

Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Силин Яков Петрович
Должность: Ректор
Дата подписания: 10.06.2026 14:43:29
Уникальный программный ключ:
24f866be2aca16484036a8cbb3c509a9531e605f

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
ФГБОУ ВО «Уральский государственный экономический университет»

Одобрена
на заседании кафедры
09.12.2025 г.

протокол № 12

И.о. зав. кафедрой Кольева Н.С.

Утверждена
Советом по учебно-методическим
вопросам и качеству образования

16 декабря 2025 г.

протокол №

Председатель Карх Д.А.



(подпись)

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

Наименование дисциплины	Разработка многоуровневых приложений
Направление подготовки	09.03.01 Информатика и вычислительная техника
Профиль	Программное обеспечение автоматизированных систем
Форма обучения	очная
Год набора	2026
Разработана:	
Профессор, д.п.н.	
Плещев В.В.	

Екатеринбург
2025 г.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	3
1. ЦЕЛЬ ОСВОЕНИЯ ДИСЦИПЛИНЫ	3
2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОПОП	3
3. ОБЪЕМ ДИСЦИПЛИНЫ	3
4. ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ ОСВОЕНИЯ ОПОП	3
5. ТЕМАТИЧЕСКИЙ ПЛАН	4
6. ФОРМЫ ТЕКУЩЕГО КОНТРОЛЯ И ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ШКАЛЫ ОЦЕНИВАНИЯ	6
7. СОДЕРЖАНИЕ ДИСЦИПЛИНЫ	7
8. ОСОБЕННОСТИ ОРГАНИЗАЦИИ ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА ПО ДИСЦИПЛИНЕ ДЛЯ ЛИЦ С ОГРАНИЧЕННЫМИ ВОЗМОЖНОСТЯМИ ЗДОРОВЬЯ	12
9. ПЕРЕЧЕНЬ ОСНОВНОЙ И ДОПОЛНИТЕЛЬНОЙ УЧЕБНОЙ ЛИТЕРАТУРЫ, НЕОБХОДИМОЙ ДЛЯ ОСВОЕНИЯ ДИСЦИПЛИНЫ	13
10. ПЕРЕЧЕНЬ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, ВКЛЮЧАЯ ПЕРЕЧЕНЬ ЛИЦЕНЗИОННОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ И ИНФОРМАЦИОННЫХ СПРАВОЧНЫХ СИСТЕМ, ОНЛАЙН КУРСОВ, ИСПОЛЬЗУЕМЫХ ПРИ ОСУЩЕСТВЛЕНИИ ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА ПО ДИСЦИПЛИНЕ	14
11. ОПИСАНИЕ МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЙ БАЗЫ, НЕОБХОДИМОЙ ДЛЯ ОСУЩЕСТВЛЕНИЯ ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА ПО ДИСЦИПЛИНЕ	15

ВВЕДЕНИЕ

Рабочая программа дисциплины является частью основной профессиональной образовательной программы высшего образования - программы бакалавриата, разработанной в соответствии с ФГОС ВО

ФГОС ВО	Федеральный государственный образовательный стандарт высшего образования- бакалавриат по направлению подготовки 09.03.01 Информатика и вычислительная техника (приказ Минобрнауки России от 19.09.2017 г. № 929)
---------	--

1. ЦЕЛЬ ОСВОЕНИЯ ДИСЦИПЛИНЫ

Изучение современных методов и средств разработки многоуровневых приложений сайтов, мобильных и Web-приложений.

2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОПОП

Дисциплина относится к части, формируемой участниками образовательных отношений.

3. ОБЪЕМ ДИСЦИПЛИНЫ

Промежуточная аттестация	Часов					З.е.
	Всего за семестр	Контактная работа (поуч.зан.)			Самостоятельная работа в том числе подготовка контрольных и курсовых	
		Всего	Лекции	Лабораторные		
Семестр 5						
Зачет	108	48	24	24	60	3
Семестр 6						
Экзамен, Курсовая работа	180	64	32	32	89	5
	288	112	56	56	149	8

4. ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ ОСВОЕНИЯ ОПОП

В результате освоения ОПОП у выпускника должны быть сформированы компетенции, установленные в соответствии ФГОС ВО.

Шифр и наименование компетенции	Индикаторы достижения компетенций
организационно-управленческий	
ПК-5 Разработка технических спецификаций информационных ресурсов	ИД-1.ПК-5 Знать: Языки формализации функциональных спецификаций; Методы и приемы формализации задач; Архитектура, устройство и принцип функционирования вычислительных систем
	ИД-2.ПК-5 Уметь: Выбирать средства реализации требований ИР Вырабатывать варианты реализации требований к ИР Производить оценку и обоснование рекомендуемых решений Осуществлять коммуникации с заинтересованными сторонами

ПК-5 Разработка технических спецификаций информационных ресурсов	ИД-3.ПК-5 Иметь практический опыт: разработки и согласования технических спецификаций Распределения заданий между программистами в соответствии с техническими спецификациями Осуществления контроля выполнения заданий Формирования и предоставление отчетности в соответствии с установленными регламентами Оценки и согласования сроков выполнения поставленных задач
ПК-6 Проектирование информационных ресурсов	ИД-1.ПК-6 Знать: Принципы построения архитектуры ИР; Типовые решения, библиотеки программных модулей, шаблоны, классы объектов, используемые при разработке ИР; Методы и средства проектирования ИР; Методы и средства проектирования баз данных; Методы и средства проектирования программных интерфейсов ИД-2.ПК-6 Уметь: Использовать существующие типовые решения и шаблоны ИР; Применять методы и средства проектирования ИР, структур данных, баз данных, программных интерфейсов; Осуществлять коммуникации с заинтересованными сторонами ИД-3.ПК-6 Иметь практический опыт: разработки, изменения архитектуры ИР, согласование с системным аналитиком и архитектором; Проектирования структур данных; Проектирования баз данных; Проектирования интерфейсов; Оценки и согласования сроков выполнения поставленных задач
ПК-4 Анализ и формализация требований к информационным ресурсам	ИД-1.ПК-4 Знать: Сетевые протоколы и основы web-технологий; Устройство и функционирование современных информационных ресурсов; Современные стандарты взаимодействия компонентов распределенных приложений; Программные средства и платформы для разработки web-ресурсов; Методики описания и моделирования процессов, средства моделирования процессов; Отраслевая нормативная техническая документация; Основы теории системного анализа и построения диаграмм взаимодействия; Правила деловой переписки ИД-2.ПК-4 Уметь: Производить анализ исполнения требований; Вырабатывать варианты реализации требований; Производить оценки обоснование рекомендуемых решений; Применять методы и приемы формализации задач; Использовать программные продукты для графического отображения алгоритмов; Осуществлять коммуникации с заинтересованными сторонами ИД-3.ПК-4 Иметь практический опыт: проведения интервьюирования заказчика в соответствии с готовой методологией; Составления формализованных описаний решений поставленных задач в соответствии с требованиями принятых в организации нормативных документов; Разработки алгоритмов решения поставленных задач в соответствии с требованиями принятых в организации нормативных документов; Согласование требований к ИР с заинтересованными сторонами; Оценка времени и трудоемкости реализации требований к ИР

5. ТЕМАТИЧЕСКИЙ ПЛАН

Тема	Часов						
	Наименование темы	Всего часов	Контактная работа .(по уч.зан.)			Самост. работа	Контроль самостоятельной работы
			Лекции	Лабораторные	Практические занятия		
Семестр 5		108					
Тема 1.	Веб-технологии (ПК-4,	6	4			2	
Тема 2.	Разработка веб-интерфейсов.	8		2		6	
Тема 3.	Разработка веб-приложений на языке PHP	6		2		4	
Тема 4.	Организация сетевого взаимодействия с базой данных (MySQL) (ПК-4,	6	4	2			
Тема 5.	Публикация данных из MySQL в Интернете (ПК-	18	6	4		8	
Тема 6.	Проектирование реляционных баз данных	14	4	2		8	
Тема 7.	Структурное программирование (ПК-	14	2	4		8	
Тема 8.	Система управления содержимым (ПК-4,ПК-5,	14	2	4		8	
Тема 9.	Форматирование содержимого с помощью регулярных выражений	12	2	2		8	
Тема10.	Куки, сессии и контроль доступа (ПК-4,ПК-5,	10		2		8	
Семестр 6		153					
Тема11.	Администрирование MySQL (ПК-4, ПК-5, ПК-	12	4	2		6	
Тема12.	Расширенные SQL-запросы (ПК-4, ПК-5,ПК-	12	4	2		6	
Тема13.	Бинарные данные (ПК-4, ПК-5, ПК-6)	14	6	4		4	
Тема14.	JavaScript интерактивность	16	4	4		8	
Тема15.	Разработка распределенных	16	4	4		8	
Тема16.	Система управления содержимым CMS(ПК-4,	20	4	6		10	
Тема17.	Мобильный сайт и мобильное приложение	17	2	4		11	
Тема18.	Нативные приложения (ПК-4, ПК-5, ПК-6)	16	2	2		12	
Тема19.	Гибридные приложения (ПК-4, ПК-5, ПК-6)	14		2		12	
Тема20.	Создание мобильных приложений (ПК-4,ПК-5,	16	2	2		12	

6. ФОРМЫ ТЕКУЩЕГО КОНТРОЛЯ И ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ШКАЛЫ ОЦЕНИВАНИЯ

Раздел/Тема	Вид оценочного средства	Описание оценочного средства	Критерии оценивания
Текущий контроль (Приложение 4)			
Темы 1-5	Практическая работа (приложение 4)	Работа состоит из 5-и задач	10 баллов
Темы 6-10	Практическая работа (приложение 4)	Работа состоит из 5-и задач	10 баллов
Тема 11-20	Практическая работа (приложение 4)	Работа состоит из 5-и задач	10 баллов
Промежуточная аттестация (Приложение 5)			
5 семестр (За)	Билет для зачета	Билет содержит 1 теоретический вопрос и 2 практических задания	100 баллов
6 семестр (Эк)	Экзаменационный билет	Билет содержит 1 теоретический вопрос и 2 практических задания	100 баллов
6 семестр (КР)	Курсовая работа	Перечень курсовых работ (Приложение 3), Методические рекомендации по выполнению курсовой работы по дисциплине (Приложение 7).	100 баллов

ОПИСАНИЕ ШКАЛ ОЦЕНИВАНИЯ

Показатель оценки освоения ОПОП формируется на основе объединения текущего контроля и промежуточной аттестации обучающегося.

Показатель рейтинга по каждой дисциплине выражается в процентах, который показывает уровень подготовки студента.

Текущий контроль. Используется 100-балльная система оценивания. Оценка работы студента в течение семестра осуществляется преподавателем в соответствии с разработанной им системой оценки учебных достижений в процессе обучения по данной дисциплине.

В рабочих программах дисциплин и практик закреплены виды текущего контроля, планируемые результаты контрольных мероприятий и критерии оценки учебных достижений.

В течение семестра преподавателем проводится не менее 3-х контрольных мероприятий, по оценке деятельности студента. Если посещения занятий по дисциплине включены в рейтинг, то данный показатель составляет не более 20% от максимального количества баллов по дисциплине.

Промежуточная аттестация. Используется 5-балльная система оценивания. Оценка работы студента по окончании дисциплины (части дисциплины) осуществляется преподавателем в соответствии с разработанной им системой оценки достижений студента в процессе обучения по данной дисциплине. Промежуточная аттестация также проводится по окончании формирования компетенций.

Порядок перевода рейтинга, предусмотренных системой оценивания, по дисциплине, в пятибалльную систему.

Высокий уровень – 100% - 70% - отлично, хорошо.

Средний уровень – 69% - 50% - удовлетворительно.

Показатель оценки	По 5-балльной системе	Характеристика показателя
100% - 85%	отлично	обладают теоретическими знаниями в полном объеме, понимают, самостоятельно умеют применять, исследовать, идентифицировать, анализировать, систематизировать, распределять по категориям, рассчитать показатели, классифицировать, разрабатывать модели, алгоритмизировать, управлять, организовать, планировать процессы исследования, осуществлять оценку результатов на высоком уровне
84% - 70%	хорошо	обладают теоретическими знаниями в полном объеме, понимают, самостоятельно умеют применять, исследовать, идентифицировать, анализировать, систематизировать, распределять по категориям, рассчитать показатели, классифицировать, разрабатывать модели, алгоритмизировать, управлять, организовать, планировать процессы исследования, осуществлять оценку результатов. Могут быть допущены недочеты, исправленные студентом самостоятельно в процессе работы (ответаи т.д.)
69% - 50%	удовлетворительно	обладают общими теоретическими знаниями, умеют применять, исследовать, идентифицировать, анализировать, систематизировать, распределять по категориям, рассчитать показатели, классифицировать, разрабатывать модели, алгоритмизировать, управлять, организовать, планировать процессы исследования, осуществлять оценку результатов на среднем уровне. Допускаются ошибки, которые студент затрудняется исправить самостоятельно.
49 % и менее	неудовлетворительно	обладают не полным объемом общих теоретическими знаниями, не умеют самостоятельно применять, исследовать, идентифицировать, анализировать, систематизировать, распределять по категориям, рассчитать показатели, классифицировать, разрабатывать модели, алгоритмизировать, управлять, организовать, планировать процессы исследования, осуществлять оценку результатов. Не сформированы умения и навыки для
100% - 50%	зачтено	характеристика показателя соответствует «отлично», «хорошо», «удовлетворительно»
49 % и менее	не зачтено	характеристика показателя соответствует «неудовлетворительно»

7. СОДЕРЖАНИЕ ДИСЦИПЛИНЫ

7.1. Содержание лекций

<p>Тема 1. Веб-технологии (ПК-4, ПК-5, ПК-6) Веб–технологии - логическая составляющая Интернет. Эпохи Web. Наполнение и интерфейсресурсов в разные эпохи. Сематический Web.</p>
<p>Тема 4. Организация сетевого взаимодействия с базой данных (MySQL) (ПК-4, ПК-5, ПК-6) Введение в базы данных. Использование приложения phpMyAdmin для выполнения SQL-запросов.Структурированный язык запросов.</p>
<p>Тема 5. Публикация данных из MySQL в Интернете (ПК-4, ПК-5, ПК-6) Создание учетной записи пользователя в MySQL. Подключение к MySQL с помощью PHP.Настройка подключения.</p>
<p>Тема 6. Проектирование реляционных баз данных (ПК-4, ПК-5, ПК-6) Выборка из нескольких таблиц. Простые связи.</p>
<p>Тема 7. Структурное программирование (ПК-4, ПК-5, ПК-6) Подключаемые файлы. Подключение HTML-кода. Подключение PHP-кода.</p>
<p>Тема 8. Система управления содержимым (ПК-4, ПК-5, ПК-6) Главная страница. Создание проекта "Поиск шуток".</p>
<p>Тема 9. Форматирование содержимого с помощью регулярных выражений (ПК-4, ПК-5, ПК-6) Регулярные выражения. Замена текста с помощью регулярных выражений. Выделение в тексте.Абзацы. Гиперссылки.</p>
<p>Тема 11. Администрирование MySQL (ПК-4, ПК-5, ПК-6) Резервное копирование баз данных в MySQL. Резервное копирование базы данных с помощьюphpMyAdmin.</p>
<p>Тема 12. Расширенные SQL-запросы (ПК-4, ПК-5, ПК-6) Оператор левого объединения. Дополнительные источники информации.</p>
<p>Тема 13. Бинарные данные (ПК-4, ПК-5, ПК-6) Полудинамические страницы. Обеспечение загрузки файлов.</p>
<p>Тема 14. JavaScript интерактивность приложений (ПК-4, ПК-5, ПК-6) Что может и не может JavaScript в браузере? «Привет, мир!» на JS.</p>
<p>Тема 15. Разработка распределенных приложений на платформе .NET (ПК-4, ПК-5, ПК-6) Понятие распределённой системы. Определение распределённой системы.</p>
<p>Тема 16. Система управления содержимым CMS (ПК-4, ПК-5, ПК-6) Сайт без кода. Как работает CMS?</p>
<p>Тема 17. Мобильный сайт и мобильное приложение (ПК-4, ПК-5, ПК-6) Интерфейс. Быстродействие. Связь и адаптация.</p>
<p>Тема 18. Нативные приложения (ПК-4, ПК-5, ПК-6) Выбор языка программирования. выбор платформы. Анализ эмуляторов</p>
<p>Тема 20. Создание мобильных приложений (ПК-4, ПК-5, ПК-6) Выбор платформы. Эмуляторы.</p>

7.2 Содержание практических занятий и лабораторных работ

<p>Тема 3. Разработка веб-приложений на языке PHP (ПК-4, ПК-5, ПК-6)</p> <p>Первый PHP скрипт. Комментарии в PHP. Переменные и их типы. Константы. Арифметические операции. Строковые операции. Логические операции. Оператор эквивалентности. Массивы.</p>
<p>Тема 4. Организация сетевого взаимодействия с базой данных (MySQL) (ПК-4, ПК-5, ПК-6)</p> <p>Создание базы данных. Создание таблицы. Добавление данных в таблицу. Вывод сохраненных данных.</p>
<p>Тема 5. Публикация данных из MySQL в Интернете (ПК-4, ПК-5, ПК-6)</p> <p>Добавление информации в базу данных. Удаление информации из базы данных. Отправка SQL-запросов с помощью PHP. Обработка результатов выполнения команды SELECT.</p>
<p>Тема 6. Проектирование реляционных баз данных (ПК-4, ПК-5, ПК-6)</p> <p>Отношение «многие ко многим». Один за многих, и многие за одного</p>
<p>Тема 7. Структурное программирование (ПК-4, ПК-5, ПК-6)</p> <p>Виды включений. Разделение подключаемых файлов. Вспомогательные функции для шаблонов. Нестандартные функции и библиотеки функций.</p>
<p>Тема 8. Система управления содержимым (ПК-4, ПК-5, ПК-6)</p> <p>Управление списком авторов. Удаление имен авторов. Добавление и редактирование имен авторов. Управление списком категорий. Управление списком шуток.</p>
<p>Тема 9. Форматирование содержимого с помощью регулярных выражений (ПК-4, ПК-5, ПК-6)</p> <p>Передача данных в реальных условиях.</p>
<p>Тема 10. Куки, сессии и контроль доступа (ПК-4, ПК-5, ПК-6)</p> <p>Управление паролями и ролями. Модерирование шуток.</p>
<p>Тема 11. Администрирование MySQL (ПК-4, ПК-5, ПК-6)</p> <p>Инкрементальное резервное копирование с помощью бинарного журнала изменений. Управление доступом к MySQL. Проблемы, связанные с именем сервера. Индексы.</p>
<p>Тема 12. Расширенные SQL-запросы (ПК-4, ПК-5, ПК-6)</p> <p>Сортировка результатов выполнения запроса SELECT. Установка лимитов. Транзакции в базе данных. Псевдонимы для столбцов и таблиц. Группирование результатов.</p>

<p>Тема 13. Бинарные данные (ПК-4, ПК-5, ПК-6)</p> <p>Запись загруженных файлов в базу данных. Типы бинарных столбцов. Сохранение файлов. Отображение сохраненных файлов. Размер пакета в MySQL.</p>
<p>Тема 14. JavaScript интерактивность приложений (ПК-4, ПК-5, ПК-6)</p> <p>Современная разметка, Внешние скрипты. Таблицы совместимости. Структура кода. Инструкции. Переменные. Имена переменных. Константы. Типы данных. Взаимодействие: alert, prompt, confirm. Объекты. Литералы и свойства.</p>
<p>Тема 15. Разработка распределенных приложений на платформе .NET (ПК-4, ПК-5, ПК-6)</p> <p>Программные компоненты. Требования к распределенным системам. Понятие промежуточной среды. Взаимодействие компонент распределённой системы.</p>
<p>Тема 16. Система управления содержимым CMS (ПК-4, ПК-5, ПК-6)</p> <p>Плагины, Темы и Расширения. Популярные Системы Управления Контентом. Плюсы и минусы использования CMS. Как правильно выбрать Систему Управления Контентом. Как начать пользоваться CMS.</p>
<p>Тема 17. Мобильный сайт и мобильное приложение (ПК-4, ПК-5, ПК-6)</p> <p>Интеграция с платформой. Фрагментация. Ресурсы. Публикация.</p>
<p>Тема 18. Нативные приложения (ПК-4, ПК-5, ПК-6)</p> <p>Натив преимущества и недостатки. Инструменты для разработки нативных приложение</p>
<p>Тема 19. Гибридные приложения (ПК-4, ПК-5, ПК-6)</p> <p>Гибрид преимущества и недостатки. Инструменты для разработки гибридных приложение</p>
<p>Тема 20. Создание мобильных приложений (ПК-4, ПК-5, ПК-6)</p> <p>Мобильное приложение. Основные этапы создания, компиляция, публикация в маркетплейс</p>

7.3. Содержание самостоятельной работы

<p>Тема 2. Разработка веб-интерфейсов. Языки разметки и стили (ПК-4, ПК-5, ПК-6)</p> <p>Изучение основной и дополнительной литературы по теме, интернет-источников. Разбор лабораторных работ.</p>
<p>Тема 3. Разработка веб-приложений на языке PHP (ПК-4, ПК-5, ПК-6)</p> <p>Изучение основной и дополнительной литературы по теме, интернет-источников. Разбор лабораторных работ.</p>

Тема 5. Публикация данных из MySQL в Интернете (ПК-4, ПК-5, ПК-6) Изучение основной и дополнительной литературы по теме, интернет-источников. Разбор лабораторных работ.
Тема 6. Проектирование реляционных баз данных (ПК-4, ПК-5, ПК-6) Изучение основной и дополнительной литературы по теме, интернет-источников. Разбор лабораторных работ.
Тема 7. Структурное программирование (ПК-4, ПК-5, ПК-6) Изучение основной и дополнительной литературы по теме, интернет-источников. Разбор лабораторных работ.
Тема 8. Система управления содержимым (ПК-4, ПК-5, ПК-6) Изучение основной и дополнительной литературы по теме, интернет-источников. Разбор лабораторных работ.
Тема 9. Форматирование содержимого с помощью регулярных выражений (ПК-4, ПК-5, ПК-6) Изучение основной и дополнительной литературы по теме, интернет-источников. Разбор лабораторных работ.
Тема 10. Куки, сессии и контроль доступа (ПК-4, ПК-5, ПК-6) Изучение основной и дополнительной литературы по теме, интернет-источников. Разбор лабораторных работ.
Тема 11. Администрирование MySQL (ПК-4, ПК-5, ПК-6) Изучение основной и дополнительной литературы по теме, интернет-источников. Разбор лабораторных работ.
Тема 12. Расширенные SQL-запросы (ПК-4, ПК-5, ПК-6) Изучение основной и дополнительной литературы по теме, интернет-источников. Разбор лабораторных работ.
Тема 13. Бинарные данные (ПК-4, ПК-5, ПК-6) Изучение основной и дополнительной литературы по теме, интернет-источников. Разбор лабораторных работ.
Тема 14. JavaScript интерактивность приложений (ПК-4, ПК-5, ПК-6) Изучение основной и дополнительной литературы по теме, интернет-источников. Разбор лабораторных работ.
Тема 15. Разработка распределенных приложений на платформе .NET (ПК-4, ПК-5, ПК-6) Изучение основной и дополнительной литературы по теме, интернет-источников. Разбор лабораторных работ.
Тема 16. Система управления содержимым CMS (ПК-4, ПК-5, ПК-6) Изучение основной и дополнительной литературы по теме, интернет-источников. Разбор лабораторных работ.
Тема 17. Мобильный сайт и мобильное приложение (ПК-4, ПК-5, ПК-6) Изучение основной и дополнительной литературы по теме, интернет-источников. Разбор лабораторных работ.

<p>Тема 18. Нативные приложения (ПК-4, ПК-5, ПК-6) Изучение основной и дополнительной литературы по теме, интернет-источников. Разбор лабораторных работ.</p>
<p>Тема 19. Гибридные приложения (ПК-4, ПК-5, ПК-6) Изучение основной и дополнительной литературы по теме, интернет-источников. Разбор лабораторных работ.</p>
<p>Тема 20. Создание мобильных приложений (ПК-4, ПК-5, ПК-6) Изучение основной и дополнительной литературы по теме, интернет-источников. Разбор лабораторных работ.</p>

7.3.1. Примерные вопросы для самостоятельной подготовки к зачету/экзамену
Приложение 1.

7.3.2. Практические задания по дисциплине для самостоятельной подготовки к зачету/экзамену
Приложение 2.

7.3.3. Перечень курсовых работ
Приложение 3.

7.4. Электронное портфолио обучающегося
Размещается курсовая работа.

7.5. Методические рекомендации по выполнению контрольной работы
Не предусмотрено.

7.6 Методические рекомендации по выполнению курсовой работы
Приложение 7.

8. ОСОБЕННОСТИ ОРГАНИЗАЦИИ ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА ПО ДИСЦИПЛИНЕ ДЛЯ ЛИЦ С ОГРАНИЧЕННЫМИ ВОЗМОЖНОСТЯМИ ЗДОРОВЬЯ

По заявлению студента

В целях доступности освоения программы для лиц с ограниченными возможностями здоровья при необходимости кафедра обеспечивает следующие условия:

- особый порядок освоения дисциплины, с учетом состояния их здоровья;
- электронные образовательные ресурсы по дисциплине в формах, адаптированных к ограничениям их здоровья;
- изучение дисциплины по индивидуальному учебному плану (вне зависимости от формы обучения);
- электронное обучение и дистанционные образовательные технологии, которые предусматривают возможности приема-передачи информации в доступных для них формах.
- доступ (удаленный доступ), к современным профессиональным базам данных и информационным справочным системам, состав которых определен РПД.

9. ПЕРЕЧЕНЬ ОСНОВНОЙ И ДОПОЛНИТЕЛЬНОЙ УЧЕБНОЙ ЛИТЕРАТУРЫ, НЕОБХОДИМОЙ ДЛЯ ОСВОЕНИЯ ДИСЦИПЛИНЫ

Сайт библиотеки УрГЭУ

<http://lib.usue.ru/>

Основная литература:

2. Сысолетин Е. Г., Ростунцев С. Д., Доросинский Л. Г. Разработка интернет-приложений [Электронный ресурс]: учебное пособие для вузов. - Москва: Юрайт, 2023. - 90 – Режим доступа: <https://urait.ru/bcode/514303>

3. Тузовский А. Ф. Проектирование и разработка web-приложений [Электронный ресурс]: учебное пособие для вузов. - Москва: Юрайт, 2023. - 219 – Режим доступа: <https://urait.ru/bcode/530767>

4. Мартишин С.А., Симонов В.Л., Храпченко М.В. Базы данных: Работа с распределенными базами данных и файловыми системами на примере MongoDB и HDFS с использованием Node.js, Express.js, Apache Spark и Scala [Электронный ресурс]: Учебное пособие. - Москва: ООО "Научно-издательский центр ИНФРА-М", 2024. - 235 – Режим доступа: <https://znanium.com/catalog/product/2139860>

5. Тузовский А. Ф. Проектирование и разработка web-приложений [Электронный ресурс]: учебное пособие для вузов. - Москва: Юрайт, 2024. - 219 – Режим доступа: <https://urait.ru/bcode/537106>

6. Панов Разработка многоуровневых приложений. Тесты. Тест 3. Презентационный уровень уровня бизнес-логики [Электронный ресурс]:. - Екатеринбург: [б. и.], 2024. - – Режим доступа: <http://lib.wbstatic.usue.ru/202406a/90.docx>

7. Панов Разработка многоуровневых приложений. Тесты. Тест 2. Ключевая роль SQL в многоуровневых приложениях [Электронный ресурс]:. - Екатеринбург: [б. и.], 2024. - – Режим доступа: <http://lib.wbstatic.usue.ru/202406a/89.docx>

8. Панов Разработка многоуровневых приложений. Курс лекций. Тема 1. Обзор общих принципов построения многоуровневых приложений [Электронный ресурс]:. - Екатеринбург: [б. и.], 2024. - 1 – Режим доступа: <https://libw.usue.ru/202408/133.mp4>

9. Панов Разработка многоуровневых приложений. Курс лекций. Тема 2. Ключевая роль SQL в многоуровневых приложениях [Электронный ресурс]:. - Екатеринбург: [б. и.], 2024. - 1 – Режим доступа: <https://libw.usue.ru/202408/134.mp4>

10. Панов Разработка многоуровневых приложений. Тесты. Тест 1. Обзор общих принципов построения многоуровневых приложений [Электронный ресурс]:. - Екатеринбург: [б. и.], 2024. - – Режим доступа: <http://lib.wbstatic.usue.ru/202406a/88.docx>

11. Панов Разработка многоуровневых приложений. Курс лекций. Тема 3. Презентационный уровень и уровень бизнес-логики [Электронный ресурс]:. - Екатеринбург: [б. и.], 2024. - 1 – Режим доступа: <https://libw.usue.ru/202408/135.mp4>

12. Ткаченко О.Н. Взаимодействие пользователя с интерфейсами информационных систем для мобильных устройств: исследование опыта [Электронный ресурс]: Учебное пособие. - Москва: ООО "Научно-издательский центр ИНФРА-М", 2026. - 152 – Режим доступа: <https://znanium.com/catalog/product/2178768>

Дополнительная литература:

1. Кислицын Е. В., Шишков Е. И. Разработка приложений на языке Java [Электронный ресурс]: учебное пособие. - Екатеринбург: [Издательство УрГЭУ], 2017. - 86 – Режим доступа: <http://lib.usue.ru/resource/limit/ump/17/p488938.pdf>

2. Зубкова Е. В., Лескова Ю. В., Федоров А. Н. Принципы построения и архитектура web-приложений [Электронный ресурс]: учебное пособие. - Екатеринбург: [Издательство УрГЭУ], 2017. - 160 – Режим доступа: <http://lib.usue.ru/resource/limit/ump/18/p490704.pdf>

3. Никсон Р., Вильчинский Н. Создаем динамические веб-сайты с помощью PHP, MySQL, JavaScript, CSS и HTML5: производственно-практическое издание. - Санкт-Петербург [и др.]: Питер, 2017. - 766

4. Мартишин С.А., Симонов В.Л. Базы данных: Работа с распределенными базами данных и файловыми системами на примере MongoDB и HDFS с использованием Node.js, Express.js, Apache Spark и Scala [Электронный ресурс]: Учебное пособие. - Москва: ООО "Научно-издательский центр ИНФРА-М", 2021. - 235 – Режим доступа: <https://znanium.com/catalog/product/1214862>

5. Кузин А. В., Чумакова Е. В. Программирование на языке Си [Электронный ресурс]: справочник: учебное пособие : учебное пособие. - Москва: Издательство "ФОРУМ", 2023. - 143 – Режим доступа: <https://znanium.ru/catalog/product/1878382>

10. ПЕРЕЧЕНЬ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, ВКЛЮЧАЯ ПЕРЕЧЕНЬ ЛИЦЕНЗИОННОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ И ИНФОРМАЦИОННЫХ СПРАВОЧНЫХ СИСТЕМ, ОНЛАЙН КУРСОВ, ИСПОЛЬЗУЕМЫХ ПРИ ОСУЩЕСТВЛЕНИИ ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА ПО ДИСЦИПЛИНЕ

Перечень лицензионного программного обеспечения:

Microsoft Visual Studio Community. Лицензия для образовательных учреждений. Срок действия лицензии - без ограничения срока.

PostgreSQL Server. Лицензия PostgreSQL. Срок действия лицензии - без ограничения срока.

Putty. Лицензия MIT license. Срок действия лицензии - без ограничения срока.

WinSCP. Лицензия GNU General Public License. Срок действия лицензии - без ограничения срока.

Notepad++. Лицензия GNU General Public License. Срок действия лицензии - без ограничения срока.

Перечень информационных справочных систем, ресурсов информационно-телекоммуникационной сети «Интернет»:

Справочно-правовая система Гарант. Договор № 58419 от 22 декабря 2015. Срок действия лицензии - без ограничения срока

Справочно-правовая система Консультант+. Договор № 143/223-У/2025 от 02.12.2025 Срок действия лицензии до 31.12.2026

Веб-программирование

<https://openedu.ru/course/ITMOUniversity/WEBDEV/>

Разработка Android-приложений для мобильных устройств

<https://openedu.ru/course/ITMOUniversity/ANDROID/>

11. ОПИСАНИЕ МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЙ БАЗЫ, НЕОБХОДИМОЙ ДЛЯ ОСУЩЕСТВЛЕНИЯ ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА ПО ДИСЦИПЛИНЕ

Реализация учебной дисциплины осуществляется с использованием материально-технической базы УрГЭУ, обеспечивающей проведение всех видов учебных занятий и научно-исследовательской и самостоятельной работы обучающихся:

Специальные помещения представляют собой учебные аудитории для проведения всех видов занятий, групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации.

Помещения для самостоятельной работы обучающихся оснащены компьютерной техникой с возможностью подключения к сети "Интернет" и обеспечением доступа в электронную информационно-образовательную среду УрГЭУ.

Все помещения укомплектованы специализированной мебелью и оснащены мультимедийным оборудованием спецоборудованием (информационно-телекоммуникационным, иным компьютерным), доступом к информационно-поисковым, справочно-правовым системам, электронным библиотечным системам, базам данных действующего законодательства, иным информационным ресурсам служащими для представления учебной информации большой аудитории.

Для проведения занятий лекционного типа презентации и другие учебно-наглядные пособия, обеспечивающие тематические иллюстрации.

7.3.1. Примерные вопросы для самостоятельной подготовки к зачету/экзамену

К зачету

1. Что такое PHP?
2. Как изолировать PHP код от HTML?
3. Как использовать комментарии в коде PHP?
4. В чем разница между одинарными и двойными кавычками в PHP?
5. Сколько в PHP типов данных? Какие это типы данных?
6. Что такое переменная и как ее использовать?
7. Что такое переменные извне?
8. В чем разница между GET и POST?.
9. Что такое константы? Как их использовать?
10. Что такое оператор?
11. Какие существуют операторы?
12. Что такое арифметические операторы?
13. Что такое логические операторы?
14. Что такое оператор присваивания?
15. Что такое побитовые операторы?
16. Что такое операторы сравнения?
17. Что такое оператор управления ошибками?
18. Что такое операторы исполнения?
19. Что такое строковые операторы?
20. В чем различия между echo и print?

К экзамену

1. В чем отличие цикла while от do while?
2. Что такое goto и как он работает?
3. Как прервать выполнение for, foreach, while, do-while или switch?
4. Как пропустить часть интерации цикла внутри for, foreach, while, do-while или switch?
5. Как удалить переменную или элемент из массива?
6. Что такое функция?
7. Что такое аргументы функции?
8. Какие существуют способы передачи аргументов в функцию?
9. Как происходит передача аргументов в функцию по значению?
10. Как происходит передача аргументов в функцию по ссылке?
11. Что такое исключения?
12. Что такое ссылка?
13. Что делают ссылки?
14. Что не делают ссылки?
15. Что такое Cookie и зачем они используются?
16. Что нельзя хранить в Cookie и почему?
17. Какие есть типы cookie?
18. Что такое сессии и зачем они используются?

7.3.2. Практические задания по дисциплине для самостоятельной подготовки к зачету/экзамену

Примерные вопросы закрытого типа (ПК-4)

1. Что такое многоуровневая архитектура приложений?
 1. Архитектура, которая состоит из нескольких уровней, каждый из которых отвечает за определенные функции.
 2. Архитектура, которая состоит из одного уровня, объединяющего все функции приложения.
 3. Архитектура, которая не имеет четкого разделения на уровни.
2. Какие преимущества предоставляет многоуровневая архитектура приложений?
 1. Улучшенная масштабируемость и поддерживаемость.
 2. Уменьшенная сложность и увеличенная производительность.
 3. Отсутствие необходимости в разделении функций на уровни.
3. Какие уровни обычно присутствуют в многоуровневой архитектуре приложений?
 1. Представление (интерфейс), бизнес-логика, доступ к данным.
 2. Только один уровень, объединяющий все функции.
 3. Только два уровня: клиентский и серверный.
4. Что такое уровень представления (интерфейса) в многоуровневой архитектуре?
 1. Уровень, отвечающий за взаимодействие с пользователем.
 2. Уровень, где располагается бизнес-логика приложения.
 3. Уровень, отвечающий за доступ к данным.
5. Какой уровень в многоуровневой архитектуре отвечает за обработку бизнес-логики?
 1. Уровень бизнес-логики.
 2. Уровень доступа к данным.
 3. Уровень представления (интерфейса).

Примерные вопросы открытого типа (ПК-5, 6)

1. Как вы понимаете концепцию многоуровневой архитектуры приложений и почему она важна для разработки программного обеспечения?
2. Расскажите о принципах разделения функциональности на уровни в многоуровневой архитектуре приложений и какие преимущества это может принести для проекта.
3. Какие вызовы и проблемы могут возникнуть при проектировании и реализации многоуровневой архитектуры приложений, и как их можно решить?
4. Какие критерии и принципы следует учитывать при выборе технологий и инструментов для каждого уровня многоуровневой архитектуры приложений?

5. Какие сценарии использования и примеры проектов могут быть наилучшими для применения многоуровневой архитектуры приложений, и какие уроки могут быть извлечены из таких проектов?

Примерные практические задания к зачету

1. Создайте переменную \$a и присвойте ей значение 3. Выведите значение этой переменной на экран.
2. Создайте переменные \$a=10 и \$b=2. Выведите на экран их сумму, разность, произведение и частное (результат деления).
3. Создайте переменные \$c=15 и \$d=2. Просуммируйте их, а результат присвойте переменной \$result. Выведите на экран значение переменной \$result.
4. Создайте переменные \$a=10, \$b=2 и \$c=5. Выведите на экран их сумму.
5. Создайте переменные \$a=17 и \$b=10. Отнимите от \$a переменную \$b и результат присвойте переменной \$c. Затем создайте переменную \$d, присвойте ей значение 7. Сложите переменные \$c и \$d, а результат запишите в переменную \$result. Выведите на экран значение переменной \$result.
6. Создайте переменную \$text и присвойте ей значение 'Привет, Мир!'. Выведите значение этой переменной на экран.
7. Создайте переменные \$text1='Привет, ' и \$text2='Мир!'. С помощью этих переменных и операции сложения строк выведите на экран фразу 'Привет, Мир!'.
8. Создайте переменную \$name и присвойте ей ваше имя. Выведите на экран фразу 'Привет, %Имя%!'. Вместо %Имя% должно стоять ваше имя.
9. Создайте переменную \$age и присвойте ей ваш возраст. Выведите на экран 'Мне %Возраст% лет!'.
10. Создайте переменную \$text и присвойте ей значение 'abcde'. Обращаясь к отдельным символам этой строки выведите на экран символ 'a', символ 'c', символ 'e'.
11. Дана произвольная строка, например, 'abcde'. Поменяйте первую букву (то есть букву 'a') этой строки на '!'.
12. Создайте переменную \$num и присвойте ей значение '12345'. Найдите сумму цифр этого числа.
13. Создайте массив \$arr=['a', 'b', 'c']. Выведите значение массива на экран с помощью функции var_dump().
14. С помощью массива \$arr из предыдущего номера выведите на экран содержимое первого, второго и третьего элементов.
15. Создайте массив \$arr=['a', 'b', 'c', 'd'] и с его помощью выведите на экран строку 'a+b, c+d'.
16. Создайте массив \$arr с элементами 2, 5, 3, 9. Умножьте первый элемент массива на второй, а третий элемент на четвертый. Результаты сложите, присвойте переменной \$result. Выведите на экран значение этой переменной.
17. Если переменная \$a равна нулю, то выведите 'Верно', иначе выведите 'Неверно'. Проверьте работу скрипта при \$a, равном 1, 0, -3.
18. Если переменная \$a больше нуля, то выведите 'Верно', иначе выведите 'Неверно'. Проверьте работу скрипта при \$a, равном 1, 0, -3.
19. Если переменная \$a меньше нуля, то выведите 'Верно', иначе выведите 'Неверно'. Проверьте работу скрипта при \$a, равном 1, 0, -3.
20. Если переменная \$a больше или равна нулю, то выведите 'Верно', иначе выведите 'Неверно'. Проверьте работу скрипта при \$a, равном 1, 0, -3.
21. Если переменная \$a меньше или равна нулю, то выведите 'Верно', иначе выведите 'Неверно'. Проверьте работу скрипта при \$a, равном 1, 0, -3.
22. Если переменная \$a не равна нулю, то выведите 'Верно', иначе выведите 'Неверно'. Проверьте работу скрипта при \$a, равном 1, 0, -3.
23. Если переменная \$a равна 'test', то выведите 'Верно', иначе выведите 'Неверно'. Проверьте работу скрипта при \$a, равном 'test', 'тест', 3.

24. Если переменная \$a равна '1' и по значению и по типу, то выведите 'Верно', иначе выведите 'Неверно'. Проверьте работу скрипта при \$a, равном '1', 1, 3.
25. Если переменная \$a пустая, то выведите 'Верно', иначе выведите 'Неверно'. Проверьте работу скрипта при \$a, равном 1, 3, -3, 0, null, true, "", '0'.
26. Если переменная \$a НЕ пустая, то выведите 'Верно', иначе выведите 'Неверно'.
27. Если переменная \$a существует, то выведите 'Верно', иначе выведите 'Неверно'. Проверьте работу скрипта при \$a, равном 3 и null.
28. Если переменная \$a НЕ существует, то выведите 'Верно', иначе выведите 'Неверно'.
29. Если переменная \$var равна true, то выведите 'Верно', иначе выведите 'Неверно'. Проверьте работу скрипта при \$var, равном true, false. Напишите два варианта скрипта - с короткой записью и с длинной.
30. Если переменная \$var НЕ равна true, то выведите 'Верно', иначе выведите 'Неверно'. Проверьте работу скрипта при \$var, равном true, false. Напишите два варианта скрипта - с короткой записью и с длинной.
31. Если переменная \$a больше нуля и меньше 5-ти, то выведите 'Верно', иначе выведите 'Неверно'. Проверьте работу скрипта при \$a, равном 5, 0, -3, 2.
32. Если переменная \$a равна нулю или равна двум, то прибавьте к ней 7, иначе поделите ее на 10. Выведите новое значение переменной на экран. Проверьте работу скрипта при \$a, равном 5, 0, -3, 2.
33. Если переменная \$a равна или меньше 1, а переменная \$b больше или равна 3, то выведите сумму этих переменных, иначе выведите их разность (результат вычитания). Проверьте работу скрипта при \$a и \$b, равном 1 и 3, 0 и 6, 3 и 5.
34. Если переменная \$a больше 2-х и меньше 11-ти, или переменная \$b больше или равна 6-ти и меньше 14-ти, то выведите 'Верно', в противном случае выведите 'Неверно'.
35. Переменная \$num может принимать одно из значений: 1, 2, 3 или 4. Если она имеет значение '1', то в переменную \$result запишем 'зима', если имеет значение '2' – 'лето' и так далее. Решите задачу через switch-case.
36. В переменной \$day лежит какое-то число из интервала от 1 до 31. Определите в какую декаду месяца попадает это число (в первую, вторую или третью).
37. В переменной \$month лежит какое-то число из интервала от 1 до 12. Определите в какую пору года попадает этот месяц (зима, лето, весна, осень).
38. В переменной \$year хранится год. Определите, является ли он високосным (в таком году есть 29 февраля). Год будет високосным в двух случаях: либо он делится на 4, но при этом не делится на 100, либо делится на 400. Так, годы 1700, 1800 и 1900 не являются високосными, так как они делятся на 100 и не делятся на 400. Годы 1600 и 2000 - високосные, так как они делятся на 400.
39. Дана строка с символами, например, 'abcde'. Проверьте, что первым символом этой строки является буква 'a'. Если это так - выведите 'да', в противном случае выведите 'нет'.
40. Дана строка с цифрами, например, '12345'. Проверьте, что первым символом этой строки является цифра 1, 2 или 3. Если это так - выведите 'да', в противном случае выведите 'нет'.
41. Дана строка из 3-х цифр. Найдите сумму этих цифр. То есть сложите как числа первый символ строки, второй и третий.
42. Дана строка из 6-ти цифр. Проверьте, что сумма первых трех цифр равняется сумме вторых трех цифр. Если это так - выведите 'да', в противном случае выведите 'нет'.
43. Дан массив с элементами 2, 5, 9, 15, 0, 4. С помощью цикла foreach и оператора if выведите на экран столбец тех элементов массива, которые больше 3-х, но меньше 10.
44. Дан массив с числами. Числа могут быть положительными и отрицательными. Найдите сумму положительных элементов этого массива.

45. Дан массив с элементами 1, 2, 5, 9, 4, 13, 4, 10. С помощью цикла `foreach` и оператора `if` проверьте есть ли в массиве элемент со значением, равным 4. Если есть - выведите на экран 'Есть!' и выйдите из цикла. Если нет - ничего делать не надо.
46. Дан массив числами, например: `['10', '20', '30', '50', '235', '3000']`. Выведите на экран только те числа из массива, которые начинаются на цифру 1, 2 или 5.
47. Дан массив с элементами 1, 2, 3, 4, 5, 6, 7, 8, 9. С помощью цикла `foreach` создайте строку `'-1-2-3-4-5-6-7-8-9-'`.
48. Составьте массив дней недели. С помощью цикла `foreach` выведите все дни недели, а выходные дни выведите жирным.
49. Составьте массив дней недели. С помощью цикла `foreach` выведите все дни недели, а текущий день выведите курсивом. Текущий день должен храниться в переменной `$day`.
50. Возведите 2 в 10 степень. Результат запишите в переменную `$st`.
51. Найдите квадратный корень из 245.
52. Дан массив с элементами 4, 2, 5, 19, 13, 0, 10. Найдите корень из суммы квадратов его элементов. Для решения воспользуйтесь циклом `foreach`.
53. Дана строка `'php'`. Сделайте из нее строку `'PHP'`.
54. Дана строка `'PHP'`. Сделайте из нее строку `'php'`.
55. Дана строка `'london'`. Сделайте из нее строку `'London'`.
56. Дана строка `'London'`. Сделайте из нее строку `'london'`.
57. Дана строка `'london is the capital of great britain'`. Сделайте из нее строку `'London Is The Capital Of Great Britain'`.
58. Дана строка `'LONDON'`. Сделайте из нее строку `'London'`.
59. Дана строка `'html css php'`. Найдите количество символов в этой строке.
60. Дана переменная `$password`, в которой хранится пароль пользователя. Если количество символов пароля больше 5-ти и меньше 10-ти, то выведите пользователю сообщение о том, что пароль подходит, иначе сообщение о том, что нужно придумать другой пароль.
61. Дана строка `'html css php'`. Найдите количество символов в этой строке.
62. Дана переменная `$password`, в которой хранится пароль пользователя. Если количество символов пароля больше 5-ти и меньше 10-ти, то выведите пользователю сообщение о том, что пароль подходит, иначе сообщение о том, что нужно придумать другой пароль.

Примерные практические задания к экзамену

1. Дан массив `$arr`. Подсчитайте количество элементов этого массива.
2. Дан массив `$arr`. С помощью функции `count` выведите последний элемент данного массива.
3. Дан массив с числами. Проверьте, что в нем есть элемент со значением 3.
4. Дан массив `[1, 2, 3, 4, 5]`. Найдите сумму элементов данного массива.
5. Дан массив `[1, 2, 3, 4, 5]`. Найдите произведение (умножение) элементов данного массива.
6. Дан массив `$arr`. С помощью функций `array_sum` и `count` найдите среднее арифметическое элементов (сумма элементов делить на их количество) данного массива.
7. Создайте массив, заполненный числами от 1 до 100.
8. Создайте массив, заполненный буквами от 'a' до 'z'.
9. Создайте строку `'1-2-3-4-5-6-7-8-9'` не используя цикл.
10. Найдите сумму чисел от 1 до 100 не используя цикл.
11. Найдите произведение чисел от 1 до 10 не используя цикл.
12. Даны два массива: первый с элементами 1, 2, 3, второй с элементами 'a', 'b', 'c'. Сделайте из них массив с элементами 1, 2, 3, 'a', 'b', 'c'.
13. Дан массив с элементами 1, 2, 3, 4, 5. С помощью функции `array_slice` создайте из него массив `$result` с элементами 2, 3, 4.

14. Дан массив [1, 2, 3, 4, 5]. С помощью функции `array_splice` преобразуйте массив в [1, 4, 5].
15. Дан массив [1, 2, 3, 4, 5]. С помощью функции `array_splice` запишите в новый массив элементы [2, 3, 4].
16. Дан массив [1, 2, 3, 4, 5]. С помощью функции `array_splice` сделайте из него массив [1, 2, 3, 'a', 'b', 'c', 4, 5].
17. Дан массив [1, 2, 3, 4, 5]. С помощью функции `array_splice` сделайте из него массив [1, 'a', 'b', 2, 3, 4, 'c', 5, 'e'].
18. Дан массив 'a'=>1, 'b'=>2, 'c'=>3'. Запишите в массив `$keys` ключи из этого массива, а в `$values` – значения.
19. Даны два массива: ['a', 'b', 'c'] и [1, 2, 3]. Создайте с их помощью массив 'a'=>1, 'b'=>2, 'c'=>3'.
20. Дан массив 'a'=>1, 'b'=>2, 'c'=>3. Поменяйте в нем местами ключи и значения.
21. Дан массив с элементами 1, 2, 3, 4, 5. Сделайте из него массив с элементами 5, 4, 3, 2, 1.
22. Дан массив ['a', '-', 'b', '-', 'c', '-', 'd']. Найдите позицию первого элемента '-'.
23. Дан массив ['a', '-', 'b', '-', 'c', '-', 'd']. Найдите позицию первого элемента '-' и удалите его с помощью функции `array_splice`.
24. Дан массив ['a', 'b', 'c', 'd', 'e']. Поменяйте элемент с ключом 0 на '!', а элемент с ключом 3 - на '!!'.
25. Дан массив '3'=>'a', '1'=>'c', '2'=>'e', '4'=>'b'. Попробуйте на нем различные типы сортировок.
26. Дан массив с элементами 'a'=>1, 'b'=>2, 'c'=>3. Выведите на экран случайный ключ из данного массива.
27. Дан массив с элементами 'a'=>1, 'b'=>2, 'c'=>3. Выведите на экран случайный элемент данного массива.
28. Дан массив `$arr`. Перемешайте его элементы в случайном порядке.
29. Заполните массив числами от 1 до 25 с помощью `range`, а затем перемешайте его элементы в случайном порядке.
30. Создайте массив, заполненный буквами от 'a' до 'z' так, чтобы буквы шли в случайном порядке и не повторялись.
31. Сделайте строку длиной 6 символов, состоящую из маленьких английских букв, расположенных в случайном порядке. Буквы не должны повторяться.
32. Дана строка '1234567890'. Найдите сумму цифр из этой строки не используя цикл.
33. Создайте массив ['a'=>1, 'b'=2... 'z'=>26] не используя цикл.
34. Создайте массив вида [[1, 2, 3], [4, 5, 6], [7, 8, 9]] не используя цикл.
35. Дан массив с элементами 1, 2, 4, 5, 5. Найдите второй по величине элемент. В нашем случае это будет 4.
36. Дан массив с числами. Найдите среднее арифметическое его элементов (сумма элементов делить на количество) не используя цикл.
37. Найдите сумму чисел от 1 до 100 не используя цикл.
38. Выведите столбец чисел от 1 до 100 не используя цикл.
39. Заполните массив 10-ю иксами не используя цикл.
40. Заполните массив 10-ю случайными числами от 1 до 10 так, чтобы они не повторялись. Цикл использовать нельзя.
41. Найдите факториал заданного числа не используя цикл. Факториал - это произведение чисел от 1 до заданного числа включительно.
42. Дано число. Найдите сумму цифр этого числа не используя цикл.
43. Дана строка. Сделайте заглавным последний символ этой строки не используя цикл.
44. Дан массив с числами. Получите из него массив с квадратными корнями этих чисел не используя цикл.

45. Заполните массив числами от 1 до 26 так, чтобы ключами этих чисел были буквы английского алфавита: ['a'=>1, 'b'=>2...]. Сделайте это не используя цикл.
46. Дана строка с числами '1234567890'. Найдите сумму пар чисел: 12+34+56+78+90. Решите задачу, не используя цикл.
47. Выведите на экран текущий год, месяц, день, час, минуту, секунду.
48. Выведите текущую дату-время в форматах '2025-12-31', '31.12.2025', '31.12.13', '12:59:59'.
49. С помощью функций mktime и date выведите 12 февраля 2025 года в формате '12.02.2025'.
50. Создайте массив дней недели \$week. Выведите на экран название текущего дня недели с помощью массива \$week и функции date. Узнайте какой день недели был 06.06.2006, в ваш день рождения.
51. Создайте массив месяцев \$month. Выведите на экран название текущего месяца с помощью массива \$month и функции date.
52. Найдите количество дней в текущем месяце. Скрипт должен работать независимо от месяца, в котором он запущен.
53. Сделайте поле ввода, в которое пользователь вводит год (4 цифры), а скрипт определяет високосный ли год.
54. Сделайте форму, которая спрашивает дату в формате '31.12.2025'. С помощью функций mktime и explode переведите эту дату в формат timestamp. Узнайте день недели (словом) за введенную дату.
55. Сделайте форму, которая спрашивает дату в формате '2025-12-31'. С помощью функций mktime и explode переведите эту дату в формат timestamp. Узнайте месяц (словом) за введенную дату.
56. Узнайте сколько дней осталось до Нового Года. Скрипт должен работать в любом году.
57. Сделайте форму с одним полем ввода, в которое пользователь вводит год. Найдите все пятницы 13-е в этом году. Результат выведите в виде массива дат.
58. Узнайте какой день недели был 100 дней назад.
59. Сделайте функцию, которая параметрами принимает 2 числа. Если эти числа равны - пусть функция вернет true, а если не равны - false.
60. Сделайте функцию, которая параметрами принимает 2 числа. Если их сумма больше 10 - пусть функция вернет true, а если нет - false.
61. Сделайте функцию, которая параметром принимает число и проверяет - отрицательное оно или нет. Если отрицательное - пусть функция вернет true, а если нет - false.
62. С помощью цикла for сформируйте строку '123456789' и запишите ее в переменную \$str.
63. С помощью цикла for сформируйте строку '987654321' и запишите ее в переменную \$str.
64. С помощью цикла for сформируйте строку '-1-2-3-4-5-6-7-8-9-' и запишите ее в переменную \$str. Спросите у пользователя имя с помощью формы. Сделайте чекбокс: если он отмечен, то поприветствуйте пользователя, если не отмечен - попрощайтесь с пользователем.
65. Спросите у пользователя, какие из языков он знает: html, css, php, javascript. Выведите на экран те языки, которые знает пользователь.
66. Спросите у пользователя знает ли он PHP с помощью двух radio-кнопок. Выведите результат на экран. Сделайте так, чтобы по умолчанию один из вариантов был уже отмечен.
67. Спросите у пользователя его возраст с помощью нескольких radio-кнопок. Варианты ответа сделайте такими: менее 20 лет, 20-25, 26-30, более 30.
68. Спросите у пользователя его возраст с помощью select. Варианты ответа сделайте такими: менее 20 лет, 20-25, 26-30, более 30.

69. Спросите у пользователя с помощью мультиселекта, какие из языков он знает: html, css, php, javascript. Выведите на экран те языки, которые знает пользователь.

**Приложение 3
к рабочей программе**

Федеральное государственное бюджетное образовательное учреждение высшего образования
УРАЛЬСКИЙ ГОСУДАРСТВЕННЫЙ ЭКОНОМИЧЕСКИЙ УНИВЕРСИТЕТ

УТВЕРЖДЕНЫ
на заседании кафедры информационных
технологий и статистики

**Перечень курсовых работ по дисциплине
Разработка многоуровневых приложений**

Перечень курсовых работ

1. Разработка интернет магазина для крупного промышленного предприятия
2. Разработка корпоративного мессенджера
3. Разработка приложения "интернет аукцион"
4. Разработка приложения автоматизации предприятий автосервиса
5. Разработка приложения автоматизации работы деканата ВУЗа
6. Разработка приложения для учебного тренажера обучения иностранным языкам
7. Разработка приложения мониторинга продукции, представленной в интернет-магазинах на российском рынке
8. Разработка приложения обеспечения оценки знаний студентов
9. Разработка приложения онлайнкинотеатр
10. Разработка приложения оценки кредитоспособности физических лиц
11. Разработка приложения резервирования билетов
12. Разработка приложения с использованием базы данных MySQL с доступом через Web-интерфейс.
13. Разработка приложения с использованием скриптового языка программирования PHP и базы данных MySQL
14. Разработка приложения учета взаимоотношений с клиентами компании (продажа компьютеров и комплектующих)
15. Разработка приложения учёта материально-технических средств на примере предприятия
16. Разработка программного приложения "Виртуальный магазин по продаже мебели"
17. Разработка форума с использованием PHP 5 и MySQL
18. Разработка Web-приложения "Магазин книг"
19. Разработка Web-приложения с использованием JavaScript каркаса Node.js
20. Создание web-приложения, представляющего собой интернет-магазин по продаже специализированного оборудования для ПК

В электронном портфолио обучающегося по дисциплине размещается

<http://portfolio.usue.ru>

- курсовая работа

Методические рекомендации по выполнению курсовой работы по дисциплине «Разработка многоуровневых приложений»

Общие положения

Осуществление будущей профессиональной деятельности бакалавра по направлениям «Информатика и вычислительная техника», «Прикладная информатика», «Математическое обеспечение и администрирование информационных систем» предполагает использование полученных в процессе обучения систематизированных теоретических знаний для решения практических задач. Важнейшим фактором формирования высокого профессионального уровня специалиста, его умений и навыков является самостоятельная работа, выполняемая в форме курсовых работ по изучаемым дисциплинам.

В процессе обучения студенты указанных выше направлений подготовки бакалавриата должны выполнить курсовую работу по дисциплине «Разработка многоуровневых приложений».

Задачи, решаемые в процессе выполнения курсовой работы:

- освоение методологии разработки многоуровневых приложений;
- уяснение принципов разработки программного обеспечения;
- ознакомление с методами и средствами разработки многоуровневых приложений находящимся за пределами учебного курса.

Курсовая работа должна быть сдана научному руководителю не позднее, чем за 2 недели до сессии (или по специальной договоренности с руководителем при форс-мажорных обстоятельствах) с целью ее своевременной проверки и защиты. Студент, не защитивший курсовую работу, не допускается к сдаче экзамена по данной дисциплине.

Выбор темы и консультирование

При определении темы курсовой работы необходимо учитывать возможность использования современных методов, алгоритмов и соответствующего программного обеспечения.

Тематика работ определяется кафедрой в соответствии с учебным планом, направлениями научно-исследовательской работы студента.

В течение семестра проводятся консультации с руководителем курсовой работы, во время которой обсуждаются сложности, с которыми сталкивается студент при выполнении задания. Руководитель оказывает помощь студенту в составлении графика работы, подборе необходимой литературы, проводит консультации, проверяет выполнение проекта и организует его защиту. Однако контроль руководителя не освобождает студента от ответственности за правильность и своевременность выполнения проекта. Основным объемом при выполнении курсового проекта составляет самостоятельная работа.

Защита курсовой работы проводится в следующем порядке:

а) не позднее, чем за две недели до защиты курсовой работы студентом руководитель рассматривает код написанной программы (или демонстрационные примеры использования выбранной технологии) и допускает до защиты;

б) студент оформляет курсовую работу и готовит доклад для выступления;

в) защита выполняется публично, путем доклада перед комиссией, в которой обязательно присутствует преподаватель дисциплины, а также другие преподаватели кафедры информационных технологий и статистики и практикующие специалисты в области разработки программного обеспечения. В результате защиты оценивается сложность работы, качество оформления материалов, выполненный доклад. На основании усреднения баллов членов комиссии выставляется оценка, даются рекомендации по публикации результатов, размещении результатов проекта на сайте кафедры.

Студенты могут по своему усмотрению выбрать темы, которые не включены в прилагаемый список, согласовав предлагаемые оригинальные варианты

с руководителем. Студенты выполняют курсовые работы на объектно-ориентированных языках программирования C#, Java. При реализации WEB интерфейса допускается использование PHP, SQL, JavaScript, а также использование CMS Joomla, WordPress и 1СБитрикс (при желании и договоренности с руководителем, работа может быть выполнена на других объектно-ориентированных языках и платформах).

Руководят работой, консультируют, помогают уточнить план, список литературы, контролируют ход выполнения работы преподаватели кафедры информационных технологий и статистики.

Организация выполнения курсовой работы

Структура и требования к содержанию курсовой работы

Сущность темы должна быть изложена четко и последовательно, каждый последующий раздел должен быть логическим продолжением предыдущего, вытекать из него и быть с ним взаимосвязанным. Общий объем работы должен составлять 25 – 35 страниц, включая список использованных источников (не включая приложения). Курсовая работа должна быть оформлена в соответствии с Положением о требованиях к оформлению рефератов, отчетов по практике, контрольных, курсовых и дипломных работ, утвержденным Решением Совета по учебно-методическим вопросам и качеству образования УрГЭУ от 17.03.2011 г.

Курсовая работа включает:

титульный лист;

содержание;

введение;

основную часть, соответствующую теме курсовой работы;

заключение;

список использованных источников;

приложения.

Титульный лист должен содержать все необходимые идентификационные признаки исполнителя (см. приложение А).

В содержании последовательно перечисляются заголовки разделов курсовой работы, указываются номера страниц, с которых они начинаются.

Во введении к курсовой работе необходимо:

- обосновать актуальность темы курсовой работы;
- сформулировать цель курсовой работы и ее задачи;
- Привести краткую аннотацию работы.

Объем введения – 1-2 страницы текста.

В основной части работы раскрывается содержание заявленной темы.

В заключении приводятся основные выводы по курсовой работе, подтверждающие выполнение поставленных задач и отражающие полученные в курсовой работе результаты.

Объем заключения – 1-2 страницы.

Список использованных источников должен содержать перечень использованных при написании курсовой работы литературных источников с их полным описанием по требованиям стандартов. В список в обязательном порядке включаются источники, материалы которых использовались в процессе рассмотрения разделов, указанных в содержании работы (ссылки на Интернет – источники обязательны). Список использованных источников должен включать не менее 20 наименований, в том числе не менее 2 наименований на иностранном языке. В тексте курсовой работы обязательно должны находиться ссылки на указанную в данном списке литературу (в квадратных скобках).

В приложения выносятся расчеты, систематизирующие первичные материалы, а также иллюстрационный материал, не носящий принципиального характера, но требуемый логикой изложения, отчетные формы документов, листинги программ с комментариями и другие источники информации.

Содержание

Слово «содержание» записывают в виде заголовка симметрично тексту прописными буквами. В содержании работы указывается перечень всех глав и параграфов курсовой работы, а также номера страниц, с которых начинается каждый из них. Главы в курсовой работе должны иметь в пределах всей работы порядковые номера, обозначенные арабскими цифрами. Параграфы каждой главы должны иметь нумерацию в пределах каждой главы. Номер параграфа состоит из номера главы и непосредственно номера параграфа в данной главе, отделенного от номера главы точкой.

Введение

Во введении должно присутствовать: актуальность темы, цель разработки системы, назначение, решаемые задачи, краткое описание содержания курсовой работы. Введение не должно превышать двух страниц.

Основная часть работы

Основная часть оформляется в виде следующих глав и параграфов.

Первая глава является аналитической и посвящена анализу предметной области. Первый параграф посвящен *описанию предметной области*: чему посвящена предметная область, какие в ней есть термины и понятия, субъекты и объекты, способы взаимодействия субъектов, способы использования объектов, закономерности. Отдельный параграф должен быть посвящен выделению *словаря предметной области*: что входит в словарь предметной области, отдельно выделив список существительных и список глаголов, которые могут быть связаны с существительными. Третий параграф посвящен *анализу существующих программных решений* и известных подходов к решению подобных задач. Четвертый параграф посвящен подготовке технического задания на разработку программного обеспечения. *Техническое задание* в обязательном порядке включает в себя функциональные требования (с диаграммой вариантов использования и описания вариантов использования), требования к экранным формам, модель данных и нефункциональные требования. Другие пункты технического задания могут включаться дополнительно.

Вторая глава курсовой работы является проектной и посвящена проектированию и разработке программного обеспечения. Первый параграф должен содержать описание процесса *проектирования и разработки классов*. Входной информацией являются определенные на этапе анализа задач объекты, общая характеристика программы. Проектирование системы классов начинается с обработки словаря предметной области. Эта обработка состоит в выявлении того, какие слова соответствуют объектам, классам, свойствам и операциям. Список существительных служит основой для выделения классов и их свойств, а список глаголов – для определения операций. Следующий шаг состоит в том, чтобы определить, какой из классов какие свойства и функции содержит. Следует обратить внимание на то, что эти наборы у разных классов могут «пересекаться». Ещё одним вопросом, требующим решения на данном шаге, является выявление отношений между классами. Речь идёт об отношениях наследования и включения. Следует обратить внимание, что понятие «наследование» чаще всего возникает тогда, когда разные классы обладают частично схожими наборами свойств и методов. При составлении словаря предметной области далеко не всегда в список могут попасть понятия, которым можно сопоставить базовые классы в иерархии наследования. Поэтому следует внимательно проанализировать список классов, свойств, методов, их соответствие, и, возможно, выделить ряд новых классов, связанных с имеющимися отношениями наследования и включения. На данном этапе можно уже учитывать не только законы предметной области, но и такие принципы объектно-ориентированного подхода как абстракция, инкапсуляция, полиморфизм. Завершается этап тем, что разрабатываются полные спецификации базовых классов системы. При этом характеристики объектов вписываются в поля классов, а на основании операций разрабатываются методы и определяются способы доступа (*public, private, protected*). Следует иметь в виду, что все имена классов, полей и методов должны быть осмысленными, начинаться с заглавной буквы. На этапе проектирования для именования классов, полей и методов рекомендуется использовать русские

слова, однако, на этапе разработки предпочтительным является английский язык. Большим плюсом к проекту будет являться использование известных паттернов (шаблонов) проектирования, таких как: одиночка (singleton), прототип (prototype), строитель (builder), абстрактная фабрика (abstract factory), адаптер (adapter), декоратор (decorator), заместитель (proxy), мост (bridge), компоновщик (composite), фасад (facade), итератор (iterator), команда (command), наблюдатель (observer), стратегия (strategy), посетитель (visitor), посредник (mediator), состояние (state). Завершается этот этап подготовкой и представлением в этом разделе UML- диаграмм классов.

Второй параграф посвящен *проектированию пользовательского интерфейса*. На этом этапе разрабатываются структурная схема пользовательского интерфейса программы, детали управления системой (пользовательские истории) и дизайн-макеты основных окон графического интерфейса. Структурная схема позволяет выверить все детали проекта, определить взаимоотношения между отдельными частями программы, а также определяет содержание программных сообщений. На основании этой схемы в дальнейшем можно построить схему движения информационных потоков, диаграмму взаимодействия классов и т.д.

Третий параграф посвящен *описанию разработанных алгоритмов и модулей*. В данном параграфе приводится иерархическая структура приложения и схема связности модулей, а также описание наиболее важных (сложных) алгоритмов, разработанных для данного программного комплекса. Алгоритмы можно описать с использованием классического языка блок-схем, языка программирования (с комментариями) или с использованием нотации UML-диаграмм (activity, sequence). В курсовой работе в обязательном порядке должны быть использованы классические алгоритмы обработки структур данных (сортировка, поиск, списки, деревья, графы и пр.) и собственные разработанные алгоритмы. Исходный код разработанных программ следует включать в приложения к курсовой работе. Четвертый параграф содержит описание процесса тестирования программного комплекса, включая описа-

ние тестов, их анализ, достоинства и недостатки разработки. Пятый параграф содержит краткое *руководство пользователя* по эксплуатации разработанной системы.

Названия глав и параграфов должны быть, по возможности, отличными от предложенных и отражать тематику курсовой работы. В некоторых работах могут присутствовать не все вышеперечисленные параграфы, тогда как в некоторых работах могут добавляться дополнительные параграфы, если того требует тематика курсовой работы.

Заключение

В заключении логически последовательно излагаются основные результаты, использованные методы, оценка полноты решений поставленных задач, значимость работы, область применения полученных результатов.

Список использованных источников

Список использованных источников является составной частью работы и отражает степень изученности рассматриваемой проблемы. При этом в список включаются, как правило, не только те источники, на которые в работе имеются библиографические ссылки, но и те, которые были изучены при исследовании темы работы: список использованной литературы, фундаментальные труды, книги, публикации в периодической печати, Интернет – источники. При этом, в основной части работы должны быть представлены ссылки на указанные в данном списке источники.

Приложения

В приложении в обязательном порядке должен быть представлен полный листинг разработанной программы с комментариями.

Также, в приложения следует относить вспомогательный материал, который при включении в основную часть работы загромождает текст. К вспомогательному материалу относятся промежуточные расчеты, таблицы вспомогательных цифровых данных, программный код, инструкции, методики, распечатки текстов программ для компьютеров, иллюстрации вспомогательного характера, заполненные формы отчетности и других документов.

Порядок представления на проверку и защита курсовой работы

Защита курсовой работы проводится в следующем порядке:

а) не позднее, чем за две недели до защиты курсовой работы студентом руководитель рассматривает код написанной программы (или демонстрационные примеры использования выбранной технологии) и допускает до защиты;

б) студент оформляет курсовую работу и готовит доклад для выступления;

в) защита выполняется публично, путем доклада перед комиссией, в которой обязательно присутствует преподаватель дисциплины и другие преподаватели. В результате защиты оценивается сложность работы, качество оформления материалов, выполненный доклад. На основании усреднения баллов членов комиссии выставляется оценка, даются рекомендации по публикации результатов, размещении результатов проекта на сайте кафедры.

После защиты курсовой работы студент получает окончательную оценку, которая проставляется в зачетной книжке и в ведомости.

Курсовая работа позволяет студенту обобщить и систематизировать полученные ранее знания, углубить их в области экономических дисциплин, развить способности к творческому мышлению в сфере управления предприятием, а также является фундаментом для подготовки выпускной квалификационной работы.

Типовая структура курсовой работы

1. Аналитическая часть

1.1. Описание предметной области

- 1.2.Словарь предметной области
- 1.3.Анализ существующих программных решений
- 1.4.Техническое задание
 - 1.4.1. Функциональные требования
 - 1.4.2. Требования к экранным формам
 - 1.4.3. Модель данных
 - 1.4.4. Нефункциональные требования
- 2. Проектная часть
 - 2.1.Проектирование и разработка классов
 - 2.2.Проектирование и разработка базы данных *(если имеется)*
 - 2.3.Проектирование пользовательского интерфейса
 - 2.4.Описание разработанных алгоритмов и программных модулей
 - 2.4.1. Иерархическая структура приложения
 - 2.4.2. Алгоритмы и математическое обеспечение
 - 2.4.3. Описание основных программных модулей
 - 2.5.Тестирование программного комплекса
 - 2.5.1. Описание методики тестирования
 - 2.5.2. Результаты тестов и их анализ
 - 2.6.Руководство пользователя

Критерии оценки курсовых работ

Критерий оценки	Оценка «отлично»	Оценка «хорошо»	Оценка «удовлетворительно»
1. Актуальность выбранной темы, обоснованность значимости проблемы для объекта исследования	В работе четко обоснована актуальность темы КР с точки зрения значимости проблемы исследования. Формулировки точные и аргументированные.	В работе недостаточно полно обоснована актуальность темы КР и значимость проблемы исследования. Допускаются отдельные недочеты в формулировках.	Актуальность темы КР обоснована не точно. Значимость проблемы исследования сформулирована фрагментарно.
2. Теоретическая и практическая значимость исследования;	В работе обоснована теоретическая значимость исследования, отражена его связь с задачами профессиональной деятель-	В работе не четко обоснована теоретическая значимость исследования, отражена связь исследования с задачами профессиональ-	В работе не обоснована теоретическая значимость исследования, связь исследования с задачами профессиональной дея-

Критерий оценки	Оценка «отлично»	Оценка «хорошо»	Оценка «удовлетворительно»
	ности. Работа содержит рекомендации по использованию результатов проведенного исследования на практике	ной деятельности. Работа содержит элементы рекомендаций по использованию результатов проведенного исследования на практике	тельности не четко определена.
3. Четкость формулировок цели и задач исследования, методическая грамотность	Цель и задачи исследования грамотно сформулированы, структура работы им полностью соответствует. Правильно выбраны и применены необходимые методы исследования.	Цель и задачи грамотно сформулированы, структура работы в основном им соответствуют. Имеются незначительные ошибки в выборе и/или применении методов исследования.	Цель и задачи сформулированы недостаточно четко, слабо связаны со структурой работы. Имеются грубые ошибки в выборе и/или применении методов исследования.
4. Логичность изложения материала. Наличие аргументированных выводов по результатам работы, их соответствие целевым установкам	В работе имеется четкая структура, внутреннее единство и композиционная целостность, логическая последовательность изложения материала. Сделаны аргументированные выводы по результатам работы, они соответствуют целевым установкам.	Материал изложен логично и последовательно, но имеются недочеты в структуре работы. Сделаны выводы по результатам работы, но они не всегда аргументированы. Выводы в основном соответствуют целевым установкам.	В работе отсутствует внутреннее единство, имеются нарушения в логике и последовательности изложения материала. Выводы поверхностные, не всегда соответствуют целевым установкам
5. Наличие обоснованных предложений по совершенствованию предметной области, изложение своего видения перспектив дальнейшего исследования проблемы	В работе содержатся самостоятельно сформулированные предложения по совершенствованию деятельности в организации, имеются обоснования, намечены пути дальнейшего исследования темы. (Возможно наличие акта внедрения)	В работе сформулированы предложения по совершенствованию деятельности организации, однако обоснования выполнены недостаточно корректно. Видение перспектив дальнейших исследований не структурированное.	Рекомендации по совершенствованию деятельности организации носят формальный характер. Видение перспектив дальнейших исследований отсутствует.
6. Уровень языковой и стилистической грамотности.	Высокий уровень языковой и стилистической грамотности. В работе отсутствуют речевые и орфографические ошибки. Автор свободно владеет деловым стилем речи	В работе допущены некоторые стилистические и речевые погрешности, при этом автор хорошо владеет деловым стилем речи	Недостаточное владение деловым стилем речи В работе имеются различного рода ошибки, опечатки исправлены не полностью.
7. Качество оформления работы	Работа оформлена в соответствии со всеми требованиями, предъявляемыми к КР.	Имеются незначительные недочеты в оформлении	Много недочетов в оформлении.
8. Качество презентационных материалов и устного выступления	Лаконичный и содержательный доклад отражающий основные положения и результаты исследования	Недостаточное освещение проблем исследования, некоторые сложности в формулировке главных	Доклад не дает представления о содержании и результатах исследования. Несоблюдение временного

Критерий оценки	Оценка «отлично»	Оценка «хорошо»	Оценка «удовлетворительно»
	<p>дования. Соблюдение установленного регламента. Ясные и четкие ответы на задаваемые вопросы и высказываемые замечания. Свободная ориентация в теме</p>	<p>выводов. Нарушение временного регламента незначительное. Ясные и четкие ответы на задаваемые вопросы и высказываемые замечания. Свободная ориентация в теме.</p>	<p>го регламента. Затруднения в ответах на вопросы, неточные формулировки.</p>

Основные требования по оформлению курсовой работы

Общие требования

Курсовая работа должна быть выполнена на одной стороне листа белой бумаги формата А4. Текст работы должен быть подготовлен компьютерным способом в текстовом редакторе MS Word (тип шрифта – Times new roman; размер шрифта – 14 пт; межстрочный интервал – 1,5; цвет шрифта – чёрный, равномерная плотность, контрастность и четкость изображения) и распечатан.

Разрешается использовать компьютерные возможности акцентирования внимания на определенных терминах и формулах, применяя курсив, полужирный шрифт не применяется. Не допускается цветное оформление заголовков рисунков, предложений и отдельных слов, а также вложение листов в отдельные файлы.

В печатном тексте допускается вписывание отдельных слов, формул, знаков только черными чернилами или черной тушью.

Текст работы следует печатать, соблюдая поля (правое – 10 мм, верхнее и нижнее – 20 мм, левое – 30 мм), абзацный отступ – 1,25 см. Рамки, ограничивающие текст, не вычерчиваются.

Описание работы следует излагать научным языком, просто, сжато и логически последовательно.

Следует избегать повторов, лишних слов, затрудняющих восприятие текста. В тексте всей работы необходимо соблюдать единообразие терминов, обозначений, символов.

Сокращение русских слов и словосочетаний допускается при условии соблюдения требований ГОСТ 7.12–93 «Система стандартов по информации, библиотечному и издательскому делу. Библиографическая запись. Сокращение слов на русском языке. Общие требования и правила».

В тексте курсовой работы допускаются общепринятые сокращения и аббревиатуры, установленные правилами орфографии и соответствующими нормативными документами, например: год – г., годы – гг., и так далее – и т.

д., метр – м, тысяч – тыс., миллион – млн, миллиард – млрд, триллион – трлн, страница – с., Российская Федерация – РФ, общество с ограниченной ответственностью – ООО.

При использовании авторской аббревиатуры необходимо при первом ее упоминании дать полную расшифровку, например: «... Уральский государственный экономический университет (далее – УрГЭУ)...».

Не допускается использование сокращений и аббревиатур в заголовках письменной работы, глав и параграфов.

В тексте следует избегать личных местоимений, заменяя их безличными формами (вместо «я считаю» следует писать «автор считает» или «считается»).

При необходимости в тексте курсовой работы могут быть приведены перечисления. Перед каждым элементом перечисления следует ставить дефис (иные маркеры не допустимы!).

Например:

«...закключение содержит:

- краткие выводы;
- оценку решений;
- разработку рекомендаций».

При необходимости ссылки в тексте работы на один из элементов перечисления вместо дефиса ставятся строчные буквы в порядке русского алфавита, начиная с буквы а (за исключением букв ё, з, й, о, ч, ъ, ы, ь). Для дальнейшей детализации перечислений необходимо использовать арабские цифры, после которых ставится скобка, а запись производится с абзацного отступа. Например:

- а) ...;
- б) ...;
- 1) ...;
- 2) ...;

в)

Страницы курсовой работы нумеруются арабскими цифрами внизу, по центру. Номер страницы не ставится на титульном листе и содержании, но они входят в общую нумерацию страниц так же, как и приложения. Иллюстрации по тексту работы, расположенные на отдельных листах, и страницы приложений нумеруются.

Оформление структурных элементов курсовой работы

Структурными элементами курсовой работы являются: содержание, введение, основная часть, заключение, список использованных источников.

Каждый структурный элемент курсовой работы необходимо начинать с новой страницы. Следующий параграф внутри одной главы начинается через 2 межстрочных интервала на том же листе, где закончился предыдущий.

Расстояние между заголовком структурного элемента и текстом, заголовками главы и параграфа, заголовком параграфа и текстом составляет 2 межстрочных интервала.

Наименования структурных элементов курсовой работы («СОДЕРЖАНИЕ», «ВВЕДЕНИЕ», «ЗАКЛЮЧЕНИЕ» и т.д.) служат заголовками структурных элементов. Данные наименования пишутся по центру страницы прописными (заглавными) буквами без точки в конце и без подчеркивания.

Главы и параграфы основной части должны иметь заголовки. Их следует нумеровать арабскими цифрами и записывать по центру страницы прописными (заглавными) буквами без точки в конце, не подчеркивая. Номер главы указывается цифрой (например, 1, 2, 3), номер параграфа включает номер главы и порядковый номер параграфа, разделенные точкой (например, 1.1, 2.1, 3.3). После номера главы и параграфа в тексте точку не ставят. Если заголовок состоит из двух предложений, их разделяют точкой. Переносы слов в заголовках не допускаются. Не допускается писать заголовок параграфа на одном листе, а его текст – на другом.

Использование и оформление рисунков

В курсовой работе для наглядности, уменьшения физического объема сплошного текста следует использовать иллюстрации – графики, схемы, диаграммы, чертежи, рисунки и фотографии. Все иллюстрации именуются рисунками. Их количество зависит от содержания работы и должно быть достаточно для того, чтобы придать ей ясность и конкретность.

Рисунки должны быть созданы с помощью инструментов Microsoft Word, возможно использование цвета. При цветном исполнении рисунков следует использовать принтер с возможностью цветной печати. При использовании в рисунках черно-белой печати следует применять черно-белую штриховку элементов рисунка.

На все рисунки должны быть даны ссылки в тексте работы, например: «... в соответствии с рисунком 2 ...» или «... тенденцию к снижению (рисунок 2)».

Рисунки следует располагать в работе непосредственно после текста, в котором они упоминаются впервые (при наличии достаточного пространства для помещения рисунка со всеми поясняющими данными), или на следующей странице. Если рисунок достаточно большой по размерам или если в нем имеется много детализированной информации, его можно размещать на отдельном листе. Допускается поворот рисунка по часовой стрелке (если он выполнен на отдельном листе). Рисунки, размеры которых больше формата А4, учитывают как одну страницу и размещают в приложениях.

Рисунки, за исключением рисунков в приложениях, следует нумеровать арабскими цифрами сквозной нумерацией по всей работе. Каждый рисунок (схема, график, диаграмма), обозначаемый словом «Рисунок», должен иметь заголовок и подписываться следующим образом – посередине строки без абзачного отступа, например:

Рисунок 1 – Структура администрации района

Если на рисунке отражены показатели, то после заголовка рисунка через запятую указывается единица измерения, например:

Рисунок 1 – Структура издержек, %

Рисунки каждого приложения обозначают отдельной нумерацией арабскими цифрами с добавлением перед цифрой обозначения приложения (например, рисунок А.3).

Если рисунок взят из первичного источника без авторской переработки, следует сделать ссылку, например:

Рисунок 2 – Система работы с кадрами [8, с. 15]

Если рисунок является авторской разработкой, необходимо после заголовка рисунка поставить знак сноски и указать в форме подстрочной сноски внизу страницы, на основании каких источников он составлен, например:

¹ Составлено автором по: [15, 23, 42].

При необходимости между рисунком и его заголовком помещаются поясняющие данные (подрисуночный текст), например, легенда.

Использование и оформление таблиц

В курсовой работе фактический материал в обобщенном и систематизированном виде может быть представлен в виде таблиц для наглядности и удобства сравнения показателей.

На все таблицы должны быть ссылки в работе. При ссылке следует писать слово «таблица» с указанием ее номера, например: «...в таблице 2 представлены ...» или «... характеризуется показателями (таблица 2)».

Таблицу следует располагать в работе непосредственно после текста, в котором она упоминается впервые, или на следующей странице.

Таблицы, за исключением таблиц в приложениях, следует нумеровать арабскими цифрами сквозной нумерацией по всей работе. Каждая таблица должна иметь заголовок, который должен отражать ее содержание, быть точным, кратким. Заголовок таблицы следует помещать над таблицей слева, без абзацного отступа в одну строку с ее номером через тире, например:

Таблица 1 – Динамика показателей за 2010–2011 гг.

Если таблица взята из первичного источника без авторской переработки, следует сделать ссылку, например:

Таблица 1 – Источники набора персонала [15, с. 35]

Если таблица является авторской разработкой, необходимо после заголовка таблицы поставить знак сноски и указать в форме подстрочной сноски внизу страницы, на основании каких источников она составлена, например:

¹ Составлено автором по: [1, 3, 10].

Располагают таблицы на странице обычно вертикально. Помещенные на отдельной странице таблицы могут быть расположены горизонтально, причем графа с наименованиями показателей должна размещаться в левой части страницы. Таблицы ограничивают линиями.

Таблицу с большим числом строк допускается переносить на другую страницу. При переносе части таблицы на другую страницу слово «Таблица» указывают один раз слева над первой частью таблицы. На странице, на которую перенесена часть таблицы, слева пишут «Продолжение таблицы» или «Окончание таблицы» с указанием номера таблицы и повторением шапки таблицы.

Если таблица переносится, то на странице, где помещена первая часть таблицы, нижняя ограничительная линия таблицы не проводится. Это же относится к странице (страницам), где помещено продолжение (продолжения) таблицы. Нижняя ограничительная линия таблицы проводится только на странице, где помещено окончание таблицы.

Заголовки граф и строк таблицы следует писать с прописной буквы в единственном числе, а подзаголовки граф – со строчной буквы, если они составляют одно предложение с заголовком, или с прописной буквы, если они имеют самостоятельное значение. В конце заголовков и подзаголовков таблиц точки не ставят. Заголовки граф, как правило, записывают параллельно строкам таблицы. При необходимости допускается перпендикулярное расположение заголовков граф.

Примечания к таблице (подтабличные примечания) размещают непосредственно под таблицей в виде: а) общего примечания; б) сноски; в) отдельной графы или табличной строки с заголовком. Выделять примечание в отдельную графу или строку целесообразно лишь тогда, когда примечание относится к большинству строк или граф. Примечания к отдельным заголовкам граф или строк следует связывать с ними знаком сноски. Общее примечание ко всей таблице не связывают с ней знаком сноски, а помещают после заголовка «Примечание» или «Примечания», оформляют как внутритекстовое примечание.

Допускается применять размер шрифта в таблице меньший, чем в тексте работы, но не менее 10 пт.

Если все показатели, приведенные в графах таблицы, выражены в одной и той же единице измерения, то ее обозначение необходимо помещать над таблицей справа. Если показатели таблицы выражены в разных единицах измерения, то обозначение единицы измерения указывается после наименования показателя через запятую. Допускается при необходимости выносить в отдельную графу обозначения единиц измерения.

Текст, повторяющийся в строках одной и той же графы и состоящий из одиночных слов, чередующихся с цифрами, заменяют кавычками. Если повторяющийся текст состоит из двух или более слов, то при первом повторении его заменяют словами «То же», а далее – кавычками. Если предыдущая фраза является частью последующей, то допускается заменить ее словами «То же» и добавить дополнительные сведения. При наличии горизонтальных линий текст необходимо повторять. Если в ячейке таблицы приведен текст из нескольких предложений, то в последнем предложении точка не ставится.

Заменять кавычками повторяющиеся в таблице цифры, математические знаки, знаки процента и номера, обозначения нормативных материалов, марок материалов не допускается.

При отсутствии отдельных данных в таблице следует ставить прочерк (тире). Цифры в графах таблиц должны проставляться так, чтобы разряды чисел во всей графе были расположены один под другим, если они относятся к одному показателю. В одной графе должно быть соблюдено, как правило, одинаковое количество десятичных знаков для всех значений величин.

Если таблицы размещены в приложении, их нумерация имеет определенные особенности. Таблицы каждого приложения нумеруют отдельной нумерацией арабскими цифрами. При этом перед цифрой, обозначающей номер таблицы в приложении, ставится буква соответствующего приложения, например:

Таблица В.1.– Динамика показателей за 2010–2011 гг.

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

Использование и оформление формул

Для составления математических формул используется редактор формул Microsoft Word.

Формулы и уравнения следует выделять из текста в отдельную строку. Выше и ниже каждой формулы и уравнения необходимо оставлять не менее одной свободной строки.

Формулы следует нумеровать порядковой нумерацией в пределах всей работы арабскими цифрами в круглых скобках в крайнем правом положении на строке, например:

$$R = X_{\max} - X_{\min}, \quad (4)$$

Формулы, размещаемые в приложениях, должны нумероваться 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. Например: «.. в формуле (1)...».

В качестве символов физических величин в формуле следует применять обозначения, установленные соответствующими нормативными документами. Пояснение символов и числовых коэффициентов, если они не пояснены ранее, должны быть приведены непосредственно под формулой, после которой ставится запятая.

Пояснение каждого символа следует давать с новой строки в той последовательности, в которой символы приведены в формуле. Первая строка пояснения должна начинаться без абзацного отступа со слова «где» (без двоеточия). Например:

$$R = X_{\max} - X_{\min}, \quad (4)$$

где X_{\max} – максимальное значение контролируемого параметра в выборке;

X_{\min} – минимальное значение контролируемого параметра в выборке.

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

Переносить формулы на следующую строку допускается только на знаках выполняемых операций, причем знак в начале следующей строки повторяют. При переносе формулы на знаке умножения применяют знак «х».

Порядок оформлений математических уравнений идентичен порядку оформления формул.

Оформление программного кода

Программный код на любом языке программирования оформляется следующим образом: шрифт – Courier New, 12 пт, межстрочный интервал – 1,15. Остальные требования аналогичны обычному тексту.

Перед написанием кода пишется слово «Листинг» и через пробел указывается его номер в курсовой работе (сквозная нумерация). Комментарии в коде обязательны.

Пример оформления части программного кода:

```
Листинг 1. Программа Hello World
public class Hello { // объявление класса Hello
    public static void main(String args[]) { // метод main
        System.out.println("Hello, world!"); // вывод на
                                                // консоль текста
    }
}
```

Оформление примечаний и ссылок

При необходимости пояснить содержание текста, таблицы или иллюстрации в курсовой работе следует помещать примечания. Их размещают непосредственно в конце страницы, таблицы, иллюстрации, к которым они относятся, и печатают с прописной буквы с абзацного отступа после слова «Примечание» или «Примечания». Если примечание одно, то после слова «Примечание» ставится тире и примечание печатается с прописной буквы. Одно примечание не нумеруют. Если их несколько, то после слова «Примечания» ставят двоеточие и каждое примечание печатают с прописной буквы с

новой строки с абзацного отступа, нумеруя их по порядку арабскими цифрами.

Цитаты, а также все заимствованные из печати данные (нормативы, цифры и др.), должны иметь библиографическую ссылку на первичный источник. Ссылка ставится непосредственно после того слова, числа, символа, предложения, по которому дается пояснение, в квадратных скобках. В квадратных скобках указывается порядковый номер источника в соответствии со списком использованных источников и номер страницы, с которой взята информация, например: [3, с. 15].

Приводимые в работе цитаты должны быть по возможности краткими. Если цитата полностью воспроизводит предложение цитируемого текста, она начинается с прописной буквы. Если цитата включена на правах части в предложение авторского текста, она пишется со строчной буквы. Если в цитату вошла только часть предложения цитируемого источника, то либо после кавычки ставится многоточие и цитата начинается с маленькой буквы, либо цитата начинается с большой буквы и заканчивается многоточием, например: Ф.Котлер подчеркивал, что современный маркетинг «...все в большей степени ориентируется на удовлетворение потребностей индивидуального потребителя» [26, с. 84].

Правила оформления списка использованных источников

Список должен содержать сведения об источниках, использованных при написании курсовой работы. Сведения об источниках приводятся в следующем порядке:

- ~ официальные материалы;
- ~ книги, статьи, материалы конференций и семинаров;
- ~ статистические сборники, инструктивные материалы, методические рекомендации, реферативная информация, нормативно-справочные материалы;
- ~ иностранная литература;

~ интернет-сайты.

В списке использованных источников применяется сквозная нумерация с применением арабского алфавита. Все объекты печатаются единым списком, группы объектов не выделяются.

Объекты описания списка должны быть обозначены терминами в квадратных скобках¹:

~ [Видеозапись];

~ [Мультимедиа];

~ [Текст];

~ [Электронный ресурс].

При занесении источников в список литературы следует придерживаться установленных правил их библиографического описания. Например:

Официальные материалы. В начале списка дается перечень использованных нормативных правовых актов федерального уровня в следующем порядке: международные нормативно-правовые акты, Конституция, кодексы, федеральные законы, указы Президента РФ, постановления Правительства РФ, нормативно-правовые акты иных федеральных органов государственной власти. Нормативные правовые акты одного уровня располагаются в хронологическом порядке, от принятых в более ранние периоды к принятым в более поздние периоды.

После федеральных нормативно-правовых актов перечисляются нормативно-правовые акты регионального, а затем муниципального уровней в том же порядке.

Примеры оформления нормативно-правовых актов:

1. Об общих принципах организации законодательных (представительных) и исполнительных органов власти субъектов Российской Федерации [Текст]: Федеральный закон РФ от 06.10.1999. N 184-ФЗ //Собрание законодательства РФ. –1999.– №43.

¹ Полный перечень см. в: Система стандартов по информации, библиотечному и издательскому делу. Библиографическая запись. Библиографическое описание. Общие требования и правила составления [Текст]: ГОСТ 7.1-2003.

2. О порядке разработки и утверждения административных регламентов исполнения государственных функций (предоставления государственных услуг) [Электронный ресурс]: Постановление Правительства РФ от 11.11.2005 г. N 679. – Доступ из справочно-правовой системы «Консультант-Плюс». – Режим доступа: <http://www.consultant.ru>.

Книги, статьи, материалы конференций и семинаров. Располагаются по алфавиту фамилии автора или названию, если книга печатается под редакцией. Например:

3. Боголюбов, А.Н. О вещественных резонансах в волноводе с неоднородным заполнением [Текст] / А.Н. Боголюбов, А.Л. Делицын, М.Д. Малых // Вестн. Моск. ун-та. Сер. 3, Физика. Астрономия. – 2001. – N 5. – С. 23–25.

4. Голубков, Е.П. Маркетинг как концепция рыночного управления [Текст] // Маркетинг в России и за рубежом. – 2001. – N 1. – С. 89–104.

5. Государственные и муниципальные финансы [Текст] : учебник / Под ред. проф. С.И. Лушина, проф. В.А. Слепова. – М.: Экономист, 2006. – 280 с.

6. Двинянинова, Г.С. Комплимент: Коммуникативный статус или стратегия в дискурсе [Текст] / Г.С. Двинянинова // Социальная власть языка: сб. науч. тр. / Воронеж. межрегион. ин-т обществ. наук, Воронеж. гос. ун-т, Фак. романо-герман. истории. – Воронеж, 2001. – С. 101–106.

7. История России [Текст] : учеб. пособие для студентов всех специальностей / В.Н. Быков [и др.] ; отв. ред. В.Н. Сухов ; М-во образования Рос. Федерации, С.-Петерб. гос. лесотехн. акад. – 2-е изд., перераб. и доп. / при участии Т.А. Суховой. – СПб.: СПбЛТА, 2001. – 231 с.

8. Семенов, В.В. Философия: итог тысячелетий. Философская психология [Текст] / В.В. Семенов; Рос. акад. наук, Пушкин. науч. центр, Ин-т биофизики клетки, Акад. проблем сохранения жизни. – Пушкино: ПНЦ РАН, 2000. – 64 с.

9. Черткова, Е.Л. Утопия как способ постижения социальной действительности [Электронный ресурс] / Е.Л. Черткова // Социемы: журнал Ураль-

ского гос. ун-та. – 2002. – N 8. – Режим доступа: <http://www2/usu.ru/philosoph/chertkova>.

10. Юридический советник [Электронный ресурс]. – 1 электрон. опт. диск (CD-ROM) : зв., цв. ; 12 см. – Прил.: Справочник пользователя [Текст] / сост. В.А. Быков. – 32 с.

Статистические сборники, инструктивные материалы, методические рекомендации, реферативная информация, нормативно-справочные материалы. Располагаются по алфавиту. Например:

11. Аппаратура радиоэлектронная бытовая. Входные и выходные параметры и типы соединений. Технические требования [Текст]: ГОСТ Р 517721-2001. – Введ. 2002-01-01. – М.: Изд-во стандартов, 2001. – 34 с.

12. Временные методические рекомендации по вопросам реструктуризации бюджетной сферы и повышения эффективности расходов региональных и местных бюджетов (Краткая концепция реструктуризации государственного и муниципального сектора и повышения эффективности бюджетных расходов на региональном и местном уровнях) [Текст]. – М.: ИЭПП, 2006. – 67 с.

13. Свердловская область в 1992-1996 годах [Текст]: Стат. сб. / Свердл. обл. комитет гос. статистики Госкомстата РФ. – Екатеринбург, 1997. – 115 с.

14. Социальное положение и уровень жизни населения России в 2010 г. [Текст]: Стат. сб. / Росстат. – М., 2002. – 320 с.

15. Социально-экономическое положение федеральных округов в 2010 г. [Электронный ресурс]. – Режим доступа: <http://www.gks.ru>.

Иностранная литература. Располагается по алфавиту. Например:

16. An Interview with Douglass C. North [Text] // The Newsletter of The Cliometric Society. – 1993. – Vol. 8. – N 3. – P. 23–28.

17. Burkhead, J. The Budget and Democratic Government [Text] / Lyden F.J., Miller E.G. (Eds.) / Planning, Programming, Budgeting. Markham : Chicago, 1972. 218 p.

18. Miller, D. Strategy Making and Structure: Analysis and Implications for Performance [Text] // Academy of Management Journal. – 1987. – Vol. 30. – N 1. – P. 45–51.

Интернет-сайты. Например:

19. Министерство финансов Российской Федерации: [Электронный ресурс]. – Режим доступа: <http://www.minfin.ru>

20. Российская книжная палата: [Электронный ресурс]. – Режим доступа: <http://www.bookchamber.ru>

Правила оформления приложений

В приложения рекомендовано включать материалы, которые по каким-либо причинам не могут быть включены в основную часть: материалы, дополняющие работу; промежуточные математические доказательства, формулы и расчеты; таблицы вспомогательных цифровых данных; инструкции, методики, описания алгоритмов и программ задач, иллюстрации вспомогательного характера; нормативные акты, например, должностные инструкции. В приложения также включают иллюстрации, таблицы и распечатки, выполненные на листах формата А3.

Приложения оформляют как продолжение данного документа на последующих его листах после списка использованных источников.

Приложения обозначают заглавными буквами русского алфавита, начиная с А, за исключением букв Ё, З, Й, О, Ч, Ь, Ы, Ъ (ПРИЛОЖЕНИЕ А, ПРИЛОЖЕНИЕ Б, ПРИЛОЖЕНИЕ В и т.д.). Допускается обозначение приложений буквами латинского алфавита, за исключением букв I и O. В случае полного использования букв русского и латинского алфавитов допускается обозначать приложения арабскими цифрами.

Само слово «ПРИЛОЖЕНИЕ» пишется прописными (заглавными) буквами.

Если в работе одно приложение, оно обозначается «ПРИЛОЖЕНИЕ А».

Каждое приложение следует начинать с новой страницы. При этом слово «ПРИЛОЖЕНИЕ» и его буквенное обозначение пишутся с абзацного отступа.

Приложение должно иметь заголовок, который записывают на следующей строке после слова «ПРИЛОЖЕНИЕ» с абзацного отступа. Заголовок пишется с прописной буквы.

В тексте работы на все приложения должны быть даны ссылки, например: «... в приложении Б...». Приложения располагают в порядке ссылок на них в тексте работы.

Текст каждого приложения, при необходимости, может быть разделен на разделы, подразделы, пункты, подпункты, которые нумеруют в пределах каждого приложения. Перед номером ставится обозначение этого приложения.

Приложения должны иметь общую с остальной частью документа сквозную нумерацию страниц.

Оформление содержания

В содержании курсовой работы наименования структурных элементов указываются с левого края страницы, при этом первая буква наименования является прописной (заглавной), остальные буквы являются строчными, например:

Введение	3
1. Постановка задачи	5
1.1 Цели и задачи разрабатываемой системы	5
1.2 Входная, нормативно-справочная и выходная информация	7
1.3 Перечень используемых функций	9

2. Теоретические аспекты создания клиент-серверных приложений	10
2.1 Клиент-серверные приложения: основные положения	10
.....	
3. Алгоритмы разработки игровых приложений	15
.....	
Заключение	30
Список использованных источников	32
Приложение А	33

Не ставятся точки между названием элемента содержания и номером страницы.

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ
ФЕДЕРАЦИИ

ФГБОУ ВО Уральский государственный экономический университет

КУРСОВАЯ РАБОТА

Дисциплина: Разработка многоуровневых приложений

Тема:

Институт цифровых технологий
управления и информационной
безопасности

Направление:

Профиль:

Кафедра информационных техно-
логий и статистики

Дата защиты

Оценка

Исполнитель:

Группа

Руководитель:

Екатеринбург

2024