

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Магнитогорский государственный технический университет
им. Г.И.Носова»
Многопрофильный колледж



**МЕТОДИЧЕСКИЕ УКАЗАНИЯ
ПО ВЫПОЛНЕНИЮ КУРСОВОГО ПРОЕКТА**

ПМ.02 ОСУЩЕСТВЛЕНИЕ ИНТЕГРАЦИИ ПРОГРАММНЫХ МОДУЛЕЙ

МДК.02.02 Инструментальные средства разработки программного обеспечения

для студентов специальности

специальности 09.02.07 Информационные системы и программирование

Магнитогорск, 2020

ОДОБРЕНО

Предметно-цикловой комиссией «Информатики и
вычислительной техники»

Председатель *И.Г.Зорина*

Протокол № 7 от 17.02.2020

Методической комиссией МпК

Протокол №3 от «26» февраля 2020г

Разработчик:

преподаватель ФГБОУ ВО «МГТУ им. Г.И. Носова» Многопрофильный колледж

Власта Диляуровна Тутарова

Методические указания по выполнению курсового проекта разработаны на основе рабочей программы ПМ.02 ОСУЩЕСТВЛЕНИЕ ИНТЕГРАЦИИ ПРОГРАММНЫХ МОДУЛЕЙ.

Содержание курсового проекта ориентировано на формирование общих и профессиональных компетенций по программе подготовки специалистов среднего звена по специальности 09.02.07 Информационные системы и программирование

ВВЕДЕНИЕ

Настоящие методические указания предназначены для студентов очной формы обучения в качестве регламентирующего материала по выполнению и предоставлению курсового проекта по профессиональному модулю «Осуществление интеграции программных модулей» для специальности 09.02.07 Информационные системы и программирование.

Выполнение курсового проекта рассматривается как вид учебной деятельности по профессиональному модулю профессионального учебного цикла и реализуется в пределах времени, отведенного на его изучение.

Выполнение студентом курсового проекта по профессиональному модулю проводится с целью:

- систематизации и закрепления полученных теоретических знаний и практических умений по междисциплинарному курсу;
- углубления теоретических знаний в соответствии с заданной темой;
- формирования умений применять теоретические знания при решении поставленных задач, использовать справочную, нормативную и научно-техническую литературу (формирование профессиональных компетенций);
- формирование общих и профессиональных компетенций – развитие творческой инициативы, дисциплинированности, целеустремлённости, аккуратности, самостоятельности, ответственности и организованности;
- подготовка к государственной итоговой аттестации (ГИА).

1 Общие положения

В соответствии с рабочей программой ПМ.02 Осуществление интеграции программных модулей МДК.02.02 Инструментальные средства разработки программного обеспечения предусмотрено выполнение курсового проекта.

Курсовой проект является одним из основных видов учебной деятельности и формой контроля учебной работы студентов.

Продолжительность выполнения курсового проекта – 30 часов. Курсовой проект осуществляется на заключительном этапе изучения междисциплинарного курса, в ходе которого формируются умения, ПК и ОК при решении задач, связанных со сферой профессиональной деятельности будущих специалистов.

Курсовой проект выполняется после изучения теоретической части МДК.02.02 **Инструментальные средства разработки программного обеспечения**, ПМ.02 Осуществление интеграции программных модулей.

В результате выполнения курсового проекта, Вы будете уметь:

уметь:

- У1 использовать выбранную систему контроля версий.
- У2 использовать методы для получения кода с заданной функциональностью и степенью качества.
- У3. анализировать проектную и техническую документацию.
- У4. использовать специализированные графические средства построения и анализа архитектуры программных продуктов.
- У5. организовывать заданную интеграцию модулей в программные средства на базе имеющейся архитектуры и автоматизации бизнес-процессов.
- У6. определять источники и приемники данных.
- У7. использовать приемы работы в системах контроля версий.
- У8. выполнять отладку, используя методы и инструменты условной компиляции (классы Debug и Trace)).
- У9. оценивать размер минимального набора тестов.
- У10. разрабатывать тестовые пакеты и тестовые сценарии.
- У11. выявлять ошибки в системных компонентах на основе спецификаций.
- У12 использовать различные транспортные протоколы и стандарты форматирования сообщений.
- У13 выполнять тестирование интеграции.
- У14 организовывать постобработку данных.
- У15 создавать классы-исключения на основе базовых классов.
- У16 выполнять ручное и автоматизированное тестирование программного модуля.
- У17 использовать инструментальные средства отладки программных продуктов.

Содержание курсового проекта ориентировано на формирование

общих компетенций:

ОК 1. Выбирать способы решения задач профессиональной деятельности, применительно к различным контекстам

ОК 2. Осуществлять поиск, анализ и интерпретацию информации, необходимой для выполнения задач профессиональной деятельности.

ОК 3 Планировать и реализовывать собственное профессиональное и личностное развитие.

ОК 4 Планировать и реализовывать собственное профессиональное и личностное развитие.

ОК 5 Планировать и реализовывать собственное профессиональное и личностное развитие.

ОК 6 Проявлять гражданско-патриотическую позицию, демонстрировать осознанное поведение на основе традиционных общечеловеческих ценностей

ОК 7 Содействовать сохранению окружающей среды, ресурсосбережению, эффективно действовать в чрезвычайных ситуациях.

ОК 8 Использовать средства физической культуры для сохранения и укрепления здоровья в процессе профессиональной деятельности и поддержания необходимого уровня физической подготовленности

ОК 9 Использовать информационные технологии в профессиональной деятельности.

ОК 10 Пользоваться профессиональной документацией на государственном и иностранном языке

ОК 11 Планировать предпринимательскую деятельность в профессиональной сфере

профессиональных компетенций:

ПК 2.1. Разрабатывать требования к программным модулям на основе анализа проектной и технической документации на предмет взаимодействия компонент

ПК 2.2. Выполнять интеграцию модулей в программное обеспечение

ПК 2.3 Выполнять отладку программного модуля с использованием специализированных программных средств

ПК 2.4 Осуществлять разработку тестовых наборов и тестовых сценариев для программного обеспечения.

ПК 2.5. Производить инспектирование компонент программного обеспечения на предмет соответствия стандартам кодирования

Курсовой проект по профессиональному модулю выполняется в сроки, определённые учебным планом по программе подготовке специалистов среднего звена.

Процесс выполнения курсовых проектов включает следующие этапы:

1 Изучение настоящих методических указаний.

2 Выбор темы и её согласование с руководителем

3 Формулировка цели и составление плана.

4.Изучение предметной области

5.Поиск аналогов, выбор и обоснование проектного решения

6.Выбор инструментария, установка необходимых средств проектирования на ПК

7.Разработка и интеграция программных модулей

8.Отладка и тестирование программного продукта

9. Оформление пояснительной записки.

10 Подготовка к защите и защита курсового проекта.

Контроль за выполнением разделов КП осуществляется преподавателем-консультантом, заведующим отделением.

Примерная тематика курсового проекта:

Создание настольных, веб и мультимедийных приложений для организаций и предприятий разного уровня: для государственных учреждений, организаций сферы услуги, предприятий связи, производственных предприятий и т.д., создание игрового приложения различных жанров.

Типы разрабатываемых настольных, веб, мультимедийных и игровых приложений:

- электронное сопровождение учебного процесса по дисциплинам и профессиональным модулям;
- разработка, внедрение и адаптация web-приложений различной тематики;
- разработка автоматизированных информационных систем различной тематики;
- создание мобильных, компьютерных игр различных жанров.

2 Структура курсового проекта

Структура курсового проекта включает:

- пояснительную записку;
- программную часть.

Текстовый документ курсового проекта должен включать в указанной последовательности следующие элементы:

Титульный лист;

Задание;

Содержание;

Введение

1 Теоретико-информационный анализ предметной области

1.1 Анализ предметной области

1.2 Анализ и оценка существующих программных продуктов

1.3 Постановка задачи

2 Проектирование программного обеспечения

2.1 Моделирование программного обеспечения

2.2 Обоснование выбора средств разработки

2.3 Разработка базы данных

2.4 Проектирование меню и интерфейса программного обеспечения

2.5 Спецификация программы

2.6 Описание запросов при разработке программного обеспечения

2.7 Проектирование отчетов

2.8 Администрирование программного обеспечения

3 Описание работы программного обеспечения

3.1 Руководство пользователя

3.2 Руководство программисту

Заключение

Список использованных источников

Приложение А (листинг программного продукта)

Титульный лист оформляется в соответствии с приложением А.

Содержание

В нем содержится название глав и разделов с указанием страниц. Содержание оформить с помощью автоматического оглавления.

Введение

Во введении обосновывается актуальность выбранной темы, ее важность. Формируются цель и задачи курсового проекта. Желательно, чтобы было отражено значение рассматриваемых вопросов, помимо этого, во введении должна обосновываться структура курсового проекта.

1 Теоретико-информационный анализ предметной области

Данный раздел предназначен для раскрытия предметной области темы курсового проекта. Данный раздел состоит из:

1.1 Анализ предметной области

В данном разделе изучаются особенности предметной области.

При таком анализе принято выделять три этапа:

1 — анализ требований и информационных потребностей;

2 — определение информационных объектов и связей между ними;

3 — конструирование концептуальной модели предметной области.

Данный этап включает следующие задачи:

- определение перечня задач по извлечению, обработке, хранению, транспортировке и представлению (в том числе документированию) информации;
- определение требований к составу, структуре, формам представления информации;
- прогнозирование возможных изменений информационных ресурсов как в количественном, так и в содержательном плане.

Наиболее важными при определении целей проекта создания ИС является получение полной информации о бизнес-процессах предприятия. Необходимо описать и проанализировать все особенности организационной структуры всего предприятия и отдельных подразделений, правила движения информационных потоков и используемые программно-аппаратные средства. В результате такого анализа должны быть учтены все правила, регламентирующие деятельность сотрудников предприятия, документооборот и систему управления качеством. Только после полного анализа всех элементов присутствующих в деятельности подразделений предприятия, возможно успешное определение целей проекта и эффективное планирование дальнейших действий:

- реинжиниринг бизнес-процессов и активностей предприятия;
- определение потребностей предприятия;
- оценка возможных перспектив развития.

1.2 Анализ и оценка существующих программных продуктов

В данном разделе следует выполнить анализ и оценку уже существующих программных продуктов аналогичной направленности (не менее трех).

При оценке указать кем и когда создан, основной функционал программы, достоинства и недостатки. Привести скрины экрана программы.

1.3 Постановка задачи

Постановка задачи – это точная формулировка условий задачи с описанием входной и выходной информации, а также это обязательный этап создания компьютерного программного обеспечения, который включает в себя формальное описание создаваемого проекта.

Указать цель и задачи разработки программного продукта, аудиторию назначения. Перечислить планируемый функционал, пользователей с вариантами доступа к функционалу.

2 Проектирование программного обеспечения

Проектирование программного обеспечения является важнейшей стадией при создании информационной системы, т.к. именно на этом этапе принимаются очень важные стратегические решения.

2.1 Моделирование программного обеспечения

В данном разделе приводится информационная модель, концептуальная модель, схема базы данных, функциональная модель, модель прецедентов, модель последовательности, алгоритм работы программного обеспечения и т.д.

Самой популярной технологией высокоуровневого проектирования данных является концепция модели «сущность-связь». Диаграммы «сущность-связь» (рис. 1) предназначены для графического представления моделей данных разрабатываемой программной системы и предлагают некоторый набор стандартных обозначений для определения данных и отношений между ними. С помощью этого вида диаграмм можно описать отдельные компоненты концептуальной модели данных и совокупность взаимосвязей между ними, имеющих важное значение для разрабатываемой системы.

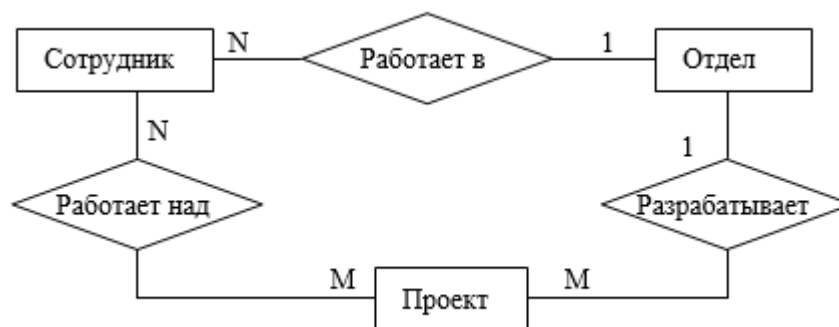


Рисунок 1 – Диаграмма «сущность-связь» для общего примера компании

На этом этапе необходимо рассмотреть, из каких объектов может состоять база данных, каковы свойства этих объектов, как связаны объекты между собой. После разбиения базы данных на отдельные объекты необходимо рассмотреть свойства каждого из этих объектов, другими словами, установить, какими параметрами описывается каждый объект.

Цель логического проектирования – преобразование концептуальной модели на основе выбранной модели данных в логическую модель, не зависящую от особенностей используемой в дальнейшем СУБД для физической реализации базы данных. Для ее достижения выполняются следующие процедуры.

1. *Выбор модели данных.* Чаще всего выбирается реляционная модель данных в связи с наглядностью табличного представления данных и удобства работы с ними.

2. *Определение набора таблиц исходя из ER-модели и их документирование.* Для каждой сущности ER-модели создается таблица. Имя сущности – имя таблицы. Осуществляется формирование структуры таблиц. Устанавливаются связи между таблицами посредством механизма первичных и внешних ключей. Структуры таблиц и установленные связи между ними документируются.

3. *Нормализация таблиц.* На этом шаге проверяется корректность структуры таблиц, созданных на предыдущем шаге, посредством применения к ним процедуры нормализации. В результате нормализации получается очень гибкий проект базы данных, позволяющий легко вносить в нее нужные расширения.

4. *Определение требований поддержки целостности данных и их документирование.* Эти требования представляют собой ограничения, которые вводятся с целью предотвратить помешение в базу данных противоречивых данных. На этом шаге вопросы целостности данных освещаются безотносительно к конкретным аспектам ее реализации. Должны быть рассмотрены следующие типы ограничений:

обязательные данные. Выясняется, есть ли атрибуты, которые не могут иметь Null-значений;
ограничения для значений атрибутов. Определяются допустимые значения для атрибутов;
целостность сущностей. Она достигается, если первичный ключ сущности не содержит Null-значений;

ссылочная целостность. Она понимается так, что значение внешнего ключа должно обязательно присутствовать в первичном ключе одной из строк таблицы для родительской сущности;

Сведения обо всех установленных ограничениях целостности данных помещаются в словарь данных.

5. *Создание окончательного варианта логической модели данных.* На этом шаге подготавливается окончательный вариант ER-модели, представляющей логическую модель данных. Сама модель и обновленная документация, включая словарь данных и реляционную схему связи таблиц, представляется для просмотра и анализа пользователям, которые должны убедиться, что она точно отображает предметную область.

Построенная логическая модель данных в дальнейшем будет востребована на этапе физического проектирования, а также на этапе эксплуатации и сопровождения уже готовой системы, позволяя наглядно представить любые вносимые в базу данных изменения.

На физическом уровне производится выбор рациональной структуры хранения данных и методов доступа к ним, которые обеспечивает выбранная СУБД. На этом уровне решаются вопросы эффективного выполнения запросов к базе данных, для чего строятся дополнительные структуры, например, индексы. В физической модели содержится информация обо всех объектах базы данных (таблицах, индексах, процедурах и др.) и используемых типах данных. Физическая модель зависит от конкретной СУБД. Одной и той же логической модели может соответствовать несколько разных физических моделей. Физическое проектирование является начальным этапом реализации базы данных.

2.2 Обоснование выбора средств разработки

Задача выбора состоит в установлении соответствия между характеристиками разрабатываемого программного обеспечения, организационно-техническими условиями его разработки и параметрами выбираемых инструментальных средств автоматизации с учетом ограничений ресурсов проектирования. При этом проблемная область разрабатываемого программного обеспечения является одним из важнейших факторов, учитываемых при выборе.

В ходе планирования и разработки программного обеспечения необходимо проанализировать и выбрать средства разработки, указав их достоинства и недостатки при разработке конкретного продукта.

2.3 Разработка базы данных

На этапе реализации производится создание базы данных (рис.2) и разработка программ (приложений) в выбранной СУБД.

```
CREATE TABLE `aircraft` (  
  `id_aircraft` BIGINT(225) NOT NULL AUTO_INCREMENT,  
  `personal_number` INT(11) NOT NULL,  
  `type` INT(200) NOT NULL,  
  `number_of_seats` INT(11) NOT NULL,  
  `description` VARCHAR(255) NOT NULL,  
  PRIMARY KEY (`id_aircraft`),  
  FOREIGN KEY (`type`) REFERENCES `type` (`id_type`))  
ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

Рисунок 2 – Пример скрипта создания таблицы

Эксплуатация начинается с заполнения базы данных реальными данными. На этом этапе необходимо сопровождение базы данных, т.е. проведение контроля непротиворечивости, резервное копирование, архивирование и т.д.

По мере использования базы данных происходит выявление недоработок, уточнение и, возможно, изменение требований к базе данных. В результате может быть принято решение о ее модификации.

2.4 Проектирование меню и интерфейса программного обеспечения

Интерфейс пользователя должен быть разработан и функционировать в соответствии с техническим заданием; приложение предварительно смоделировать; использовать анимационные эффекты; код оформить в соответствии со стандартами кодирования.

Интерфейс с пользователем (все виды взаимодействия с пользователем: интерактивный режим, обработка ошибок, автономная работа).

В ходе разработки интерфейса пользователя необходимо определить требования, общие спецификации пользовательских функций и алгоритмов обработки данных.

Требования к графическому интерфейсу:

Содержать привычные и понятные пользователю пункты меню, соответствующие функциям обработки.

Ориентироваться на пользователя, который общается с программой на внешнем уровне взаимодействия.

Удовлетворять правилу "шести" - в одну линейку меню включать не более 6 понятий, каждое из которых содержит не более 6 опций.

Сохранять стандартизированное назначение и местоположение на экране графических объектов.

При создании интерфейса, к нему предъявляются следующие требования:

- наличие форм ввода данных;
- наличие форм вывода данных;
- удобная навигация;
- наличие меню;
- наличие всплывающих подсказок;
- защита от некорректного ввода данных;
- обработка исключительных ситуаций;
- наличие сведений о программе;
- наличие сведений об авторе.

2.5 Спецификация программы

Спецификация программы – точная и полная формулировка задачи, содержащая информацию, необходимую для построения алгоритма (программы) решения этой задачи. В отличие от компьютерной программы спецификация обращена, прежде всего, к человеку и представляет собой описание в терминах, характерных для самой задачи, а не для ее реализации. Она служит основой дальнейшей детализации и разработки.

Программное обеспечение учета стоматологических услуг представлено ниже в виде схемы модулей (рис. 3).



Рисунок 3 – Спецификация программы

Каждый модуль отвечает за свои определенные действия и выполняет определенные функции.

Модуль авторизации проверяет на подлинность введенные данные, а именно – логин и пароль. Если эти данные введены неверно, появляется сообщение об ошибке. Если данные успешно прошли проверку, пользователь попадает на модуль формы меню.

Модуль должности/специальности представляет собой форму для просмотра информации об должностях и специальностях, а также их данных. Через этот модуль имеется возможность добавление новых, изменение существующих или удаление ненужной информации о должностях и специальностях, а также имеется возможность сортировки и поиска информации.

Модуль графика работы специалистов представляет собой форму для просмотра информации о графике работы специалистов, которые будут предоставлять стоматологические услуги. Через этот модуль имеется возможность добавление новой, изменение существующей или удаление ненужной информации о графике работы, а также имеется возможность фильтрации

информации. Только отсюда имеется возможность перейти на модуль кабинеты. В этом модуле есть возможность ввода информации о графике работы, которая отобразится в модуле оформления и т.д.

2.6 Описание запросов при разработке программного обеспечения

Запрос – это специальное строковое обращение к базе, в котором отражаются поля (частицы данных) и условия, по которым эту информацию необходимо предоставить. Запросы нужны для автоматизации работы с базой данных.

При помощи запросов SQL можно:

- создавать таблицы БД;
- изменять таблицы БД;
- удалять таблицы БД;
- вставлять записи (строки) в таблицы БД;
- редактировать записи в таблицах БД;
- извлекать выборочную информацию из таблиц БД;
- удалять выборочную информацию из БД.

Создание запросов должно сопровождаться обоснованием их целесообразности и необходимости. Прописывается текст запроса, SQL-запрос и результат выполнения данного запроса (рис.4).

Запрос при поиске рейсов.

```
SELECT schedule.id_schedule, id_flight, rate.id_rate, sum.id_sum, grade.id_grade, number_flight, a1.airport_name as an1, a2.airport_name as an2, a1.city as city_ot, a2.city as city_kud, travel_time, departure_date, stravel_time, aname, price, rate_name, grade_name FROM schedule INNER JOIN (aeroflot INNER JOIN (grade INNER JOIN (rate inner join (sum inner join flight on sum.id_sum=flight.id_sum) on rate.id_rate=sum.id_rate) on grade.id_grade=sum.id_grade) on aeroflot.id_aeroflot=flight.id_aeroflot) on schedule.id_schedule=flight.id_schedule INNER JOIN airport a1 ON a1.id_airport = schedule.whence INNER JOIN airport a2 ON a2.id_airport = schedule.swhere WHERE (a2.city = 'Челябинск' and a1.city = 'Москва')
```

Результаты поиска

ФИО: Резник Петр Евгеньевич
Дата бронирования: 25 декабря 2018

Откуда: Москва, Шереметьево Куда: Челябинск, Баландино Номер рейса: UYU77

Дата вылета: 2019-01-09
Время вылета: 08:27:00
Время в пути: 01:25:00

Аэрофлот: Амур
Тариф: Самый дешевый
Класс: эконом-класс

Стоимость: 5845.00

Рисунок 4 – Пример запроса и результат вывода

2.7 Проектирование отчетов

Создание отчетов должно сопровождаться обоснованием их целесообразности и необходимости. Отчетом считается любой выходной документ с расширением .docx, .xlsx, .pdf (рис.5).



Рисунок 5 – Пример вывода отчета в PDF

2.8 Администрирование программного обеспечения

Администрирование программного обеспечения – это приведение программного обеспечения в соответствие с целями и задачами, для которых оно предназначено. Достигается путём управления, позволяющего минимизировать затраты времени и ресурсов, направляемых на управление системой, и в тоже время максимизировать доступность, производительность и продуктивность системы.

Одним из важных пунктов является защита сервера и базы данных от случайных или намеренных действий, способных привести к удалению, изменению или обнародованию данных. С помощью средств управления пользователями и привилегиями нужно настроить и защитить доступ к серверу баз данных (рис.6).



Рисунок 6 – Авторизация пользователя

3 Описание работы программного обеспечения

В разделе необходимо рассмотреть порядок работы с программным обеспечением.

3.1 Руководство пользователя

Один из важных эксплуатационных документов является Руководство пользователя. При разработке этого документа следует учитывать следующие рекомендации:

- руководство должно содержать все инструкции, необходимые пользователю;
- изложение должно быть ясным, короткими предложениями;
- следует избегать технического жаргона и узкоспециальной терминологии;
- будьте точны и рациональны – длинные и запутанные руководства обычно никто не читает, например, лучше привести рисунок формы, чем долго ее описывать.

Руководство пользователя содержит следующие разделы:

общие сведения о программном продукте (наименование программного продукта, краткое описание его функций, реализованных методов и возможных областей применения);

описание запуска (описание действий по запуску программы и сообщений, которые при этом могут быть получены);

инструкции по работе или описание пользовательского интерфейса (описание режимов работы, форматов ввода-вывода информации и возможных настроек);

сообщения пользователю (содержит перечень возможных сообщений, описание их содержания и действий, которые необходимо предпринять по этим сообщениям).

3.2 Руководство программисту

Руководство программиста должно содержать следующие разделы:

Назначение и условия применения - назначение и функции, условия, необходимые для выполнения (объем оперативной памяти, требования к составу и параметрам периферийных устройств, требования к программному обеспечению).

Характеристики программы - описание основных характеристик и особенностей программы (временные характеристики, режим работы, средства контроля правильности выполнения и самовосстанавливаемости программы и т.п.)

Обращение к программе - описание процедур вызова программы (способы передачи управления и параметров данных и др.).

Входные и выходные данные - описание организации входной и выходной информации и, при необходимости, ее кодирования.

Сообщения - тексты сообщений, выдаваемых программисту или оператору в ходе выполнения программы, описание их содержания и действий, которые необходимо предпринять по этим сообщениям.

В зависимости от особенностей документа допускается объединять отдельные разделы и вводить новые.

В приложении к руководству программиста могут быть приведены дополнительные материалы (примеры, иллюстрации, таблицы, графики и т.п.).

Заключение

В этой части студент формулирует основные выводы по теме и рекомендации относительно возможностей практического применения материалов курсового проекта. Следует также указать на имеющиеся проблемы и возможные перспективы их решения.

Список использованной литературы

Перечисляется литература, использованная при выполнении курсового проекта, в том числе и ГОСТы.

Приложение

В приложение нужно поместить листинги программного кода.

3 Требования к оформлению пояснительной записки

Пояснительная записка является неотъемлемой частью проекта и представляется вместе с проектной частью.

Пояснительная записка курсового проекта включает:

– введение, в котором раскрывается актуальность и значение темы, формируется цель и задачи выполнения курсового проекта;

– аналитическая часть: анализ предметной области, объекта проектирования, обоснование выбора средств разработки;

– проектная часть: создание базы данных, физическая и логическая структура сайта, разработка интерфейса пользователя, создание запросов и отчетов, создание пользователей и предоставление им прав доступа, руководство пользователя;

– заключение, в котором раскрывается значимость рассмотренных вопросов; приводятся главные выводы, характеризующие итоги проделанной работы; излагаются предложения и рекомендации по внедрению полученных результатов и дальнейшему развитию темы;

– список используемых источников;

– приложение.

Оформление пояснительной записки должно строго соответствовать:

– СМК-О-СМГТУ-42-09 Курсовая работа (проект): структура, содержание, общие правила выполнения и оформления;

– СМК-К-РИ-109-15 Порядок организации выполнения и защиты курсовой работы (проекта) в многопрофильном колледже.

Общие требования

Страницы текста, включая иллюстрации и таблицы, должны соответствовать формату А4 (210x297 мм) по ГОСТ 9327.

Текст должен быть выполнен с одной стороны листа белой бумаги рукописным способом, а также с применением печатающих и графических устройств ЭВМ с соблюдением следующих размеров полей: левое – 20 мм, правое – 10 мм, верхнее – 20 мм, нижнее – 20 мм. При наборе текста в Microsoft Word следует придерживаться следующих требований: основной шрифт Times New Roman или Arial, размер шрифта 12-14 пт, цвет – черный, абзацный отступ 10-12,5 мм, межстрочный интервал – одинарный или полуторный. Разрешается использовать компьютерные возможности акцентирования внимания на определенных терминах, формулах, применяя шрифты разной гарнитуры.

Построение текста

Текст курсового проекта следует делить на разделы, подразделы, пункты. Пункты, при необходимости, могут быть разделены на подпункты.

Каждый раздел текста рекомендуется начинать с новой страницы.

Разделы курсового проекта должны иметь порядковые номера, обозначенные арабскими цифрами и записанные с абзацного отступа. Подразделы должны иметь нумерацию в пределах каждого раздела, пункты – в пределах подраздела, подпункты – в пределах пункта.

Если раздел или подраздел состоит, соответственно, из одного подраздела или пункта, то этот подраздел или пункт нумеровать не следует. Точка в конце номеров разделов, подразделов, пунктов, подпунктов не ставится. Если заголовок состоит из двух предложений, их разделяют точкой.

Внутри разделов, подразделов, пунктов или подпунктов могут быть приведены перечисления.

Перед каждым перечислением следует ставить дефис или, при необходимости, в случае наличия ссылки в тексте документа на одно из перечислений, строчную букву (за исключением ё, з, о, ь, й, ы, ь), после которой ставится скобка.

Для дальнейшей детализации перечислений необходимо использовать арабские цифры, после которых ставится скобка, а запись производится с абзацного отступа.

Разделы, подразделы должны иметь заголовки. Пункты и подпункты заголовков могут не иметь.

Заголовки должны четко и кратко отражать содержание разделов, подразделов, пунктов.

Заголовки разделов, подразделов и пунктов следует начинать с абзацного отступа, с прописной буквы, без точки в конце, не подчеркивая. В начале заголовка помещают номер соответствующего раздела, подраздела, либо пункта. Переносы слов в заголовках не допускаются. Если заголовок состоит из двух предложений, их разделяют точкой.

Расстояние между заголовком и текстом должно быть равно удвоенному межстрочному расстоянию; между заголовками раздела и подраздела - одному межстрочному расстоянию.

4 Требования к изложению текста курсового проекта

Текст излагается кратким чётким языком. Терминология и обозначения должны соответствовать установленным стандартам, а при отсутствии стандартов - общепринятым нормам в научно - технической литературе.

Изложение текста курсового проекта должно строго соответствовать

–СМК-О-СМГТУ-42-09 Курсовая работа (проект): структура, содержание, общие правила выполнения и оформления;

–СМК-К-РИ-109-15 Порядок организации выполнения и защиты курсовой работы (проекта) в многопрофильном колледже.

В ТД не допускается:

– применять для одного и того же понятия различные научно-технические термины, близкие по смыслу (синонимы), а также иностранные слова и термины при наличии равнозначных слов и терминов в русском языке;

– применять произвольные словообразования;

– применять индексы стандартов (ГОСТ, ГОСТ Р, ОСТ и т.п.), технических условий (ТУ), строительных норм и правил (СНиП) и других документов без регистрационного номера;

– использовать в тексте математические знаки и знак \emptyset (диаметр), а также знаки № (номер) и % (процент) без числовых значений. Следует писать: «температура 20 °С»; «номер опыта» (но не « № опыта»); «влажность 98 %», «процент выхода» (но не « % выхода»).

Условные буквенные обозначения, изображения или знаки должны соответствовать принятым в действующем законодательстве и государственных стандартах.

В ТД следует применять стандартизированные единицы физических величин, их наименования и обозначения.

5 Оформление иллюстраций и таблиц

Оформление иллюстраций и таблиц курсового проекта должно строго соответствовать:

–СМК-О-СМГТУ-42-09 Курсовая работа (проект): структура, содержание, общие правила выполнения и оформления;

–СМК-К-РИ-109-15 Порядок организации выполнения и защиты курсовой работы (проекта) в многопрофильном колледже.

Таблицы применяют для лучшей наглядности и удобства сравнения показателей и, как правило, оформляются в соответствии с рисунком 7.



Рисунок 7 - Пример оформления таблицы

Таблица помещается в тексте сразу же за первым упоминанием о ней или на следующей странице.

Таблицы, за исключением приведенных в приложении, нумеруются в пределах каждого раздела. В этом случае номер таблицы состоит из номера раздела и порядкового номера таблицы в пределах раздела, разделенных точкой. Допускается сквозная нумерация таблиц арабскими цифрами по всему ТД. Таблицы каждого приложения обозначают отдельной нумерацией арабскими цифрами с добавлением перед цифрой обозначения приложения.

Если в тексте одна таблица, то она должна быть обозначена «Таблица 1» или «Таблица В.1», если она приведена в приложении В.

Название таблицы, при его наличии, должно отражать ее содержание, быть точным, кратким. Название таблицы помещают над таблицей после ее номера через тире, с прописной буквы (остальные строчные), без абзацного отступа. Надпись: «Таблица...» пишется над левым верхним углом таблицы и выполняется строчными буквами (кроме первой прописной) без подчеркивания (рисунок 1).

Заголовки граф таблицы выполняют с прописных букв, а подзаголовки - со строчных, если они составляют одно предложение с заголовком, и с прописной - если они самостоятельные. В конце заголовка и подзаголовка знаки препинания не ставятся. Заголовки указываются в единственном числе. Допускается применять размер шрифта в таблице меньший, чем в тексте. Диагональное деление головки таблицы не допускается.

Таблицу с большим количеством строк допускается переносить на другую страницу. При переносе части таблицы на другую страницу заголовок помещают только перед первой частью таблицы, над другими частями справа пишется слово «Продолжение» и указывается порядковый номер таблицы, например: «Продолжение таблицы 2.7». Нижнюю горизонтальную черту, ограничивающую таблицу, не проводят.

Таблицы с большим количеством граф допускается делить на части и помещать одну под другой на одном листе. Над последующими частями таблиц указывается слово: «Продолжение», а при наличии нескольких таблиц в ТД указывается номер таблицы, например: «Продолжение таблицы 2.3».

Если строки и графы таблицы выходят за формат страницы, то в первом случае в каждой части таблицы повторяется головка, во втором случае – боковик.

Если цифровые данные в пределах графы таблицы выражены в одних единицах физической величины, то они указываются в заголовке каждой графы. Включать в таблицу отдельную графу «Единицы измерений» не допускается. Числовые значения величин, одинаковые для нескольких строк, допускается указывать один раз.

Количество иллюстраций, помещаемых в ТД, должно быть достаточным для раскрытия содержания работы. Иллюстрации (чертежи, графики, схемы, компьютерные распечатки, диаграммы, фотоснимки и т.п.) следует располагать непосредственно после первого упоминания в

тексте, или на следующей странице. Иллюстрации могут быть в компьютерном исполнении, в том числе и цветные.

Все иллюстрации именуется в тексте рисунками и нумеруются в пределах каждого раздела. Номер иллюстрации составляется из номера раздела и порядкового номера иллюстрации в пределах данного раздела, разделенных точкой, например: «рисунок 5.1» (первый рисунок пятого раздела). Допускается сквозная нумерация рисунков арабскими цифрами по всему ТД. Если рисунок один, то он обозначается «Рисунок 1». Иллюстрации каждого приложения обозначают отдельной нумерацией арабскими цифрами с добавлением перед цифрой обозначения приложения. Например, «Рисунок А.3».

На все иллюстрации должны быть даны ссылки в ТД. При ссылках на иллюстрации следует писать «... в соответствии с рисунком 2» при сквозной нумерации и «... в соответствии с рисунком 1.2» при нумерации в пределах раздела.

Иллюстрации, помещаемые в ТД, должны соответствовать требованиям государственных стандартов Единой системы конструкторской документации (ЕСКД).

Иллюстрация располагается по тексту документа, если она помещается на листе формата А4. Если формат иллюстрации больше А4, то ее следует помещать в приложении. Фотоснимки размером меньше формата А4 должны быть наклеены на стандартные листы белой бумаги. Иллюстрации следует размещать так, чтобы их можно было рассматривать без поворота документа или с поворотом по часовой стрелке.

Иллюстрации следует выполнять на той же бумаге, что и текст, либо на кальке того же формата с соблюдением тех же полей, что и для текста. При этом кальку с иллюстрацией следует помещать на лист белой непрозрачной бумаги.

Иллюстрации, при необходимости, могут иметь наименование, например: «Рисунок В.2 - Схема алгоритма» и пояснительные данные (подрисовочный текст). Слово «рисунок», его номер и наименование помещают ниже изображения после пояснительных данных симметрично иллюстрации.

Формулы следует выделять из текста в отдельную строку. Выше и ниже каждой формулы или уравнения должно быть оставлено не менее одной свободной строки.

Формулы должны приводиться в общем виде с расшифровкой входящих в них буквенных значений. Буквы греческого, латинского алфавитов и цифры следует выполнять с помощью компьютерного набора курсивом или чертежным шрифтом, в соответствии с требованиями стандартов ЕСКД, высота букв и цифр при компьютерном наборе должна быть на 2 пт больше, чем в основном тексте работы.

Если уравнение или формула не вмещается в одну строку, то оно должно быть перенесено после знака равенства «=» или после знаков плюс «+», минус «-», умножения «х», деления «:», или других математических знаков, причем этот знак в начале следующей строки повторяют. При переносе формулы на знаке, символизирующем операцию умножения, применяют знак «х».

Пояснение значения символов и числовых коэффициентов, входящих в формулу, должны быть приведены непосредственно под формулой. Значение каждого символа дают в той последовательности, в какой они приведены в формуле. Первая строка расшифровки должна начинаться со слова «где» без двоеточия после него.

Формулы, следующие одна за другой и не разделенные текстом, отделяют запятой.

Формулы, за исключением приведенных в приложении, должны нумероваться в пределах всего ТД арабскими цифрами в круглых скобках в крайнем правом положении на строке.

Формулы, помещаемые в приложениях, обозначают отдельной нумерацией арабскими цифрами в пределах каждого приложения с добавлением перед каждой цифрой обозначения приложения, например формула (В.1). Ссылки в тексте на порядковые номера формул дают в скобках.

6 Список использованных источников

1. Гагарина, Л. Г. Технология разработки программного обеспечения [Электронный ресурс] : учебное пособие / Л. Г. Гагарина, Е. В. Кокорева, Б. Д. Сидорова-Виснадул ; под ред. Л. Г. Гагариной. — Москва : ИД «ФОРУМ» : ИНФРА-М, 2019. — 400 с. — Режим доступа: <https://new.znaniium.com/read?id=336552>
2. Гагарина, Л. Г. Введение в архитектуру программного обеспечения [Электронный ресурс] : учебное пособие / Л. Г. Гагарина, А. Р. Федоров, П. А. Федоров. — Москва : ФОРУМ : ИНФРА-М, 2018. — 320 с. — Режим доступа: <https://new.znaniium.com/read?id=315623>
3. Черткова, Е. А. Программная инженерия. Визуальное моделирование программных систем [Электронный ресурс] : учебник для среднего профессионального образования / Е. А. Черткова. — 2-е изд., испр. и доп. — Москва : Издательство Юрайт, 2019. — 147 с. — (Профессиональное образование). — ISBN 978-5-534-09823-5. — Режим доступа: <https://www.biblio-online.ru/bcode/441255>
4. Кундышева, Е. С. Математические методы и модели в экономике [Электронный ресурс] : учебник для бакалавров / Кундышева Е. С.; Под ред. Сулаков Б. А. - Москва : Дашков и К, 2017. - 286 с. ISBN 978-5-394-02488-7 - Режим доступа: <https://new.znaniium.com/read?id=300594>
5. Новиков, А. И. Экономико-математические методы и модели [Электронный ресурс] : учебник для бакалавров / А. И. Новиков. - Москва : Дашков и К, 2017. - 532 с. ISBN 978-5-394-02615-7 - Режим доступа: <https://new.znaniium.com/read?id=300712>
6. Федорова, Г. Н. Разработка, внедрение и адаптация программного обеспечения отраслевой направленности [Электронный ресурс] : учебное пособие / Г. Н. Федорова. — М. : КУРС : ИНФРА-М, 2019. — 336 с. (Среднее Профессиональное Образование). - Режим доступа: <https://new.znaniium.com/read?id=330691>
7. Хорев, П. Б. Объектно-ориентированное программирование с примерами на С# [Электронный ресурс] : учебное пособие / П. Б. Хорев. — Москва : ФОРУМ : ИНФРА-М, 2019. — 200 с. — Режим доступа: <https://new.znaniium.com/read?id=339308>
8. Гетманчук, А. В. Экономико-математические методы и модели [Электронный ресурс] : учебное пособие / Гетманчук А. В., Ермилов М. М. - Москва : Дашков и К, 2017. - 186 с.: ISBN 978-5-394-01575-5 - Режим доступа: <https://new.znaniium.com/read?id=108077>
9. Интуит Национальный открытый университет курс Язык UML 2 в анализе и проектировании программных систем и бизнес-процессов [Электронный ресурс]. – Режим доступа: <http://www.intuit.ru/studies/courses/480/336/info>, свободный.– Загл. с экрана. Яз. рус.3.3 Учебно-методическое обеспечение самостоятельной работы обучающихся
10. Интуит Национальный открытый университет курс Проектирование информационных систем в Microsoft SQL Server 2008 и Visual Studio 2008 [Электронный ресурс]. – Режим доступа: <http://www.intuit.ru/studies/courses/502/358/info>, свободный.– Загл. с экрана. Яз. рус.

7 Защита курсового проекта

В процессе подготовки к защите студент готовит доклад на 5-10 минут. В докладе должно быть раскрыто содержание курсового проекта, раскрыты главные положения, больше половины доклада должно быть посвящено практической части, заканчивается доклад выводами и предложениями.

Защита курсового проекта осуществляется перед комиссией, состоящей из преподавателей.

8 Критерии оценки курсового проекта

Курсовой проект оценивается по пятибалльной системе.

Критериями оценки курсового проекта по профессиональному модулю являются:

- качество содержания работы (достижение сформулированной цели и решение задач исследования, полнота раскрытия темы, системность подхода, отражение умений использования различных источников на основе всевозможных точек зрения по теме, нормативно-правовой информации, аргументированное обоснование выводов и предложений);
- соблюдение графика выполнения курсового проекта;
- обоснование актуальности выбранной темы;
- соответствие содержания выбранной теме;
- соответствие содержания глав и параграфов их названию;
- логика, грамотность и стиль изложения;
- внешний вид работы и ее оформление, аккуратность;
- положительные стороны, а также недостатки в работе;
- соблюдение заданного объема работы;
- наличие хорошо структурированного плана, раскрывающего содержание темы курсового проекта;
- правильность оформления списка использованной литературы;
- достаточность и новизна изученной литературы;
- ответы на вопросы при публичной защите работы.

Оценка **«отлично»** выставляется при выполнении курсового проекта в полном объеме: выполнен анализ и предварительная обработка информации; изучены типовые решения, обосновано и выбрано оптимальное решение; разработано и оформлено техническое задание в полном соответствии с рекомендациями стандартов; разделы технического задания изложены логично и технически грамотно; программное приложение разработано и корректно функционирует в полном соответствии с техническим заданием в среде программирования с использованием открытых библиотек; приложение предварительно смоделировано (применены объектные модели); код оформлен в соответствии со стандартами кодирования; интерфейс пользователя разработан и корректно функционирует в полном соответствии с техническим заданием; приложение предварительно смоделировано (применены объектные модели); использованы анимационные эффекты; код оформлен в соответствии со стандартами кодирования.

Оценка **«хорошо»** выставляется при выполнении курсового проекта в объеме: выполнена предварительная обработка информации; изучены типовые решения, выбрано оптимальное решение; разработано и оформлено техническое задание в соответствии с рекомендациями стандартов; разделы технического задания изложены логично и грамотно; веб приложение разработано и работоспособно в соответствии с техническим заданием в среде программирования с использованием открытых библиотек; приложение предварительно смоделировано; код оформлен в соответствии со стандартами кодирования; интерфейс пользователя разработан и функционирует в соответствии с техническим заданием; приложение предварительно смоделировано; использованы анимационные эффекты; код оформлен в соответствии со стандартами кодирования; работоспособность проверена.

Оценка **«удовлетворительно»** выставляется при выполнении курсового проекта в основном правильно, но без достаточно глубокой проработки некоторых разделов: частично выполнена предварительная обработка информации, выделены основные объекты и атрибуты практически соответствующие заданию; выбрано одно решение; разработано и оформлено техническое задание в соответствии с рекомендациями стандартов; разделы технического задания изложены грамотно; веб приложение разработано и работоспособно в соответствии с техническим заданием в среде программирования с использованием открытых библиотек; код оформлен с незначительными отклонениями от стандартов кодирования; интерфейс пользователя разработан и

функционирует; приложение предварительно смоделировано; использованы анимационные эффекты; код оформлен с незначительными отклонениями от стандартов кодирования.

Оценка **«неудовлетворительно»** выставляется, когда студент не может защитить свои решения, допускает грубые фактические ошибки при ответах на поставленные вопросы или вовсе не отвечает на них.

Положительная оценка выставляется в ведомость и зачетную книжку. Студент, получивший неудовлетворительную оценку, должен доработать курсовой проект. В этом случае смена темы не допускается.

Оценка уровня сформированности профессиональных и общих компетенций во время подготовки и защиты курсового проекта по профессиональному модулю определяется руководителем по универсальной шкале оценки образовательных достижений, которые включают в себя основные показатели оценки результатов

Оценка образовательных достижений обучающегося

Код и наименование компетенций	Код и наименование ОПОР (основных показателей оценки результата)	Оценка (положительная – 1/ отрицательная – 0)		
		Выполнение КП	Защита КП	Интегральная оценка ОПОР как результатов выполнения и защиты КП
ПК 2.1. Разрабатывать требования к программным модулям на основе анализа проектной и технической документации на предмет взаимодействия компонент	ОПОР 2.1.1 Анализ проектной и технической документации			
	ОПОР 2.1.2 Оформление требований к программным модулям			
	ОПОР 2.1.3 Использование специализированных графических средств построения и анализа архитектуры программных продуктов			
ПК 2.2. Выполнять интеграцию модулей в программное обеспечение	ОПОР 2.2.1 Встраивание модулей в программное обеспечение			
	ОПОР 2.2.2 Использование основных подходов к интегрированию программных модулей			
ПК 2.3 Выполнять отладку программного модуля с использованием специализированных программных средств	ОПОР 2.3.1 Выполнение отладки с использованием методов условной компиляции			
	ОПОР 2.3.2 Выполнение отладки с использованием инструментов условной компиляции			
ПК 2.4 Осуществлять разработку тестовых наборов и тестовых сценариев для программного обеспечения.	ОПОР 2.4.5 Выполнение ручного и автоматизированного тестирования программного модуля			
ПК 2.5. Производить инспектирование компонент программного обеспечения на предмет соответствия стандартам кодирования	ОПОР 2.5.1 Выполнение верификации и аттестации программного обеспечения			
ОК 01	ОПОР 01.1 Определяет профессиональную задачу с учетом профессионального и социального контекста			
	ОПОР 01.2 Осуществляет поиск информации, необходимой для решения задачи и/или проблемы.			
	ОПОР 01.3 Составляет план действий для решения задач.			
ОК 02	ОПОР 02.1 Планирует поиск информации в зависимости от поставленных задач в заявленных условиях			
	ОПОР 02.2 Структурирует получаемую информацию			
ОК 03	ОПОР 03.2 Владеет современной научной профессиональной терминологией			
ОК 04	ОПОР 04.2 Взаимодействует с коллегами, руководством, клиентами в ходе профессиональной деятельности			
ОК 05	ОПОР 05.2 Осуществляет устное общение в профессиональной деятельности в соответствии с нормами русского языка			
	ОПОР 05.3 Оформляет документы о профессиональной тематике на государственном языке			
ОК 09	ОПОР 09.1 Использует информационные технологии при решении профессиональных задач			
	ОПОР 09.2 Использует современное программное обеспечение в профессиональной деятельности			
ОК 10	ОПОР 10.1 Осуществляет коммуникацию (устную и письменную) на государственном и иностранном языках			
ОК 11	ОПОР 11.2 Презентует коммерческую идею			

max количество оценок	
количество положительных оценок	
% положительных оценок	
Оценка в универсальной шкале оценок	

Для оценки образовательных достижений обучающихся применяется универсальная шкала

Процент результативности (правильных ответов)	Качественная оценка уровня подготовки	
	балл (отметка)	вербальный аналог
90 ÷ 100	5	отлично
80 ÷ 89	4	хорошо
70 ÷ 79	3	удовлетворительно
менее 70	2	неудовлетворительно

Приложение А
Форма титульного листа

Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное учреждение
высшего образования

«Магнитогорский государственный технический университет
им. Г. И. Носова»

Многопрофильный колледж

ПЦК _____

КУРСОВОЙ ПРОЕКТ
РАСЧЁТНО-ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

По ПМ _____

МДК _____

на тему: _____

Исполнитель: _____ студент _____ курса, группа _____

Руководитель: _____
(Ф.И.О., должность, уч. степень, уч. звание)

Работа допущена к защите “ _____ ” _____ 20__ г. _____
(подпись)

Работа защищена “ _____ ” _____ 20__ г. с оценкой _____
(оценка) (подпись)

Магнитогорск, 20__

Приложение Б

Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное учреждение
высшего образования

«Магнитогорский государственный технический университет
им. Г. И. Носова»

Многопрофильный колледж

КУРСОВОЙ ПРОЕКТ

Тема: _____

Студент _____

Задание

Исходные

данные: _____

Состав и содержание проекта (работы) _____

Срок сдачи: « ____ » _____ 201__ г.

Руководитель: _____ / _____
« ____ » _____ 201__ г.

Задание получил: _____ / _____
« ____ » _____ 201__ г.

Магнитогорск, 20__