

Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Силин Яков Петрович
Должность: Ректор
Дата подписания: 09.09.2021 14:45:14
Уникальный программный ключ:
24f866be26a1668403a8cbb3c509a9531e605f

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

Объявление
на заседании кафедры

10.01.2020 г.

протокол № 6

Зав. кафедрой Сурнина Н.М.

Утверждена
Советом по учебно-методическим вопросам
и качеству образования

15 января 2020 г.

протокол № 5

Председатель

Карх Д.А.

(подпись)

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

Наименование дисциплины	Проектирование архитектуры программных систем
Направление подготовки	09.03.01 Информатика и вычислительная техника
Профиль	Автоматизированные системы управления производством
Форма обучения	очная
Год набора	2020

Разработана:
Доцент, к.э.н.
Кислицын Е.В.

Екатеринбург
2020 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	4
7. СОДЕРЖАНИЕ ДИСЦИПЛИНЫ	6
8. ОСОБЕННОСТИ ОРГАНИЗАЦИИ ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА ПО ДИСЦИПЛИНЕ ДЛЯ ЛИЦ С ОГРАНИЧЕННЫМИ ВОЗМОЖНОСТЯМИ ЗДОРОВЬЯ	11
9. ПЕРЕЧЕНЬ ОСНОВНОЙ И ДОПОЛНИТЕЛЬНОЙ УЧЕБНОЙ ЛИТЕРАТУРЫ, НЕОБХОДИМОЙ ДЛЯ ОСВОЕНИЯ ДИСЦИПЛИНЫ	11
10. ПЕРЕЧЕНЬ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, ВКЛЮЧАЯ ПЕРЕЧЕНЬ ЛИЦЕНЗИОННОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ И ИНФОРМАЦИОННЫХ СПРАВОЧНЫХ СИСТЕМ, ОНЛАЙН КУРСОВ, ИСПОЛЬЗУЕМЫХ ПРИ ОСУЩЕСТВЛЕНИИ ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА ПО ДИСЦИПЛИНЕ	12
11. ОПИСАНИЕ МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЙ БАЗЫ, НЕОБХОДИМОЙ ДЛЯ ОСУЩЕСТВЛЕНИЯ ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА ПО ДИСЦИПЛИНЕ	13

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 по направлению подготовки 09.03.01 Информатика и вычислительная техника (уровень бакалавриата) (приказ Минобрнауки России
ПС	

1. ЦЕЛЬ ОСВОЕНИЯ ДИСЦИПЛИНЫ

изучение современных инженерных принципов (методов) создания надежного, качественного программного обеспечения, удовлетворяющего предъявляемым к нему требованиям; формирование у студентов умений и навыков применения методов программной

2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОПОП

Дисциплина относится к базовой части учебного плана.

3. ОБЪЕМ ДИСЦИПЛИНЫ

Промежуточный контроль	Часов					З.е.
	Всего за семестр	Контактная работа (по уч.зан.)			Самостоятельная работа в том числе подготовка контрольных и курсовых	
		Всего	Лекции	Лабораторные		
Семестр 5						
Экзамен	144	42	14	28	66	4

4. ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ ОСВОЕНИЯ ОПОП

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

Профессиональные компетенции (ПК)

Шифр и наименование компетенции	Индикаторы достижения компетенций
организационно-управленческий	
ПК-5 Способен выполнять работы и управлять работами по созданию (модификации) и сопровождению автоматизированных систем управления	ИД-1.ПК-5 Знать: принципы автоматизации задач организационного управления и бизнес-процессов; основы математического и имитационного моделирования бизнес-процессов, способы проектирования архитектуры программных систем, основы производственной деятельности, концепции ERP. Уметь: создавать автоматизированные системы управления производством; использовать средства анализа и моделирования при изучении бизнес-процессов организации; проектировать бизнес-процессы; модифицировать ERP-системы. Иметь навыки: разработки автоматизированных систем управления, моделей производственных и информационных систем.
проектный	

ПК-1 разрабатывать требования и проектировать программное обеспечение	Способен	ИД-1.ПК-1 Знать: методологии разработки программного обеспечения и технологии программирования, методологии и технологии проектирования и использования баз данных, методы и средства проектирования программных интерфейсов, принципы построения архитектуры программного обеспечения. Уметь: формировать требования к программному обеспечению; использовать существующие типовые решения и шаблоны проектирования ПО. Иметь навыки: разработки и изменения архитектуры программного обеспечения, проектирования структур и баз данных, программных интерфейсов.
--	----------	--

5. ТЕМАТИЧЕСКИЙ ПЛАН

Тема	Наименование темы	Всего часов	Контактная работа (по уч.зан.)			Самост. работа	Контроль самостоятельной работы
			Лекции	Лабораторные	Практические занятия		
			Часов				
Семестр 5		108					
Тема 1.	Архитектура программных систем. Унифицированный процесс	11	1	2		8	
Тема 2.	Требования к программной системе.	11	1	2		8	
Тема 3.	Классы и объекты. Модель предметной области.	13	2	4		7	
Тема 4.	Анализ и проектирование. Рабочий поток проектирования.	13	2	4		7	
Тема 5.	Документирование проекта архитектуры программной системы.	13	2	4		7	
Тема 6.	Прямое и обратное проектирование.	13	2	4		7	
Тема 7.	Архитектурные стили и шаблоны.	11	1	2		8	
Тема 8.	Шаблоны проектирования GRASP. Шаблоны проектирования GoF.	13	2	4		7	
Тема 9.	Архитектурные стили сетевых и распределенных программных систем.	10	1	2		7	

6. ФОРМЫ ТЕКУЩЕГО КОНТРОЛЯ И ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ШКАЛЫ ОЦЕНИВАНИЯ

Раздел/Тема	Вид оценочного средства	Описание оценочного средства	Критерии оценивания
Текущий контроль (Приложение 4)			
Темы 1-2	Тест (приложение 4)	Тест содержит 29 вопросов	10 баллов
Темы 3-4	Практическая работа (приложение 4)	Содержит практические задания	10 баллов
Темы 5-6	Практическая работа (приложение 4)	Содержит практические задания	10 баллов
Промежуточный контроль (Приложение 5)			
5 семестр (Эк)	Экзаменационный билет (приложение 5)	Билет содержит 2 теоретических вопроса и 1 практическое задание	100 баллов

ОПИСАНИЕ ШКАЛ ОЦЕНИВАНИЯ

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

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

Текущая аттестация. Используется 100-балльная система оценивания. Оценка работы студента в течении 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 текущей аттестации, 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 окончанию 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 окончанию 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 (ответа и 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 решения
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

Тема 1. Архитектура программных систем. Унифицированный процесс разработки.
Вводная информация. Понятие архитектуры программной информационной системы (ИС). Роль архитектуры в контексте процесса разработки программного обеспечения. Проект и архитектура. Цель проектирования архитектуры ИС. Парадигмы проектирования: структурная, объектно-ориентированная, функциональная, логическая. Объектно-ориентированный анализ и проектирование (ООАиП). Унифицированный процесс (УП) разработки программного обеспечения (ПО). Методология IBM Rational Unified Process (IBM RUP) как пример унифицированного процесса.

Тема 2. Требования к программной системе.

Прямое и обратное проектирование ПО и структуры базы данных (БД). Стереотипы UML и их использование. Заинтересованные лица, свойства ИС, классификация свойств. Формулировка проблемы. Диаграмма причинно-следственных связей. Матрица трассировки требований. Выявления трудоемкости реализации прецедента и свойств ИС. Понятие риска. Понятие объема работ проекта (Scope). Инструментальная поддержка процесса проектирования. Известные подходы к идентификации прецедентов. Первичные и вторичные экторы. Эскиз и спецификация прецедента. Выявление прецедентов в бизнес-процессах. Типовая структура спецификации. Основной и альтернативные потоки действий. Шаблоны выявления прецедентов ИС. Учет бизнес-правил при составлении спецификации прецедента. Пред-условия и пост-условия

Тема 3. Классы и объекты. Модель предметной области.

Понятия класса и объекта. Отношения между классами и их визуальное представление с помощью UML. Ассоциация и ее разновидности. Понятия навигации ассоциации, наследования, зависимости. Отличие агрегации от композиции. Понятие кратности ассоциации. Шаблоны действий при создании и преобразовании UML диаграмм классов. Ассоциативные классы и их использование в проекте. Процесс формирования визуальной модели программного обеспечения. Регистрируемые события и объекты. Связь модели предметной области с задачей идентификации функциональных требований к ИС.

Тема 4. Анализ и проектирование. Рабочий поток проектирования.

Обзор дисциплины «Анализ и проектирование». Входные и выходные артефакты (рабочие материалы) дисциплины. Роли и задачи дисциплины. Назначение и задачи этапов дисциплины. Используемые диаграммы языка UML в задачах проектирования. Роль архитектора в проекте и выполняемые им задачи. Ключевые абстракции и их идентификация. Диаграммы UML для визуализации ключевых абстракций. Архитектурные механизмы, их назначение. Архитектурные шаблоны и их использование. Понятие слоя и его представление в языке UML. Структура «реализация прецедента» (use case realization) и ее представление в визуальной модели.

Тема 5. Документирование проекта архитектуры программной системы.

Документирование проектируемой архитектуры ИС. Способы описания и документирования архитектуры ИС. Архитектурный документ как артефакт проекта. Представление архитектуры N+1 (4+1). Документ программной архитектуры (Software Architecture Document, SAD). Архитектурные представления. Архитектурные факторы. Архитектурные решения. Применение и оформление диаграмм UML в документации проекта. Государственный стандарт ГОСТ Р 57100-2016. Использование этого стандарта.

Тема 6. Прямое и обратное проектирование.

Лучшие практики разработки ИС. Назначение лучших практик. Связь практик с визуальным моделированием. Инструментальная поддержка практик. Понятие компонента архитектуры ИС. Интерфейс компонента. Версионный контроль основных артефактов проекта. Понятие конфигурации. CASE-средства визуального моделирования. Прямое и обратное проектирование.

Тема 7. Архитектурные стили и шаблоны.

Понятия архитектурного стиля, шаблона, фреймворка. Отличия архитектурного шаблона от шаблона проектирования. Популярные стили и шаблоны архитектуры ИС. Шаблон «Слои». Трёхзвенная (многозвенная) архитектура. Шаблоны «модель-представление-контроллер» (MVC), «сенсор- вычислитель-контроллер» (SCC). Стили «программа и подпрограммы», «виртуальная машина», «клиент-сервер», «каналы-и-фильтры», «классная доска», «издатель-подписчики», «событийная архитектура», «пиринговая архитектура», «конечный автомат». Плюсы и минусы различных архитектурных стилей. Соответствие стиля задаче.

Тема 8. Шаблоны проектирования GRASP. Шаблоны проектирования GoF. Шаблоны проектирования GRASP. Шаблоны: Creator, Information Expert, Low Coupling, Controller, High Cohesion, Polymorphism, Pure Fabrication, Indirection, Protected Variations. Шаблоны проектирования GoF. Шаблоны: Adapter, Factory, Singleton, Strategy, Composite, Facade, Observer и другие. Связь между шаблонами проектирования и архитектурными стилями. Применение шаблонов проектирования. Принципы проектирования: SRP, OCP, LSP, ISP, DIP. Адаптация и поддержка архитектуры ИС. Тестирование и рефакторинг при проектировании архитектуры. Итеративность и инкрементальность при проектировании архитектуры ИС. Элементы гибких процессов проектирования.

Тема 9. Архитектурные стили сетевых и распределенных программных систем. Проектирование архитектуры сетевых и распределенных программных систем. Архитектурный стиль «пиринговая архитектура» (peer-to-peer). Архитектурный стиль Representational state transfer (REST). Связь архитектурного стиля REST и устройства всемирной сети Интернет. Архитектурный шаблон «Удаленный вызов процедур» (RPC). Веб-сервисы. Стиль «Сервис-ориентированная архитектура приложений» (SOA). Проектирование архитектуры мобильных и встроенный приложений.

7.2 Содержание практических занятий и лабораторных работ

Тема 1. Архитектура программных систем. Унифицированный процесс разработки.

Основные принципы визуального моделирования. Сложность ПО и архитектурные представления. Статические и динамические диаграммы языка UML. Репозиторий модели CASE-средства.

Тема 2. Требования к программной системе.

Визуальное представление на UML модели прецедентов (сценариев использования). Словарь проекта. Концепция ИС. Модель прецедентов.

Тема 3. Классы и объекты. Модель предметной области.

Используемые UML-диаграммы, шаблоны при их построении. Понятие состояния объекта и его визуальное представление. Поиск экторов в модели предметной области.

Тема 4. Анализ и проектирование. Рабочий поток проектирования.

Выявление классов-участников. Применение прецедентов для проектирования объектной динамики. Стереотипы «граничный» («boundary»), «управляющий» («control»), «сущность» («entity») классов- участников реализации. Шаблоны для идентификации классов. Диаграммы последовательности и коммуникации UML. Правила использования стереотипов «boundary», «control», «entity» при проектировании объектной динамики. Диаграмма классов-участников (VOPC). Правила идентификации операций (ответственностей) классов. Правила идентификации отношений между классами – ассоциаций и зависимостей.

Понятие заинтересованных лиц. Детальность проекта. Архитектурные шаблоны и их использование в архитектурном проекте. Использование модели предметной области при решении задачи проектирования. Пакеты в UML. Связи между пакетами UML и их использование в представлении архитектуры. Задача архитектора «Идентификация проектных элементов». Определение классов, подлежащих декомпозиции. Упаковка концептуальных классов по пакетам проекта. Зависимости между пакетами. Архитектурный шаблон «Слои» и его использование при проектировании ИС. Задача архитектора «Идентификация проектных механизмов». Понятие проектного шаблона. Представление шаблонов в CASE-средстве. Типы архитектурных механизмов и их представление в визуальной модели.

Задача проектировщика «Проектирование прецедентов». Реализация прецедента на этапе проектирования. Применение архитектурных механизмов. Представление компонентов в проекте. Использование интерфейсов на диаграммах последовательности UML. Задача проектировщика «Проектирование подсистем». Отличие пакетов и подсистем. Структурный класс UML и его использование. Операции интерфейса и их реализация в проекте подсистемы. Диаграмма последовательности UML. Динамическое и статическое представление дизайна подсистемы. Использование проектных механизмов. Зависимости классов и пакетов.

Проектные классы. Ответственности и операции классов. Выбор атрибутов классов. Атрибуты и понятие состояния объекта. Шаблоны выявления состояний. Специальные состояния – начальное и конечное. Супер состояние и его использование. Диаграммы состояний UML. Условия на диаграмме состояний. Правила перехода из состояния в состояние и из запись на диаграмме. Диаграммы деятельности языка UML. Их использование в рамках унифицированного процесса. Семантика деятельности. События, исключения. Конечные автоматы. Конечные автоматы в рамках унифицированного процесса. Конечные автоматы и язык UML. Состояния и переходы. Другие

Тема 5. Документирование проекта архитектуры программной системы.

Нефункциональные требования. Эффективность, сложность, масштабируемость, расширяемость, адаптируемость, безопасность, зависимости системы. Учёт нефункциональных требований в проекте архитектуры ИС. Трассируемость требований в архитектуре при проектировании. Объектный язык ограничений OCL. Возможности OCL и область применения. Синтаксис и типы выражений. OCL в диаграммах UML.

Тема 6. Прямое и обратное проектирование.

Различия рисования и визуального моделирования. Репозиторий CASE-средства и синхронизация его содержимого на UML диаграммах. Понятие каркасного кода при прямом проектировании. Сравнение возможностей CASE-средств визуального моделирования.

Тема 7. Архитектурные стили и шаблоны.

Описание и оценка архитектурного стиля. Архитектурные фреймворки: 4+1, TOGAF, RM-ODP, SOMF.

Тема 8. Шаблоны проектирования GRASP. Шаблоны проектирования GoF.

Реализация и развертывание ИС. Рабочий поток реализации унифицированного процесса. Учёт особенностей реализации и развертывания при проектировании архитектуры ИС. Связь программного и аппаратного обеспечения. Влияние особенностей аппаратного обеспечения на программную архитектуру.Arteфакты реализации. Диаграммы развертывания языка UML. Применение диаграмм развертывания для проектирования процесса развертывания и адаптации ИС.

Тема 9. Архитектурные стили сетевых и распределенных программных систем.

Защищенность как ключевое нефункциональное требование к ИС. Проектирование защищенных ИС. Принципы и шаблоны проектирования архитектуры для обеспечения защищенности программной системы. Контроль доступа. Проектирование и разработка программной системы — командная работа. Роли и распределение задач в команде. Значение организации процесса при проектировании. Ещё раз о лучших практиках разработки и проектирования: гибкость, итеративность, инкрементальность процесса. История создания и развития языка UML. Группа Object Management Group (OMG). Заключительные замечания.

7.3. Содержание самостоятельной работы

Тема 1. Архитектура программных систем. Унифицированный процесс разработки.

Изучение основной и дополнительной литературы по теме, интернет-источников. Разбор лабораторного практикума. Разработка программных решений. Подготовка к текущему контролю.

Тема 2. Требования к программной системе.

Изучение основной и дополнительной литературы по теме, интернет-источников. Разбор лабораторного практикума. Разработка программных решений. Подготовка к текущему контролю.

Тема 3. Классы и объекты. Модель предметной области.

Изучение основной и дополнительной литературы по теме, интернет-источников. Разбор лабораторного практикума. Разработка программных решений. Подготовка к текущему контролю.

Тема 4. Анализ и проектирование. Рабочий поток проектирования.

Изучение основной и дополнительной литературы по теме, интернет-источников. Разбор лабораторного практикума. Разработка программных решений. Подготовка к текущему контролю.

Тема 5. Документирование проекта архитектуры программной системы.

Изучение основной и дополнительной литературы по теме, интернет-источников. Разбор лабораторного практикума. Разработка программных решений. Подготовка к текущему контролю.

Тема 6. Прямое и обратное проектирование.

Изучение основной и дополнительной литературы по теме, интернет-источников. Разбор лабораторного практикума. Разработка программных решений. Подготовка к текущему контролю.

Тема 7. Архитектурные стили и шаблоны.

Изучение основной и дополнительной литературы по теме, интернет-источников. Разбор лабораторного практикума. Разработка программных решений. Подготовка к текущему контролю.

Тема 8. Шаблоны проектирования GRASP. Шаблоны проектирования GoF.

Изучение основной и дополнительной литературы по теме, интернет-источников. Разбор лабораторного практикума. Разработка программных решений. Подготовка к текущему контролю.

Тема 9. Архитектурные стили сетевых и распределенных программных систем.

Изучение основной и дополнительной литературы по теме, интернет-источников. Разбор лабораторного практикума. Разработка программных решений. Подготовка к текущему контролю.

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

Не предусмотрено.

7.4. Электронное портфолио обучающегося

Материалы не размещаются.

7.5. Методические рекомендации по выполнению контрольной работы

Не предусмотрено.

7.6 Методические рекомендации по выполнению курсовой работы

Не предусмотрено.

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:

1. Антипов В. А., Бубнов А. А., Пылькин А. Н., Столчнев В. К.. Введение в программную инженерию [Электронный ресурс]:учебник для студентов вузов, обучающихся по направлению подготовки 2.09.03.04 "Программная инженерия" (квалификация "бакалавр"). - Москва: КУРС: ИНФРА-М, 2018. - 336 с. – Режим доступа: <http://znanium.com/go.php?id=944151>

2. Гагарина Л. Г., Кокорева Е. В., Сидорова-Виснадул Б. Д., Гагарина Л. Г.. Технология разработки программного обеспечения [Электронный ресурс]:учебное пособие для студентов вузов, обучающихся по направлениям подготовки 09.04.01 и 09.03.03 «Информатика и вычислительная техника». - Москва: ФОРУМ: ИНФРА-М, 2019. - 400 с. – Режим доступа: <http://znanium.com/go.php?id=1011120>

Дополнительная литература:

1. Федорова Г. Н.. Разработка, внедрение и адаптация программного обеспечения отраслевой направленности [Электронный ресурс]:учебное пособие (09.02.05 Приклад. информатика (по отр.), ПМ "Разработка, адаптация и внедрение програм. обеспечения отр. направленности) для профессиональных образовательных организаций. - Москва: КУРС: ИНФРА-М, 2016. - 336 с. – Режим доступа: <http://znanium.com/go.php?id=544732>

2. Виноградова Е. Ю.. Паттерны программирования [Электронный ресурс]:учебное пособие. - Екатеринбург: [Издательство УрГЭУ], 2017. - 27 с. – Режим доступа: <http://lib.usue.ru/resource/limit/ump/18/p490388.pdf>

3. Плещев В. В., Шишков Е. И.. Основы программирования на языках С++ и С# с примерами и упражнениями [Электронный ресурс]:учебное пособие. - Екатеринбург: [Издательство УрГЭУ], 2018. - 286 с. – Режим доступа: <http://lib.usue.ru/resource/limit/ump/18/p490708.pdf>

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

Перечень лицензионное программное обеспечение:

Astra Linux Common Edition. Договор № 1 от 13 июня 2018, акт от 17 декабря 2018. Срок действия лицензии - без ограничения срока.

Libre Office. Лицензия GNU LGPL. Срок действия лицензии - без ограничения срока.

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

MySQL Community Server. Стандартная общественная лицензия GNU (GPL). Срок действия лицензии - без ограничения срока.

Microsoft Visio. Акт предоставления прав № Tr020776 от 07.04.2017. Срок действия лицензии - 07.02.2020.

Microsoft Project. Акт предоставления прав № Tr020776 от 07.04.2017. Срок действия лицензии - 07.02.2020.

Система контроля версий Git. Лицензия GNU GPL v2 and GNU LGPL v2.1. Срок действия лицензии - без ограничения срока.

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

Формализация моделирования

<https://openedu.ru/course/spbstu/FOMO/>

Технологии программирования

<https://openedu.ru/course/urfu/PRGRMM/>

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.