**Государственное бюджетное профессиональное образовательное учреждение города Москвы**

**Московский государственный колледж электромеханики и информационных технологий**

Проектирование функциональной подсистемы организации защиты курсовых проектов.

Пояснительная записка

к курсовому проекту по МДК 02.01

“ Информационные технологии и платформы

разработки информационных систем ”

Руководитель проекта ( Т.М. Босенко )

Студент ( )

2015

Содержание

Введение 3

1 Постановка задачи 5

1.1 Организационная сущность задачи 5

1.2 Информационная модель 5

1.3 Входные данные 6

1.4 Выходные данные 6

2 Вычислительная система 8

2.1 Используемые технические средства 8

2.2 Инструменты разработки 8

3 Проектирование задачи 9

3.1 Алгоритм решения 9

3.2 Концептуальный прототип 9

3.3 Функции: логическая и физическая организация и элементы управления 11

4 Описание программного средства 12

4.1 Общие сведения: 12

4.2 Функциональное назначение 12

5 Методика испытаний 13

5.1 Технические требования 13

5.2 Порядок проведения испытаний 13

5.2.1 Функциональное тестирование 13

5.2.2 Полное тестирование 13

6 Применение 15

6.1 Назначение программы 15

6.2 Условия применения 15

6.3 Справочная система 15

Заключение 17

Литература 18

Приложение А Текст программы 19

Приложение Б Схема работы 21

Приложение В Тестовые данные 24

Введение

Целью курсового проекта является проектирование функциональной подсистемы организации защиты курсовых проектов. Данная программа предназначена для упрощения организации проведения защиты курсовых проектов.

 Для достижения цели курсового проекта нужно решить следующие задачи:

* выполнить объектно-ориентированный анализ и проектирование системы, результатом которой будет модель системы;
* определить вычислительную систему, необходимую для создания программного продукта;
* по модели выполнить проектирование задачи;
* разработать программный продукт;
* описать созданный программный продукт;
* выбрать методику испытаний;
* описать процесс тестирования;
* привести примеры области применения.

Решение поставленных задач отражено в пояснительной записке.

Пояснительная записка состоит из 6 разделов и содержит необходимую и достаточную информацию по использованию данного программного продукта.

В первом разделе «Постановка задачи» описывается исследование предметной области, определение круга задач, которые должны быть автоматизированы или модифицированы. Описывается задача, перечисляются основные функции программы (обоснование необходимости компьютерной обработки информации или процессов, как задача будет упрощать или усложнять решение поставленной задачи), приводится обзор существующих аналогов программы с указанием их достоинств и недостатков.

Во втором разделе «Вычислительная система» перечисляются требования к аппаратному обеспечению и конфигурации компьютера, проводится характеристика операционной системы, обоснование выбранной среды для разработки приложения. Описываются новые возможности программы, а также ее отличия от предыдущих версий.

В третьем разделе «Проектирование задачи» проводится объектно-ориентированный анализ задачи, строится алгоритм в виде математической модели или в виде блок-схемы. строится концептуальный прототип системы меню, диалоговых окон и элементов управления в рамках среды разработки приложения.

В четвертом разделе «Описание программного средства» содержится информация об обозначении и наименовании приложения, о программном обеспечении, необходимом для функционирования программы, о классах решаемых задач и назначении программы.

В пятом разделе «Методика испытаний» описываются требования к техническим средствам для проведения испытаний, требования к характеристикам программы применительно к условиям эксплуатации, требования к информационной и программной совместимости. Описывается проверка каждого пункта меню, каждой операции, которая выполняет приложение. Моделируются все возможные действия пользователя при работе с программой. Представляются результаты тестирования.

Шестой раздел «Применение» предназначен для описания сведений о назначении программного средства, классе решаемых задач, ограничениях, накладываемых на область применения, требования к необходимым для данного программного средства техническим и программным ресурсам, структуре справочной системы, методике использования справочной системы.и области его применения.

В заключении будет проанализировано выполнение поставленной задачи и выполненной работы, степень ее соответствия проектным решениям, причины несоответствия.

Приложения будут содержать тексты программных модулей, результаты работы приложения.

В графической части будут представлены схема работы программы, схема работы разработанного алгоритма «Тестирование пользователя», экранные формы.

1 Постановка задачи

1.1 Организационная сущность задачи

Требуется разработать программу, которая будет организовывать автоматизацию процесса защиты курсовых проектов студентов в учебном заведении. Защита курсового проекта будет основываться на таких входных данных: «ФИО претендента», «Учебная Группа», «Направление обучения», «Пометка «Красный диплом», «Название доклада», «Форма обучения(очная\заочная)», «Оценка защиты», «степень готовности работы(готов \не готов)».

1.2 Информационная модель

Согласно общим требованиям стандартный графический интерфейс разрабатываемого приложения должен:

– содержать привычные и понятные пользователю пункты меню, соответствующие функциям обработки;

– сохранять стандартное назначение и местоположение на экране графических объектов, работающих в среде Windows.

Большинство программных приложений работают в диалоговом режиме. Диалоговый режим – это способ взаимодействия пользователя с программой, при котором происходит непосредственный и двухсторонний обмен информацией, командами или инструкциями между человеком и программой. Диалоговый режим подразумевает такую скорость обработки данных, которая не сказывается на технологии действий пользователя. Различают активные и пассивные диалоговые режимы.

Состав для диалоговых систем представляет собой:

– главное меню, где пользователю нужно ввести основные данные предназначенные для начисления зарплаты.

– действия запрос-ответ, где предлагается перечень возможных выбираемых работников(кол-во которых не ограничено и изменчиво).

 В этом главном меню перед пользователем представляется ряд блоков(вводимых и выводимых) таких как: «ФИО претендента», «Учебная Группа», «Направление обучения», «Пометка «Красный диплом», «Название доклада», «Форма обучения(очная\заочная)», «Оценка защиты», «степень готовности работы(готов \не готов)». При работе с программным продуктом не должно быть трудностей, а созданная заранее справочная система должна помочь в использовании программы.

1.3 Входные данные

Входными данными являются:

– текстовые файлы уровней в формате \*.ssl, содержащие в себе информацию, необходимую для внесения данных по каждому студенту.

– динамически подключаемая библиотека, содержащая в себе файлы изображений(см. в таблице 1).

|  |  |
| --- | --- |
| Функция | Входные данные |
| Добавление | «ФИО претендента», «Учебная Группа», «Направление обучения», «Пометка «Красный диплом», «Название доклада», «Форма обучения(очная\заочная)», «Оценка защиты», «степень готовности работы(готов \не готов)» |

Таблица 1

1.4 Выходные данные

Выходные данные будут сведены в таблицы(см. в таблице 2).

Таблица 2.

|  |  |
| --- | --- |
| Функция | Выходные данные |
| Вывод | ФИО Учебная ГруппаНаправление обучения Пометка «Красный дипломНазвание доклада Форма обучения(очная\заочная)Оценка защиты |

2 Вычислительная система

2.1 Используемые технические средства

Конфигурация компьютера, на котором будет разрабатываться программное приложение:

* Intel Core i7-3960X Extreme Edition Sandy Bridge-E 3300 МГц;
* оперативная память 16gb 2400 МГц;
* жёсткий диск HDD 2Тб.
* -
* -
* -

2.2 Инструменты разработки

Инструментами разработки будут являться:

* операционная система Windows7;
* среда программирования Visual Studio 2010;
* язык программирования C#.

Исходя из результатов объектно-ориентированного анализа и проектирования (ООАП), итогов исследования предметной области, можно сделать заключение, что наиболее подходящей средой разработки программного приложения является среда программирования Visual Studio 2010(WCF).

В настоящее время программирование бурно развивается, как с точки зрения расширения круга решаемых им задач, так и с точки зрения существенного усложнения используемых в программировании технологий Причем особо необходимо отметить немалые размеры разрабатываемы программных продуктов. Все это требует максимального упрощения и ускорения процесса разработки приложений и использования ранее реализованных программных фрагментов.

Такие требования к современному программированию привели к созданию многочисленных RAD-систем (от англ. RAD — Rapid Application Development — быстрая разработка приложений), представляющих собой интегрированные среды разработчика, включающие в себя:

* средства быстрого и удобного построения программ, в том числе визуального;
* встроенные компиляторы и отладчики;
* системы коллективной разработки проектов.

Одной из таких RAD-систем является Visual Studio 2010. Итак, Visual Studio 2010 — это объектно-ориентированная среда для визуального проектирования Windows приложений с развитыми механизмами повторного использования программного кода. Суть модели заключается в поддержке системой постоянно расширяемого набора объектных компонентов, из которых и строится программа. Компоненты в Visual Studio 2010 просты для использования и развития, как результат сокрытия значительной части той структуры программы, которая близка к взаимодействию с операционной системой. Таким образом, для создания в Visual Studio 2010 несложных программных продуктов совершенно не обязательно понимать внутреннюю структуру Windows-приложения, получаемого после разработки в Visual Studio 2010. Достаточно просто уметь работать с некоторыми компонентами, поставляемыми вместе со средой разработчика. При этом начать работу со средой можно практически без предварительного ознакомления, а написание первого приложения не потребует углубления в особенности системы. Этому отчасти способствует удобный интерфейс среды разработчика, не перегруженный излишними вопросами к разработчику.

Однако такой подход совершенно неприемлем для серьезного программирования, и, рано или поздно, придется освоить и основы программирования под ОС Windows, и серьезно изучить саму среду разработки, а также возможности, которые она предоставляет. Кроме того, конечно же, для создания качественных программных продуктов необходимо глубокое понимание компонентной модели.

Проект - это программа на стадии создания. Он состоит из программного кода, файлов проекта Visual Studio 2010 и редактируемых (-ой) форм (ы). При открытии Visual Studio 2010 проект создаётся автоматически, а так же его можно создать вручную:

Меню главного окна Visual Studio 2010, далее:

Файл->Создать->Другое (File->New->Other)

Опишу некоторые пункты:

* "Application" - создаёт новый проект Windows-приложения
* "Unit" - создаёт так называемый файл представления. Этот файл является описанием типов, так же содержит описание методов, событий и свойств, которые находятся в элементе управления (объекте). Содержимое данного файла написано на языке С#;
* "Form" - добавляет новую форму будущего Windows-приложения;
* "Frame" - фрейм, используется, грубо говоря, для внедрения формы в форму;
* "Text" - создаёт текстовый файл;
* "DLL Wizard" - создаёт проект DLL (Dynamic-link library — динамически подключаемая библиотека);
* "Console Application" - этот пункт создаёт консольное приложение.

Таблица содержит данные по определенной теме, например, сведения о сотрудниках или товарах. Каждая запись в таблице включает данные об одном элементе, например о конкретном сотруднике. Запись состоит из полей и включает такие сведения, как имя, адрес и телефонный номер. Кроме того, запись обычно называется строкой, а поле – столбцом.

База данных может включать множество таблиц, в которых хранятся данные по различным темам. Каждая таблица может состоять из множества полей различного типа, включая текст, числа, даты и рисунки.

3. Проектирование задачи

3.1 Объектно-ориентированный анализ

В проектируемой системе используются стандартные классы: TForm, TLabel, TMainMenu, TMenuItem, TMemo.

Компонент TForm представляет собой главное окно программы, предназначенное для размещения остальных компонентов, и организации диалога с пользователем.

Компоненты класса TLabel предназначены для размещения на форме различного рода текстовых надписей.

Компонент TImage класса TImage служит для размещения на форме одного из трех поддерживаемых Delphi типов изображений: растрового рисунка (\*.bmp), значка (\*.ico) или метафайла (\*.wmf или \*.emf).

Компонент класса TMainMenu определяет главное меню формы. На форму можно поместить сколько угодно объектов этого класса, но отображаться в полосе меню в верхней части формы будет только тот из них, который указан в свойстве Menu формы. Каждый элемент меню является объектом класса TMenuItem.

Компонент TMemo многострочное редактируемое текстовое поле. Предназначено для ввода, редактирования и отображения достаточно длинного многострочного текста.

3.2 Концептуальный прототип

Концептуальный прототип состоит из описания внешнего пользовательского интерфейса – системы меню, диалоговых окон и элементов управления.

При создании приложения важную роль играют формы, так как они являются основным диалоговым средством работы пользователя. Разрабатываемое приложение будет малооконное. Оно будет содержать 3 формы, в приложении предусмотрено меню.

Первая форма это главное меню, состоящее из вносимых пунктов.На ней же,и будет выводится нужная нам информация. .

Таблица 1 – Структура главного меню

|  |  |
| --- | --- |
| Название | Действие |
| Справка | Вызов подменю |
| О программе  | Открывает справочную систему |
| Выход | Закрывает программу |
|  |  |

Таблица 2 – Структура формы главного меню

|  |  |
| --- | --- |
| Название | Действие |
| Кнопка: Австрия  | Открывает форму об Австрии. |
| Кнопка:Зальцбург | Открывает форму о Зальцбурге. |

3.3 Функции: логическая и физическая организация и элементы управления

Рассмотрим основные функции программы. В программном средстве «Программа создания путеводителя: Австрия – Зальцбург» для обработки событий на формах используются следующие методы:

- procedure TForm1.Button1Click(Sender: TObject); - процедура для перехода на форму «Австрия»;

- procedure TForm1.Button2Click(Sender: TObject); - процедура для перехода на форму «Зальцбург»;

- procedure TForm1.N2Click(Sender: TObject); - процедура выхода из программы;

- procedure TForm1.N3Click(Sender: TObject); - процедура предназначенная для открытия справочной системы.

4 Описание программного средства

4.1 Общие сведения

Программный продукт представляет собой самостоятельно разрабатываемый файл «Зальцбург.exe». Его объем занимает 30,5 Мбайт. Для начала необходимо запустить программу, после на форме выбрать то, что требуется пользователю (Австрия, Зальцбург). Программный продукт функционирует в среде Windows, и никакие дополнительные библиотеки не требуются.

4.2 Функциональное назначение

Главным назначением программы является получение знаний пользователем по теме: Австрия - Зальцбург. В программном продукте «Зальцбург.exe» реализованы функции, которые были определены на этапе объектно-ориентированного программирования и анализа.

Средства защиты, разграничения доступа к данным и сетевая поддержка не предусмотрены.

5 Методика испытаний

5.1 Технические требования

Для полноценного функционирования разработанного программного продукта, качественного выполнения всех поставленных задач, требуется ПК следующей конфигурации:

* Intel Core i7-3960X Extreme Edition Sandy Bridge-E 3300 МГц;
* оперативная память 16gb 2400 МГц;
* жёсткий диск HDD 2Тб.
* клавиатура, совместимая с ПК;
* мышь, совместимая с ПК.

5.2 Порядок проведения испытаний

5.2.1 Функциональное тестирование

При функциональном тестировании осуществляется проверка каждого пункта меню, каждой операции, с целью проверки выполнения всех функций, определенных на этапе объектно-ориентированного анализа и проектирования. Функциональное тестирование должно гарантировать работу всех элементов управления в автономном режиме.

Протестируем функцию …

5.2.2 Полное тестирование

Полное тестирование выполняется после успешного завершения функционального тестирования, продемонстрировавшего корректное выполнение всех функций, определенных на этапе проектирования.

Полное тестирование должно показать работу программного продукта с момента запуска до завершения с иллюстрацией реакций программы на некорректные действия пользователя.

Полное тестирование выполняется с применением тех же тестовых данных, что и для функционального тестирования. Последовательность действий по процессу полного тестирования приведена в таблице 2.

Таблица 2 – Алгоритм полного тестирования

|  |  |  |
| --- | --- | --- |
| Этап тестирования | Ожидаемый результат | Иллюстрация полученного результата |
| Запуск приложения «.exe» | Активирование главной формы «.exe» | Рисунок 2 |
| Работа с теоретическим материалом | Выбор темы: «». | Рисунок 3 |
| Выбор темы: «Зальцбург». | Рисунок 4 |
| Вывод справочной информации | Нажатие кнопки «Справка». Выводит справочную информацию на экран | Рисунок 5 |

6 Применение

6.1 Назначение программы

Разработанное программное средство «.exe», предназначено для получения знаний пользователем : «».

Программное средство обеспечено справочной системой, которая поможет пользователю в поиске или получении справочной информации. Программное средство обладает необходимым объемом информации для реализации поставленной задачи.

6.2 Условия применения

Необходимо отметить, что данное программное средство не обладает средствами, ограничивающими доступ к нему, а также средствами, препятствующими свободному распространения программы. Для работы с приложением не требуется установка дополнительных компонентов и специальных технических и программных ресурсов.

6.3 Справочная система

Справочная система для данного программного средства довольно проста. Она представляет собой описание возможных действий для работы с программой. Вся справочная информация расположена на кнопке «Справка» в главном меню форм и представлена на рисунке 5

Заключение

В рамках курсового проектирования по предмету «Информационные технологии и платформы разработки информационных систем» было разработано программное средство «.exe», позволяющее проводить защиту курсового проекта студентами. Программное средство имеет ряд достоинств: простой и понятный интерфейс, небольшой объем памяти, занимаемый приложением на различных носителях информации, удобно для быстрой обработки информации.

В программе реализованы такие задачи, как:

* Выполнен объектно-ориентированный анализ и проектирование системы, результатом которой является модель системы;
* Определена вычислительная система, необходимая для создания программного средства;
* По модели выполнено проектирование задачи;
* Разработано программное средство;
* Описано созданное программное средство;
* Выбрана методика испытаний;
* Описан процесс тестирования;
* Приведены примеры области применения.

В процессе курсового проектирования программный продукт прошел функциональное и полное тестирование, которые доказали корректность выполнения всех определенных функций, отсутствие сбоев и неадекватных реакций со стороны программы..