

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И МОЛОДЕЖНОЙ ПОЛИТИКИ
СВЕРДЛОВСКОЙ ОБЛАСТИ
ГАПОУ СО «ЕКАТЕРИНБУРГСКИЙ МОНТАЖНЫЙ КОЛЛЕДЖ»

ТЕХНИЧЕСКОЕ ЗАДАНИЕ
на курсовую работу по дисциплине «МДК 05.03 Тестирование информационных систем»

Тема курсовой работы: **«Создание и тестирование АРМ для обработки информации об участии студентов в олимпиадном движении, конкурсах профессионального мастерства»**

Выдано **Казанцеву Денису Сергеевичу** студенту группы **ИП-11**
Руководитель проектирования: Грибова А.В.

1. ВИД АВТОМАТИЗИРУЕМОЙ ДЕЯТЕЛЬНОСТИ

Приложение предназначено для ввода, хранения, обработки информации об участии студентов в олимпиадном движении, конкурсах профессионального мастерства.

Цели приложения:

- систематизация и информирование общественности о том, каких результатов добились студенты во время участия в олимпиадном движении, конкурсах профессионального мастерства;
- Обеспечение механизма формирования отчетности по олимпиадам.

Приложение должно обеспечивать работу следующих категорий пользователей: незарегистрированный пользователь, администратор, преподаватель.

Незарегистрированные пользователи могут:

- просматривать список всех олимпиад;
- фильтровать олимпиады по году;
- просматривать информацию о победителях олимпиад (протоколов);
- просматривать положение по олимпиаде;
- просматривать задания прошлых лет.

Зарегистрированные пользователи могут:

- записаться на любую из олимпиад (или на несколько олимпиад);
- просматривать свои результаты участия в олимпиадах;
- редактировать свой профиль

Функционал администратора:

- создание учетных записей преподавателей;
 - редактирования существующих учетных записей;
 - заполнение карточек олимпиад (название олимпиады, ответственный преподаватель, даты проведения олимпиады);
 - просмотр протоколов олимпиад (название олимпиады, статус (формируется, подготовлен));
 - публикацию готовых протоколов по олимпиаде (тех, у которых статус «подготовлен»).
- После публикации протоколы становятся видимыми незарегистрированным пользователям.

Администратор может формировать следующие виды отчетов:

- протокол по выбранной олимпиаде (Word/Excel.);

- годовой отчет участия в олимпиадном движении с результатами.

Преподаватель отвечает за:

- создание и прикрепление положения по олимпиаде;
- создание учетных записей студентов (ФИО, дата рождения, электронная почта, логин, пароль, учебное заведение, уровень образования, курс обучения, специальность);
- добавление, редактирование «протоколов» об участии студентов в олимпиадном движении, конкурсах профессионального мастерства (указывается количество баллов, полученных участниками);
- формирование протокола олимпиады в зоне своей ответственности (протокол должен выдавать количество баллов, полученных участниками, и результаты (призер, победитель, участник)) в формате Word/Excel.
- отзыв готовых протоколов, если в них необходимо внести изменения (отозвать можно только те протоколы, у которых статус «подготовлен»);
- добавлять и изменять бланки сертификатов, грамот за участие (информационная система должна предусматривать возможность добавления любого количества грамот к одному мероприятию);

2. ТРЕБОВАНИЯ К ТЕХНИЧЕСКОЙ РЕАЛИЗАЦИИ ПРОЕКТА

Требования к разработанному программному продукту:

1. Среда разработки
 - 1.1. В качестве системы управления данными (СУБД) использовать СУБД MS SQL SERVER;
 - 1.2. В качестве среды программирования использовать MS Visual Studio;
 - 1.3. Для построения клиентских приложений Windows необходимо использовать платформу Windows Presentation Foundation (WPF) в составе .NET Framework
2. Название приложения
 - 2.1. Необходимо использовать соответствующие названия для ваших приложений и файлов. Так, например, наименование настольного приложения должно обязательно включать название компании – заказчика
3. Файловая структура
 - 3.1. Файловая структура проекта должна отражать логику, заложенную в приложение. Например, все формы содержатся в одной директории, пользовательские визуальные компоненты – в другой, классы сущностей – в третьей.
 - 3.2. Каждая сущность должна быть представлена в программе как минимум одним отдельным классом;
 - 3.3. Проект должен иметь структуру (папки проекта), реализованную в концепции MVC
 - Модель (папка Model) предоставляет классы, сформированные фреймворком Entity Framework. Количество классов в модели ДОЛЖНО СОВПАДАТЬ с количеством таблиц в БД. Если для подключения EntityFramework вы используете отдельный класс (Например, класс Core), то рекомендуется разместить его в папке Model. Если возникает необходимость добавления новых свойств классам из модели, созданной Entity Framework, то для этих целей используйте папку Model/PartialClasses
 - Представление (папка View) отвечает за отображение данных, то есть в ней необходимо расположить все страницы, используемые в проекте
 - Контроллер (папка Controllers) описывает все методы, применяемые к классам Модели. Контроллер взаимодействует с БД (Model), а также формирует данные для вывода на страницы (View). Количество классов в папке Controllers ДОЛЖНО СОВПАДАТЬ с количеством таблиц в модели.

- Имя контроллера должно отражать класс, который с которым он взаимодействует из Модели (То есть если в Модели существует класс Users, то контроллер UsersController реализует все методы для работы с этим классом)
 - Имя View должно (по возможности) тоже указывать на таблицу, с которой он работает (AddUsersPage, AuthUsersPage, ...)
 - В случае, если в проекте реализуются методы, применяемые в разных контроллерах (Например, методы проверки правильности заполнения строк, методы по фильтрации данных массива), то их необходимо оформить в качестве дополнительных библиотек
 - Используйте папку Assets для ресурсов проекта (изображений, логотипов, шрифтов, словарей данных)
 - Для вспомогательных классов Используйте папку Classes
4. Реализация модели информационной системы
- 4.1. Необходимо реализовать ERD-диаграмму в СУБД MS SQL SERVER;
- 4.2. Необходимо реализовать ERD-диаграмму информационной системы на основе заданной предметной области, удовлетворяющую следующим требованиям:
- Обязательна 3 нормальная форма с обеспечением ссылочной целостности;
 - При разработке диаграммы обратите внимание на согласованную осмысленную схему именования, необходимо именовать таблицы и поля базы данных ЕДИНООБРАЗНО;
 - Нельзя использовать в именах полей и таблиц СПЕЦСИМВОЛЫ , нельзя использовать при именовании зарезервированные слова и команды SQL (Например, нельзя именовать таблицу User, Function, Procedure. Необходимо использовать незарезервированные имена: Users и т.п.)
 - Необходимо давать "расширенные" имена полям-идентификаторам (например, IdGroup вместо Id)
 - Необходимо (по мере возможности) именовать столбцы с внешними ключами так же как и столбцы, на которые они ссылаются в главной таблице
 - Необходимо (по мере возможности) добавлять NOT NULL в определение столбцов
 - Создайте необходимые первичные и внешние ключи, определите ограничения внешних ключей, отражающие характер предметной области;
- 4.3. Необходимо подключить созданную базу данных к разрабатываемому проекту;
5. Макет приложения
- 5.1. Для стилизации приложения необходимо использовать словарь ресурсов;
- 5.2. Все компоненты системы должны иметь единый согласованный внешний вид, соответствующий руководству по стилю (руководство по стилю должно быть описано в пояснительной записке к курсовому проекту);
- 5.3. Требования к разметке и дизайну:
- предпочтение отдается масштабируемой компоновке;
 - должно присутствовать ограничение на минимальный размер окна;
 - должна присутствовать возможность изменения размеров окна, где это необходимо;
 - увеличение размеров окна должно увеличивать размер контентной части, например, таблицы с данными из БД);
 - группировка элементов (в логические категории);
 - использование соответствующих элементов управления (например, выпадающих списков для отображения подстановочных значений из базы данных);
 - расположение и выравнивание элементов (метки, поля для ввода и т.д.);
 - последовательный переход фокуса по элементам интерфейса (по нажатию клавиши TAB);
 - общая компоновка логична, понятна и проста в использовании;

- 5.4. Необходимо, чтобы приложение имело последовательный пользовательский интерфейс, позволяющий перемещаться между существующими окнами в приложении (в том числе обратно, например, с помощью кнопки «Назад»);
- 5.5. Необходимо, чтобы приложение имело соответствующий заголовок на каждом окне приложения (не должно быть значений по умолчанию типа MainWindow, Form1 и тп)
6. Обратная связь с пользователем
 - 6.1. Необходимо уведомлять пользователя о совершаемых им ошибках или о запрещенных в рамках задания действиях;
 - 6.2. Необходимо запрашивать подтверждение перед удалением, предупреждать о неотвратимых операциях, информировать об отсутствии результатов поиска и т.п.
 - 6.3. Окна сообщений должны быть соответствующих типов (например, ошибка, предупреждение, информация) должны отображаться с соответствующим заголовком и пиктограммой. Текст сообщения должен быть полезным и информативным, содержать полную информацию о совершенных ошибках пользователя и порядок действий для их исправления.
 - 6.4. Необходимо использовать визуальные подсказки для пользователя при вводе данных.
 - 6.5. Не позволяйте пользователю вводить некорректные значения в текстовые поля сущностей. Например, в случае несоответствия типа данных или размера поля введенному значению необходимо оповестить пользователя о совершенной им ошибке.
 - 6.6. При возникновении непредвиденной ошибки приложение **не должно** аварийно завершать работу.
7. Оформление кода
 - 7.1. Идентификаторы переменных, методов и классов должны отражать суть и/или цель их использования, в том числе и наименования элементов управления (например, не должно быть значений по умолчанию типа Form1, button3).
 - 7.2. Идентификаторы должны соответствовать соглашению об именовании (Code Convention) и стилю CamelCase (для C# и Java)
 - 7.3. Допустимо использование не более одной команды в строке.
 - 7.4. Необходимо использовать **комментарии**
 - Используйте комментарии для пояснения неочевидных фрагментов кода. Запрещено комментирование кода. Хороший код воспринимается как обычный текст. Не используйте комментарии для пояснения очевидных действий. Комментарии должны присутствовать только в местах, которые требуют дополнительного пояснения.
 - Используйте тип комментариев, который в дальнейшем позволит сгенерировать XML - документацию, с соответствующими тегами (например, param, return(s), summary и др.)
8. Отчеты
 - 8.1. Необходимо, чтобы при работе приложения создавалась итоговая документация (отчеты) в формате Word, Excel.
9. Тестирование программного продукта
 - 9.1. Необходимо разработать dll-библиотеку, содержащую модульные тесты для проверки правильности заполнения вводимых пользователем данных;
 - 9.2. Важно, чтобы тестовые данные предусматривали различные ситуации. То есть необходимо, чтобы варианты тестирования демонстрировали различные исходы работы алгоритма.
 - 9.3. Необходимо реализовать интеграционные тесты, которые проверяют работоспособность всех методов, реализующих взаимодействие с базой данных (добавление, редактирование, удаление).

3. ТРЕБОВАНИЯ К ДОКУМЕНТАЦИИ

1. Наличие технического задания
 - 1.1. Бланк технического задания приводится в Приложении 1.
 - 1.2. Техническое задание не «подшивается» к пояснительной записке
2. Наличие пояснительно записки, содержащей все разделы
3. Наличие накопителя (CD-диск , FLASH-карта). Накопитель должен содержать пояснительную записку, приложение (проект, скрипт БД), установщик приложения.

4. СТРУКТУРА ПОЯСНИТЕЛЬНО ЗАПИСКИ

Титульный лист

- Титульный лист должен включать: Название образовательного учреждения; Тему курсового проекта, группу, ФИО студента, ФИО руководителя проектирования, город и год.
- Бланк титульного листа приводится в Приложении 1.

Ведомость проекта

- Бланк ведомости проекта приводится в Приложении 2.

Содержание

- Содержание должно генерироваться автоматически

Введение

Во введении следует обосновать актуальность избранной темы курсовой работы, раскрыть ее теоретическую и практическую значимость, сформулировать цели и задачи работы.

Введение должно подготовить читателя к восприятию основного текста работы. Оно состоит из обязательных элементов, которые необходимо правильно сформулировать. **В первом предложении называется тема курсовой работы.**

Актуальность исследования (почему это следует изучать?) Актуальность исследования рассматривается с позиций социальной и практической значимости. В данном пункте необходимо раскрыть суть исследуемой проблемы и показать степень ее проработанности в различных трудах по разработке информационных систем. Здесь же можно перечислить источники информации, используемые для исследования.

Цель исследования (какой результат будет получен?) Цель должна заключаться в решении исследуемой проблемы путем ее анализа и практической реализации.

Объект исследования (что будет исследоваться?). Объект предполагает работу с понятиями. В данном пункте дается определение экономическому явлению, на которое направлена исследовательская деятельность. Опишите место информационной системы в жизни общества, где может использоваться система и почему, какие задачи позволит решить);

Предмет исследования (как, через что будет идти поиск?) Здесь необходимо дать определение планируемым к исследованию конкретным свойствам объекта или способам изучения экономического явления. Предмет исследования направлен на практическую деятельность и отражается через результаты этих действий.

Задачи курсового проектирования (как идти к результату?), пути достижения цели. Задачи соотносятся с гипотезой. Определяются они исходя из целей работы. Формулировки задач необходимо делать как можно более тщательно, поскольку описание их решения должно составить содержание глав и параграфов работы. Как правило, формулируются 3 - 4 задачи.

Пример описания задач:

1. «на основе теоретического анализа литературы и Internet-источников произвести анализ предметной области...».
2. «Определить... » (выделить основные условия, факторы, причины, влияющие на объект исследования).

3. «Раскрыть...» (выделить основные условия, факторы, причины, влияющие на предмет исследования).
4. «Разработать...» (средства, условия, формы, программы).
5. «Протестировать...» (что разработали) и дать рекомендации...

Структура курсового проекта – это завершающая часть введения (что в итоге в работе представлено).

В завершающей части в назывном порядке перечисляются структурные части работы, например: «Структура работы соответствует логике исследования и включает в себя введение, практическую часть, заключение, список литературы, 5 приложений».

Здесь допустимо дать развернутую структуру курсовой работы и кратко изложить содержание глав. (Чаше содержание глав курсовой работы излагается в заключении).

Таким образом, введение должно подготовить к восприятию основного текста работы.

Краткие комментарии по формулированию элементов введения представлены в таблице 1.

Таблица 1. Комментарии по формулированию элементов введения

Элемент введения	Комментарий к формулировке
Актуальность темы	<i>Почему это следует изучать?</i> Раскрыть суть исследуемой проблемы и показать степень ее проработанности.
Цель исследования	<i>Какой результат будет получен?</i> Должна заключаться в решении исследуемой проблемы путем ее анализа и практической реализации.
Объект исследования	<i>Что будет исследоваться?</i> Дать определение явлению или проблеме, на которое направлена исследовательская деятельность.
Предмет исследования	<i>Как и через что будет идти поиск?</i> Дать определение планируемым к исследованию конкретным свойствам объекта или способам изучения явления или проблемы.
Задачи курсового проектирования	<i>Как идти к результату?</i> Определяются исходя из целей работы и в развитие поставленных целей. Формулировки задач необходимо делать как можно более тщательно, поскольку описание их решения должно составить содержание глав и параграфов работы. Рекомендуется сформулировать 3 – 4 задачи.
Структура курсового проекта (завершающая часть введения)	<i>Что в итоге в работе представлено.</i> Краткое изложение перечня и/или содержания глав работы.

Аналитическая часть

Разделы, которые должна включать аналитическая часть пояснительной записки:

1. Цели, функции и задачи системы

Необходимо ПОДРОБНО описать назначение и функционал информационной системы, из каких модулей (интерфейсных окон) состоит информационная система.

2. Описание предметной области системы

Необходимо ПОДРОБНОЕ описание области, для которой разрабатывается проект. Какая информация взята в основу проекта (какие исходные данные) и почему (какие опросы были произведены, что было изучено для того, чтобы определить исходные данные для проекта). Здесь же могут описаны формулы для расчета (если таковые есть).

3. Определение пользователей системы

Необходимо перечислить для кого предназначена ИС и почему, как планируется разграничивать доступ пользователей к ИС, какими средствами будете пользоваться для

реализации разграничения доступа. ЗДЕСЬ НУЖНО добавить диаграмму вариантов использования.

Диаграмма вариантов использования (диаграмма прецедентов) направлена на описание функций, доступных разным группам пользователей, описание принципов разграничения доступа к данным пользователями);

Требования к диаграмме вариантов использования:

- в диаграмме должны быть описаны все участники системы (актеры).
- должны быть описаны возможные варианты сценариев работы с системой с точки зрения пользователя, возможные пути использования системы
- актеры должны быть оформлены в виде "проволочных человечков"
- прецеденты (варианты использования) должны оформляться с помощью отглагольных существительных или глаголами в неопределенной форме
- связь между актером и прецедентом изображается в виде "Сплошной" линии
- связь между прецедентами изображается в виде "Пунктирной" линии (с обязательным указанием типа зависимости: <<include>> или <<extend>>)

4. Проектные решения по реализации тестирования приложения

Требования к разделу:

- Необходимо перечислить все функции приложения, которые планируется протестировать (в соответствии с таблицей)

Таблица

Requirements ID	Описание функции
Req_1	Добавление данных в таблицу «Имя таблиц»
Req_2	
...	

- Необходимо сформировать план тестирования (Test Plan) – проектную документацию описывающую и регламентирующую процесс тестирования. Должен быть сформирован документ, описывающий весь объем работ по тестированию, начиная с описания объекта до стратегии. Test Plan должен быть оформлен в соответствии с таблицей

Таблица

test case ID	описание объекта тестирования (requirements)	стратегия тестирования (вид тестирования)	Последовательность проведения тестирования в разрезе запланированных фаз разработки	Критерии начала тестирования	Критерии окончания тестирования
TC_1	неправильный формат ввода данных в поле «Логин» при авторизации				
TC_2	Добавление информации в таблицу с незаполненными полями				

- Необходимо сформировать Traceability matrix —двумерную таблицу, содержащую соответствие функциональных требований продукта (functional requirements) и

подготовленных тестовых сценариев (test cases). Traceability matrix должна быть оформлена в соответствии с таблицей

Таблица

	test case ID	TC_1	TC_2	TC_3	TC_4	TC_5	TC_6	TC_7	TC_8
Requirements ID									
Req_1		+							
Req_2			+		+				
Req_3				+			+		
Req_4			+		+				
Req_5						+			
Req_6							+		
Req_7					+			+	
Req_8									+

5. Проектные решения по реализации интерфейса приложения (разработка руководства по стилю)

5.1. Разработка руководства по стилю

Необходимо разработать руководство по стилю.

Руководство по стилю должно быть выполнено в Figma. Образец руководства по стилю: <https://www.figma.com/file/JKu2XYFYTIKzmXsmy6TAyT/Untitled?node-id=0%3A1&t=R1KISvv9R3Nu8KdY-1>

Рабочие ссылки на руководство по стилю должны быть представлены в пояснительной записке к информационной системе, а также продемонстрированы в презентации при публичной защите курсового проекта.

Требования к руководству по стилю:

- Необходимо описать требования к шрифтам, которые используются в проекте;
- Все экранные формы пользовательского интерфейса должны иметь заголовки и логотип. Поэтому в руководстве по стилю должны быть описаны требования к логотипу проекта (логотип должен быть добавлен в руководство по стилю в виде изображения);
- Описаны требования к иконке проекта;
- Описаны требования к цветовому оформлению (требования к заднему фону, кнопкам и др. элементам оформления);
- Описаны требования к размещению элементов и их размерам; (Например, значение высоты для маленьких, обычных и больших графических элементов (кнопок, текстовых полей, и т. п.) , внутренний отступ графических элементов (padding) , внешний отступ относительно других элементов (margin) , размер скругления границ (corner radius))

5.2. Прототип приложения

Необходимо представить прототип приложения.

Необходимо описать принципы, которыми руководствовались при создании интерфейса (Описать какие элементы интерфейса более заметны и почему? Обоснование размещения элементов в модальных окнах, на втором плане. По каким принципам группировали элементы?).

Проектные решения по реализации интерфейса приложения должны быть выполнены в Figma. Рабочие ссылки на прототип приложения должны быть представлены в пояснительной записке к информационной системе, а также продемонстрированы в презентации при публичной защите курсового проекта.

1. Модель информационной системы

1.1. Схема базы данных

Необходимо создать физическую модель базы данных (ERD-диаграмму), реализованную средствами СУБД MS SQL SERVER

Если ERD-диаграмма очень большая (10 и более таблиц), то рекомендуется вынести ее в приложение, сделав ссылку на нее в проектной части пояснительной записки.

1.2. Словарь данных

Необходимо создать словарь данных. Словарь данных создается для однозначного понимания того, что изображено в ERD-диаграмме. Необходимо сформировать словарь данных, который обеспечит понимание структуры и назначения каждого поля, описанного в сущностях спроектированной базы данных. Шаблон оформления словаря данных для одной сущности представлен в таблице 1.

Таблица 1 - Шаблон для оформления словаря данных

Ключ	Имя поля	Тип данных	Нулевые значения	Дополнительные	Описание
PK	question_id	INTEGER	N	Автозаполнение с шагом 1	ID Вопроса
	question	NVARCHAR(50)	N		Вопрос
FK	prof_id	INTEGER	N		ID Профессии

2. Файловая структура приложения

Требования к описанию файловой структуры:

- Исходя из функционала, описанного в ТЗ проекта, необходимо описать файловую структуру приложения и добавить скрин структуры папок приложения.
- Необходимо описать назначение каждой из папок приложения: папки Model, папки View, папки Controllers, и других.
- Необходимо создать диаграммы классов, чтобы наглядно описать созданные контроллеры. Диаграммы классов должны использоваться в пояснительной записке для описания принципа работы приложения.
- Необходимо описать методы, реализованные в созданных контроллерах (Например, методы добавления данных в таблицу, методы по фильтрации данных массива). Каждый из созданных методов должен быть протестирован (для этих целей создаются дополнительные библиотеки).
- Необходимо представить XAML-документацию к созданным методам.

3. Реализация кода приложения

Требования к описанию разделу:

- Необходимо создать «Диаграммы последовательности» для описания процессов взаимодействия модулей программного продукта;
- Необходимо описать алгоритмы, используемые при разработке кода приложения.

4. Разработка dll-библиотек

Требования к библиотекам:

- Каждый из созданных методов должен быть протестирован.
- Для каждого из созданных методов необходимо разработать позитивные и негативные тестовые случаи (test case); Объем тестовых данных должен охватывать все возможные ситуации при выполнении заданных операций;
- Для каждого теста необходимо разработать тестовую документацию. Документация должна быть оформлена в соответствии с шаблоном, представленным в таблице 2.

Таблица 2. Шаблон для оформления тестовой документации

Тестовый пример #	
Приоритет тестирования	
Заголовок/название теста	
Краткое изложение теста	
Этапы теста	
Тестовые данные	
Ожидаемый результат	
Фактический результат	
Статус	
Предварительное условие	
Постусловие	
Примечания/комментарии	

5. Работа с системой контроля версий

Требования к разделу:

- Необходимо проявить навыки работы с системой контроля версий GitHub, GitLab, Gogs или другие;
- Необходимо подробно описать процесс работы с системой контроля версий;
- Необходимо наглядно показать процесс контроля над версионностью программного продукта при работе с системой контроля версий с помощью скриншотов/команд;
- Необходимо представить ссылку на проект, размещенный в системе контроля версий.

6. Документация по работе с информационной системой

6.1. Руководство пользователя

В руководстве представляются приводятся подробные инструкции по действиям пользователей при выполнении различных операций по использованию системы для решения различных прикладных задач.

Требования к руководству пользователя:

- Необходимо, чтобы руководство пользователя описывало последовательность действий для выполнения всех функций вашей системы.
- Необходимо использовать живые примеры и скриншоты вашей системы для более наглядного пояснения шагов работы с различным функционалом.
- Необходимо разделить руководство на подразделы, использовать ссылки на рисунки, нумерованные и маркированные списки для описания шагов и т.д.

6.2. Руководство администратора

Требования к руководству администратора:

- Для выполнения операций редактирования, добавления, удаления данных информационной системы, в системе должны быть предусмотрены специальные категории пользователей;

- Необходимо предусмотреть личный кабинет администратора, вход в который осуществляется после авторизации;
- Обязательные элементы личного кабинета администратора: страница авторизации (поля Логин, Пароль, кнопка «Войти»);
- Необходимо, чтобы руководство администратора описывало последовательность действий для выполнения всех функций вашей системы.
- Необходимо использовать живые примеры и скриншоты вашей системы для более наглядного пояснения шагов работы с различным функционалом.
- Необходимо разделить руководство администратора на подразделы, использовать ссылки на рисунки, нумерованные и маркированные списки для описания шагов и т.д.

6.3. Руководство по установке программы

Требования к руководству по установке программы:

- Необходимо создать инсталлятор приложения;
- Необходимо выполнить подробное описание процесса развертывания приложения для реализации его полной работоспособности;
- Необходимо описать процесс подключения к БД ;
- Необходимо наглядно пояснить процесс установки приложения с помощью скриншотов;
- Необходимо разделить руководство по установке программы на подразделы, использовать ссылки на рисунки, нумерованные и маркированные списки для описания шагов и т.д.

ГЛОССАРИЙ

В пояснительную записку должны быть включены:

- словарь терминов,
- список используемых сокращений, которые были использованы при написании текста пояснительной записки.

Например,

1. БД – база данных
2. АРМ - автоматизированного рабочего места
3. СУБД MS SQL SERVER – система управления базами данных Microsoft SQL Server

СПИСОК ИСТОЧНИКОВ

Список используемой литературы должен содержать 20 – 25 источников (не менее 10 книг и 10-15 материалов периодической печати), с которыми работал автор курсовой работы.

Пример оформления списка источников:

1. PlantUML в двух словах [Электронный ресурс] - Режим доступа: <https://plantuml.com/ru/>
2. Н. А. Тюкачев, В. Г. Хлебопстроев С#. Основы программирования. Учебное пособие (+ CD) - СПб.: ЭБС ЛАНЬ, 2017. – 272 с.

Заключение

В заключении необходимо подвести итоги по теме. Заключение носит форму синтеза полученных в работе результатов. Его основное назначение - резюмировать содержание работы, подвести итоги проведенного исследования. В заключении излагаются полученные выводы и их соотношение с целью исследования, конкретными задачами, гипотезой, сформулированными во введении.

Приложения

Приложения могут отсутствовать в пояснительной записке. Целесообразно вынести в приложения материал, который «не вписывается» в общий контекст описания проектной работы

В большинстве случаев, в Приложение «выносят» справочные материалы по проекту или материалы, несущие описательный характер и показывающие объем проделанной работы.

