональность каждого экрана.
3. Определить навигационные связи между экранами.
4. Определить и описать логические связи.
5. Определить и описать связи по представлению пользователей.
6. Определить и описать процессуальные связи.

Практическая часть (Пример)

На рис. 4 показана структура экранов для приложения.

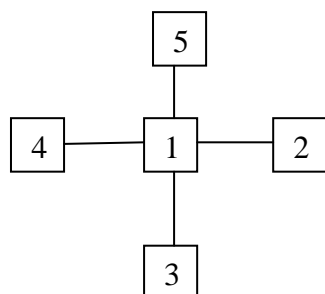


Рис. 4. Структура экранов программы

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

Методические указания

1. Выбирать наиболее адекватную навигационную систему.
2. Разработать детальный интерфейс навигационной системы.

Практическая часть (Пример)

Согласно рис.4 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

Методические указания

1. Определить состав визуальных элементов для конкретных экранов приложения.
2. Разработать взаимное расположение и поддерживающие тексты визуальных элементов.

Практическая часть (Пример)

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

Формирование и тестирование реализуются 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*; по-

сле 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. 5, 6, 7, 8 показаны основной экран, экраны создания теста, экран тестирования.

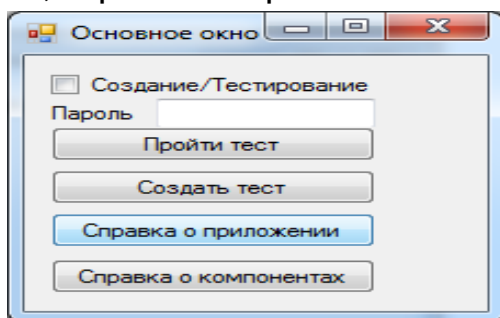


Рис. 5. Основной экран

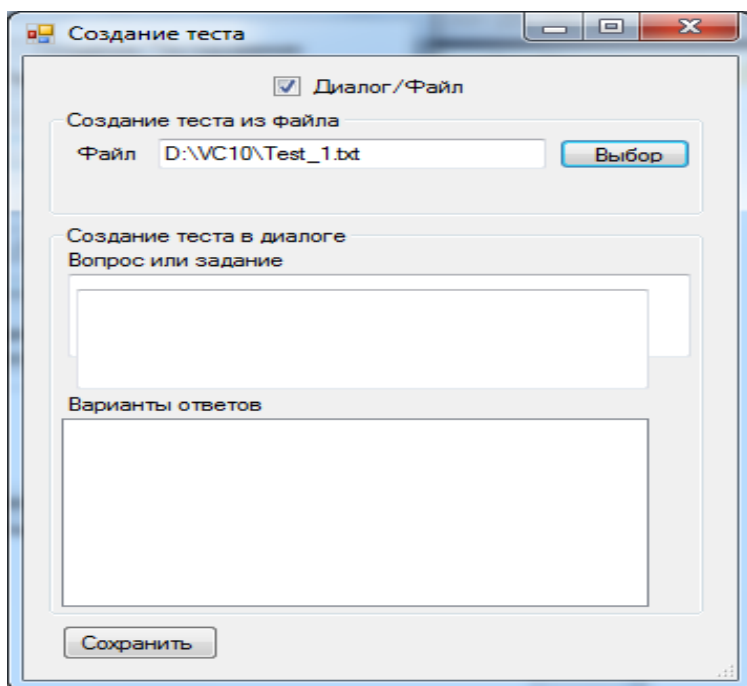


Рис.6. Экран создания теста из файла

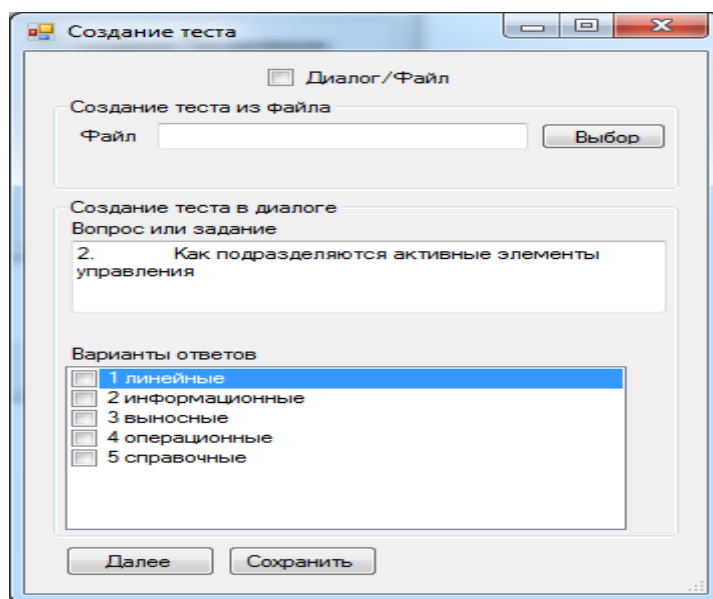


Рис. 7. Экран создания теста с помощью диалога

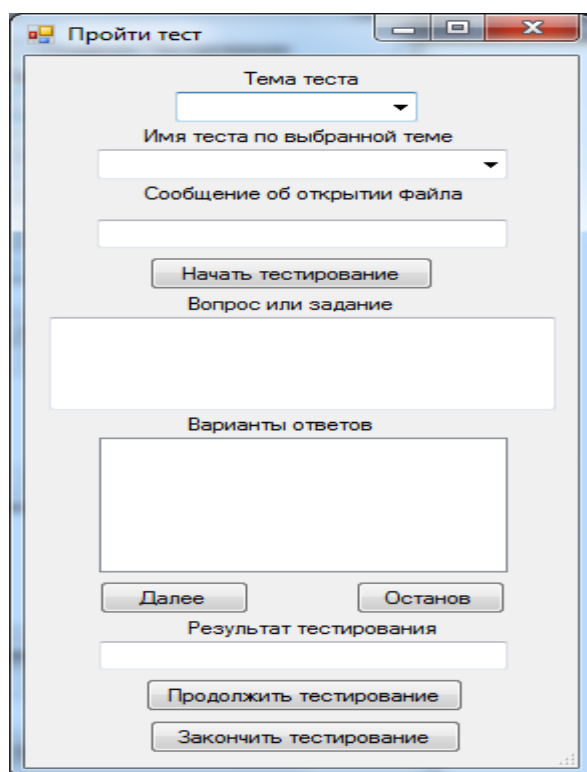


Рис. 8. Экран прохождения теста

Проектирование второстепенных экранов

На рис. 9, 10, 11 показаны вспомогательные экраны, которые выдают информацию по программе и используемым компонентам, представленных в задании.

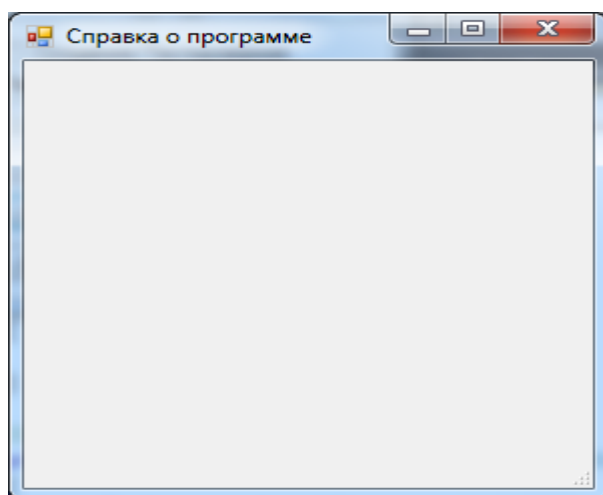


Рис. 9. Экран справки о программе

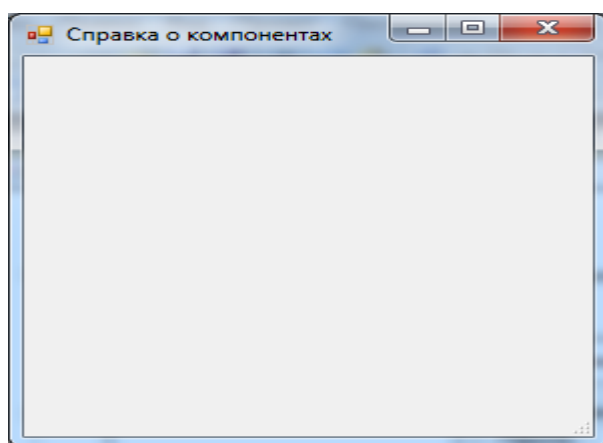


Рис. 10. Экран справки о компонентах

A screenshot of a Windows-style window titled 'Запись теста' (Test Recording). The window has a standard title bar with minimize, maximize, and close buttons. The main content area contains several form elements: a dropdown menu labeled 'Тема теста' (Test Topic), a dropdown menu labeled 'Имя теста по выбранной теме' (Test Name by selected topic), a text input field labeled 'Имя файла' (File Name), a 'Сохранить' (Save) button, a text input field labeled 'Вопрос или задание' (Question or task), and a text input field labeled 'Ответы' (Answers).

Рис. 11. Экран записи теста в файл

Лабораторная работа 4. Тестирование и организация документирования

Цель работы

Приобретение практических навыков тестирования, создания системы помощи и документации.

Студент должен знать:

- основные методы тестирования;
- методы создания системы помощи;
- методы создания системы документации.

Студент должен приобрести навыки (уметь):

- организации тестирования;
- создания системы помощи;
- создания системы документации.

Руководство по разработке лабораторной работы 4

Общие сведения

Краткое содержание первых трех лабораторных работ следующее. Первая лабораторная работа содержит задания и легенду. В задании указывается, какие визуальные элементы разрабатывать, их возможные характеристики и свойства. Легенда описывает решение задачи для некоторой предметной области. При реализации легенды предполагается обязательное использование визуальных элементов задания.

В первой лабораторной работе описываются такие элементы процесса разработки пользовательского интерфейса, как формализация контекста использования; формализация объективных критериев успеха; определение необходимой функциональности системы; анализ целей разработки; анализ действий пользователей.

Во второй лабораторной работе продолжается описание элементов процесса разработки пользовательского интерфейса, таких как постановка задачи; формализация бизнес-ролей пользователей; формализация функциональности; формализация сценариев действий пользователей; обзор интерфейса конкурирующих систем; формализация привычек и ожиданий пользователей.

В третьей лабораторной работе производится проектирование интерфейса. Рассматриваются такие элементы процесса проектирования, как проектирование структуры экранов системы; проектирование навигационной системы; низкоуровневое проектирование.

Лабораторная работа 4 является четвертой частью лабораторного практикума. В этой работе рассматриваются процессы тестирования и создания документации.

4. Тестирование интерфейса

Методические указания

1. Протестировать приложение методом «мыслим вслух».
2. Протестировать приложение методом GOMS, используя две задачи. Значения временных интервалов приведены в табл. 1.

Таблица 1. Значения временных интервалов для метода GOMS

Тип	Действие	Длительность	Комментарии
К	Нажатие на клавишу клавиатуры.	0,28	Включая клавиши Alt, Ctrl, Shift.
М	Нажатие на кнопку мыши.	0,1	
П	Перемещение курсора мыши	1,1	Время зависит как от дистанции, так и от размера цели.
В	Перемещение руки с мыши на клавиатуру и наоборот.	0,4	
Д	Ментальная подготовка	1,2	Время умственной подготовки к следующему шагу.
Р	Время реакции системы	От 0,1 до бесконечности	Для базовых операций (например, при работе с меню) время можно не считать.

Практическая часть (Пример)

Тестирование методом «Мыслим вслух»

Для проведения тестирования выберем сценарий «Создание теста при диалоге».

Вызываем основной экран, показанный на рис. 5. Нажимаем кнопку «Создать тест». Появится экран, представленный на рис. 7. На этом экране CheckBox «Файл/Диалог» не устанавливаем. Это означает, что тест формируется в диалоге.

В поле «Вопрос или задание» набираем вопрос. В поле «Варианты ответов» набираем ответы. Правильные ответы помечаем.

При нажатии на кнопку «Далее» поля «Вопрос или задание», «Варианты ответов» очищаются и можно формировать новый элемент теста. При нажатии на кнопку «Сохранить» происходит процесс записи теста в файл на внешней памяти. Файл сохранится в той папке, в которой хранится приложение. При этом появится окно записи теста в файл, представленное на рис. 11. В этом окне надо выбрать или ввести тему теста, имя теста. При записи в файл показываются все введенные вопросы или задания, а также ответы.

Тестирование методом GOMS

Для проведения тестирования выберем сценарий «Прохождение теста»:

- Выберем на экране иконку «Компьютер»: ДПМ (1.2+1.1+0.1).
- Выберем нужный логический диск: ДПМ (1.2+1.1+0.1).
- Выберем нужную папку: ДПМ (1.2+1.1+0.1).
- Выберем нужный файл: ДПМ (1.2+1.1+0.1).
- Выберем виз. элемент ComboBox: ДПМ (1.2 + 1.1 + 0.1).
- Выберем имя темы: ДПМ (1.2 + 1.1 + 0.1).
- Выберем виз. элемент ComboBox: ДПМ (1.2 + 1.1 + 0.1).
- Выберем имя теста: ДПМ (1.2 + 1.1 + 0.1).
- Время реакции системы на открытие файла: Р (2).
- Выберем кнопку «Начать тестирование»: ДПМ (1.2+1.1+0.1).
- Выдача вопроса или задания: Р (1).

- Выбор ответов: ДМ*5 (1.2+0.1)*5.
 - Выбор кнопки «Далее»: ДПМ (1.2+1.1+0.1).
 - Предыдущие три этапа процесса проходят 10 раз.
 - Вывод результата тестирования: Р(1).
 - Выбор кнопки «Закончить тестирование»: ДПМ (1.2+1.1+0.1).
- Общее время прохождения тестирования составляет 129 с.

5. Система помощи и документации

Методические указания

1. Выбрать один из указанных подходов в пунктах 2 и 3 для организации документации.

2. Выбрать программу для создания системы документации из табл. 2.

Таблица 2. Список средств создания справочной системы

Наименование средства	Назначение	Пояснение
Microsoft Help Workshop	Создается оглавление справочной системы и проект	Формат WinHelp
Help & Manual 6	Создается оглавление справочной системы и проект	Формат CHM
Dr.Explain	Создается оглавление справочной системы и проект	Файл справки или файл помощи в формате CHM - это набор веб-страниц, сжатых и скомпилированных в единый файл.
RoboHelp	Создается оглавление справочной системы и проект	Формат WinHelp 2000
Microsoft HTML Help	Создается оглавление справочной системы и проект	Средство просмотра справочника в формате HTML Help базируется на компонентах Иетернет-браузера Internet Explorer версии 4 и выше
HNW	Создается оглавление справочной системы и проект	Формат CHM
HelpScribble	Создается оглавление справочной системы и проект	В основном программа HelpScribble ориентирована на подготовку справочника WinHelp
AnetHelp Tool	Создается оглавление справочной системы и проект	Программа позволяет получать электронную справку в форматах WinHelp, HTML Help и Java Help
FrameMaker	Создание документов большого объема, имеющих сложную структуру	Можно разделить документ на главы, собрать главы в книгу

3. Для создания системы документации в C# использовать визуальный элемент TreeView или аналогичный визуальный элемент в другой среде.
4. Для создания системы помощи в C# использовать визуальные элементы StatusTip и ContextMenuStrip или аналогичные визуальные элементы в другой среде.

Практическая часть (Пример)

На рис. 12 показан пример использования визуального элемента StatusTip, когда появляется объясняющий текст при наведении на него мышью.

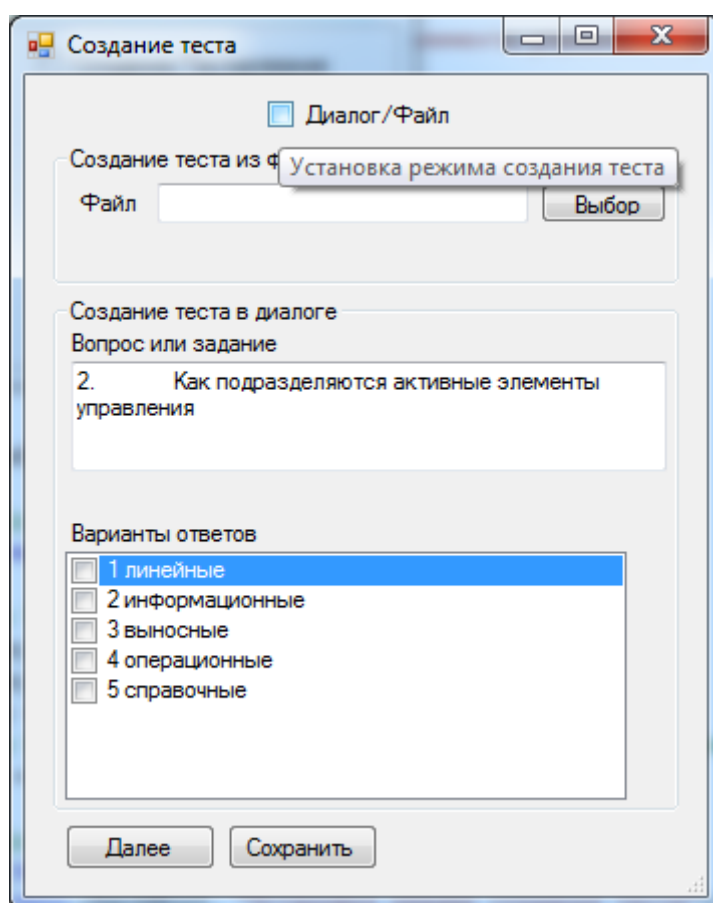


Рис. 12. Пример объяснения элемента при наведении на него курсором

На рис. 13, 14, 15 показана справка о компонентах CheckBox и RadioButton, созданная с помощью визуальных элементов TreeView и TextBox.

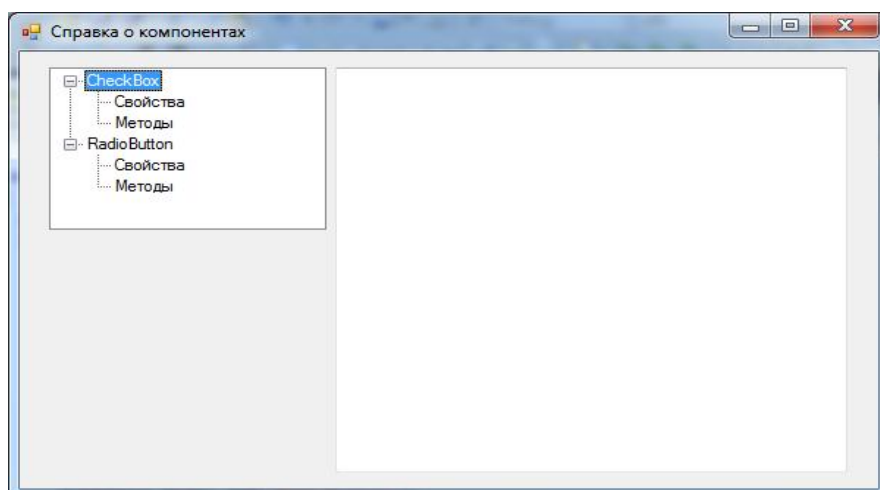


Рис. 13. Содержание справки о компонентах CheckBox и RadioButton

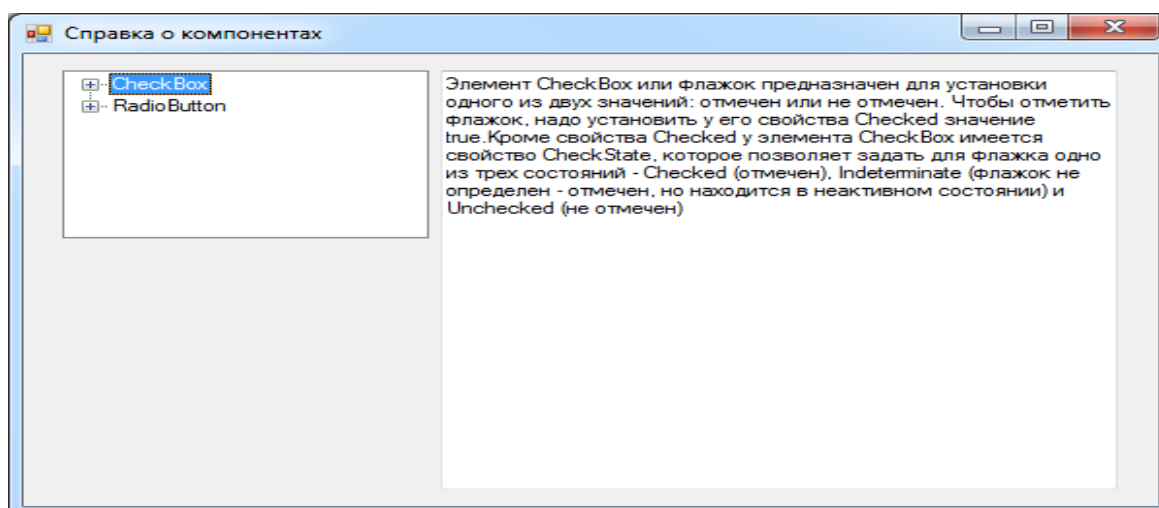


Рис. 14. Краткая справка о компоненте CheckBox

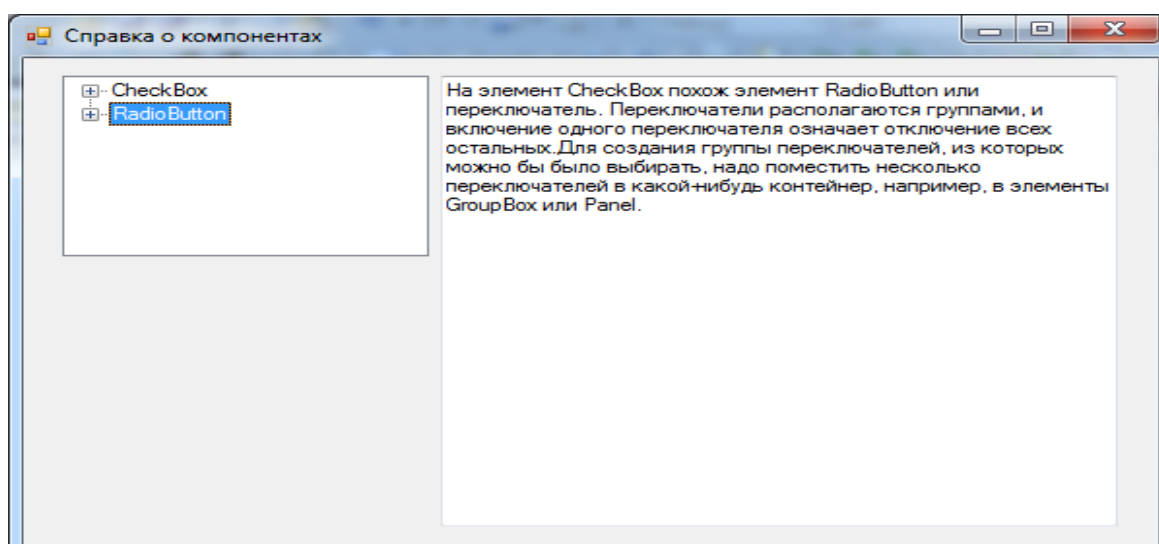


Рис. 15. Краткая справка о компоненте RadioButton

Аналогично создана справка о программе, в которой кратко сообщается о назначении приложения и краткие характеристики.

6. Задания на лабораторные работы

Указание 1. Для заданного визуального элемента сформулировать легенду, т.е. задачу, в которой обязательно должны использоваться визуальные элементы из задания. Принцип разработки следующий: все, что сказано в легенде, обязательно для проектирования. Все, что сказано в задании, обязательно для программирования.

Для лабораторного практикума обязательно запрограммировать два визуальных элемента из легенды. Визуальные элементы, представленные в окне, могут инициировать вызов других визуальных элементов, расположенных в других окнах, которые, в свою очередь, могут инициировать вызов других визуальных элементов. В этом случае получается иерархическая связь между визуальными элементами. Визуальные элементы в этом случае можно представлять вершинами дерева. В любом дереве имеются три типа вершин: корень (начальная вершина, которая имеет несколько выходов), промежуточная вершина (имеет один вход и несколько выходов), конечная вершина (листок, который имеет только один вход). Связь между визуальными элементами может быть представлена в виде нескольких деревьев. Если программируется начальная вершина дерева, то должны быть запрограммированы все визуальные элементы, соответствующие узлам дерева, расположенным по некоторому пути до некоторой конечной вершины. Если программируется некоторая промежуточная вершина *A* дерева, то должны быть запрограммированы все визуальные элементы, соответствующие узлам поддерева с начальной вершиной *A*, расположенным по некоторому пути до некоторой конечной вершины. Если программируется конечная вершина, то должна быть реализована функциональность для визуального элемента, соответствующего этой вершине.

Для контрольной работы обязательно запрограммировать дополнительно еще один визуальный элемент из легенды.

Для задания обязательна справка о компоненте. Для легенды обязательна справка о приложении.

Указание 2. В легенде с помощью некоторых визуальных элементов или сочетаний клавиш предусмотреть для пользователя расширение функциональных возможностей визуальных элементов, указанных в задании.

Например, в задании 1 указаны следующие возможности для визуального элемента Button:

- задание фона кнопки цветовой гаммой RGB;
- задание фона кнопки рисунком, представленным в формате BMP;
- задание вида кнопки 3D;
- задание вида кнопки flat;
- при наведении мышки на кнопку она должна подсвечиваться выбранным цветом;
- при наведении мышки на кнопку должно происходить небольшое ее смещение и отбрасывание тени;
- предусмотреть одновременное применение нескольких совместимых возможностей.

Для того чтобы пользователь мог воспользоваться указанными режимами, надо предусмотреть размещение дополнительных кнопок или поместить внизу на панели пояснения о сочетаниях клавиш, реализующих дополнительные возможности.

В табл. 3 указаны некоторые свойства и методы для реализации указанных ранее возможностей.

Таблица 3. Некоторые свойства и методы, реализующие возможности компонента Button

Класс	Свойство	Метод	Комментарии
Button	BackColor		Получает или задает цвет фона.
	FlatStyle		Получает или задает плоский внешний вид для кнопки.
	ForeColor		Возвращает или задает цвет элемента управления.
	Image		Получает или задает изображение, отображаемое в элементе.
	ResizeRedraw		Возвращает или задает значение перерисовки кнопки.

Табл. 3. Некоторые свойства и методы, реализующие возможности компонента Button (Окончание табл. 3)

Класс	Свойство	Метод	Комментарии
	Size		Возвращает или задает размеры элемента.
	Text		Возвращает или задает текст для элемента.
		BringToFront()	Помещает кнопку в начало Z-порядка.
		DrawToBittmap(Btmp, Rctgl)	Отрисовка элемента.
		GetTopLevel()	Определение нахождения элемента на верхнем уровне.
		Hide()	Скрытие элемента.
		Show()	Отображение элемента.
ColorDialog	FullOpen		True – отображение элемента с расширенными возможностями.
	SoliColorOnly		Выбор между однотонными оттенками цветов.
	Color		Возвращает или задает выбранный цвет.
		ShowDialog()	Запуск диалогового окна.

Задание 1. Разработать визуальный компонент, который представляет собой кнопку. Компонент предоставляет возможность изменения своего внешнего вида путем задания фона кнопки или цветовой гаммой RGB (цветовая заливка), или рисунком, представленным в формате BMP. Кроме того, необходимо предусмотреть возможность задания вида кнопки 3D или flat, в обоих режимах должен поддерживаться “горячий след”, т.е. при наведении мышкой на подпись кнопки, последняя должна или подсвечиваться выбранным цветом, или весь компонент должен выделяться на фоне путем небольшого смещения и отбрасывания тени. Предусмотреть возможность применения нескольких видов выделения одновременно. Компонент должен поддерживать темы ОС Windows.

Задание 2. Разработать визуальный компонент, реализующий элемент «Подпись» (потомок Label). Компонент должен обладать возможностью включения и отключения выделения, т.е. при наведении на него мышкой, компонент или должен выделяться простым подсвечиванием, выбранным цветом, или должен быть обведен рамкой с выбираемыми параметрами (толщина

обводки, цвет рамки, цвет внутренней заливки, прозрачность заливки), или выделен путем небольшого смещения и отбрасывания тени. Предусмотреть возможность применения нескольких видов выделения одновременно.

Задание 3. Разработать визуальный компонент, реализующий элемент "TextBox". Компонент должен обладать возможностью набора одной строки или нескольких строк, возможностью менять цвет символов текста и фона, и возможностью изменения шрифта и его размера.

Задание 4. Разработать визуальный компонент, реализующий элемент "ListBox". Компонент должен обладать возможностью добавления, удаления и выделения элемента и изменения цвета текста и фона.

Задание 5. Разработать визуальный компонент, реализующий элемент "ComboBox". Компонент должен обладать возможностью добавления, удаления и выделения элемента, изменения цвета текста и фона.

Задание 6. Разработать визуальный компонент, реализующий элемент "CheckedListBox". Компонент должен обладать возможностью добавления, удаления и выделения элемента, изменения цвета текста и фона.

Задание 7. Разработать визуальный компонент, реализующий элемент "ListView". Компонент должен обладать возможностью добавления, удаления и выделения элемента, изменения цвета текста и фона.

Задание 8. Разработать визуальный компонент, реализующий элемент "TreeView". Компонент должен обладать возможностью добавления, удаления и выделения элемента (узла, подузла), изменения цвета текста и фона.

Задание 9. Разработать визуальный компонент, реализующий элемент управления "ProgressBar". Компонент должен позволять изменять свой внешний вид путем задания цвета для каждого элемента компонента (шкала, насечки на шкале, подписи к насечкам, собственно ползунков) или путем задания внешнего вида каждого элемента компонента из файла. Продумать возможность градиентной заливки области перемещения ползунка от на-

чальной отметки до текущего положения (начальный и конечный цвета задаются).

Задание 10. Разработать визуальный элемент, реализующий элемент управления «WebBrowser». Компонент должен переходить к определенному документу по URL. Компонент должен выдавать заголовок и текстовое содержание документа в случае подключения к Интернету, прокрутку документа, а также изменение цвета просмотренного документа.

Задание 11. Разработать визуальный компонент, реализующий элемент "ToolStrip". Компонент должен обладать возможностью добавления и выделения элемента, изменения цвета текста и фона.

Задание 12. Разработать визуальный компонент, реализующий элемент "MenuStrip". Компонент должен обладать возможностью добавления и выделения элемента, изменения цвета текста и фона.

Задание 13. Разработать визуальный компонент, использующий элемент "ContextMenuStrip", привязанный к элементу "TextBox". Компонент должен обладать возможностью добавления и выделения элемента функциональности. Компонент должен обладать возможностью изменения цвета текста и фона и цвета выделения элемента функциональности.

Задание 14. Разработать визуальный компонент, использующий элемент "ContextMenuStrip", привязанный к элементу "Button". Компонент должен обладать возможностью добавления и выделения элемента функциональности, изменения цвета текста и фона и цвета выделения элемента функциональности.

Задание 15. Разработать визуальный компонент, использующий элемент "ContextMenuStrip", привязанный к элементу строки элемента "ListBox". Компонент должен обладать возможностью добавления и выделения элемента функциональности, изменения цвета текста и фона и цвета выделения элемента функциональности.

Список рекомендуемой литературы

1. **Головач, В.** Дизайн пользовательского интерфейса.
<http://www.uibook1.ru/>
2. **Денисов, Ю.А.** Основы информационных систем, вып. I (11),
http://www.citforum.ru_80/operating_systems/ois/a.shtml
3. **Программирование** на Visual C++,
<http://vnews.uka.ru/html/articles.htm>
4. **Official** Guidelines for User Interface Developers and Designers, <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnwue/html/welcome.asp>
5. **Macintosh** Human Interface Guidelines,
<http://developer.apple.com/techpubs/mac/HIGuidelines/HIGuidelines-2.html#avail1-0>
6. **X Window** System technology and standards,
<http://www.x.org/about.htm>
7. **Нильсен, Я.** WEB-дизайн, / **Я. Нильсен**, М.: Символ-Плюс, 2002.
8. **Кирсанов, Д.** Веб-дизайн, / **Д. Кирсанов**, М.: Символ-Плюс, 2002.
9. **Раскин, Д.** Интерфейс: новые направления в проектировании компьютерных систем, / **Д. Раскин**, М.: Символ-Плюс, 2003.
10. **Круг, С.** Не заставляйте меня думать!, / **С.Круг, Р. Блэк**, М.: Символ-Плюс, 2001.
11. **Соло, Р.** Когнитивная психология, / **Р. Соло**, СПб: Питер, 2002.
12. **Тоффлер, Э.** Метаморфозы власти, / **Э. Тоффлер**, М.: АСТ, 2001.
13. **Скопин, И.Н.** Разработка интерфейсов программных систем. Системная информатика, В. 6. / **И.Н. Скопин**, Новосибирск: Наука, 1998.
14. **Проектирование** пользовательского интерфейса на персональных компьютерах. Стандарт фирмы IBM. -Вильнюс: DBS Ltd, 1992.
15. **Мандел, Т.** Разработка пользовательского интерфейса, / **Т. Мандел**, М.: ДМК Пресс, 2001.

16. **Дениг, В.** Диалоговые системы "Человек-ЭВМ". Адаптация к требованиям пользователя, / **В. Дениг, Г. Эссиг, С. Маас**, М.: Мир, 1984.
17. **Алексееenko, Е.А.** Оценка качества пользовательского интерфейса, / / **Е.А. Алексееenko, Е.В. Гавриленко**, Управляющие системы и машины. - 2000.- № 2.
18. **Гультияев, А.К.** Проектирование и дизайн пользовательского интерфейса, / **А.К. Гультияев, В.А. Мишин**, СПб.: КОРОНА-принт, 2000.

ПРИЛОЖЕНИЕ

Структура отчета

Все пункты и подпункты отчета рассматриваются сначала с точки зрения легенды, а затем с точки зрения задания.

Структура отчета по лабораторному практикуму: титул, оглавление, содержание, приложение.

Оглавление содержит следующие пункты: описание предметной области, постановка задачи, проектирование интерфейса, тестирование интерфейса, система помощи и документация.

Описание предметной области относится к легенде, для которой рассматриваются основные понятия и ситуации в данной области.

Пункты могут содержать несколько подпунктов. Рассмотрим состав подпунктов для некоторых пунктов.

2. Постановка задачи

2.1. Формализация контекста использования

и т.д. далее следуют названия пунктов, соответствующие первой и второй лабораторной работе.

3. Проектирование интерфейса

3.1. Проектирование структуры экранов системы

3.2. Проектирование навигационной системы

3.3. Построение прототипа ПИ

3.4. Низкоуровневое проектирование.

Свойства компонент, указанные в задании, должны быть доступны пользователю либо в виде нажатия каких-то клавиш, либо в виде приведенного дополнительного или контекстного меню, либо еще каким-то образом.

Отчет по лабораторному практикуму формируется наращиванием количества страниц в отчете путем добавления сделанных лабораторных работ. Студентам групп 3-41 и 3-42 можно вместо лабораторной работы выполнять задание по выбранной теме для дисциплин «Конструирование программного обеспечения» или «Разработка мобильного приложения».

Завершением лабораторного практикума является разработанные отчет и программа.

СОДЕРЖАНИЕ

Введение	3
Лабораторная работа 1. Постановка задачи (первая часть)	5
Общие сведения	5
2. Постановка задачи	6
2.1. Формализация контекста использования	7
2.2. Формализация объективных критериев успеха	8
2.3. Определение необходимой функциональности системы	9
2.4. Анализ целей	10
2.5. Анализ действий пользователей	11
Лабораторная работа 2. Постановка задачи (вторая часть)	12
Общие сведения	12
2.6. Постановка задачи	13
2.7. Формализация бизнес-ролей пользователей	13
2.8. Формализация функциональности	14
2.9. Формализация сценариев действий пользователей	15
2.10. Обзор интерфейса конкурирующих систем	16
2.11. Формализация привычек и ожиданий пользователей	17
Лабораторная работа 3. Проектирование интерфейса	18
Методические указания	18
3. Проектирование интерфейса	18
3.1. Проектирование структуры экранов системы	20
3.2. Проектирование навигационной системы	21
3.3. Низкоуровневое проектирование	21
Лабораторная работа 4. Тестирование и организация докумен- тирования	25
Общие сведения	25
4. Тестирование интерфейса	26
5. Система помощи и документации	28
6. Задания на лабораторные работы	31
Список рекомендуемой литературы	36
Приложение	38

ЗУБКОВ Валентин Петрович

Проектирование интерфейса компьютерных комплексов

Методические указания

Редактор Н.Б. Михалева

Подписано в печать Формат 60×84 $\frac{1}{16}$.

Печать плоская. Усл. печ. л. 2,32..

Тираж 50 экз. Заказ №

ФГБОУВПО «Ивановский государственный энергетический университет имени В.И. Ленина».

Отпечатано в УИУНЛ ИГЭУ

153003, г. Иваново, ул. Рабфаковская, 34.