

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Курганский государственный университет»

Кафедра «Программного обеспечения автоматизированных систем»



УТВЕРЖДАЮ:
Ректор

Н.В. Дубив

2020 г.

Рабочая программа учебной дисциплины

АРХИТЕКТУРЫ
ИНФОРМАЦИОННО-ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ
образовательной программы высшего образования –
программы магистратуры

09.04.04 Программная инженерия
направленность

*Методы и алгоритмы интеллектуальной обработки данных
в информационно-вычислительных системах*

формы обучения – очная

Рабочая программа составлена в соответствии с учебными планами программы магистратуры Программная инженерия (Методы и алгоритмы интеллектуальной обработки данных в информационно-вычислительных системах) для очной форм обучения, утвержденными 29.08.2019 г.

Рабочая программа одобрена на заседании кафедры Программного обеспечения автоматизированных систем 31.08.2019 года, протокол № 1.

Рабочую программу разработал

доцент кафедры ПОАС



О.С. Черепанов

Заведующий
кафедрой ПОАС



Т.Р. Змызгова

Согласовано:

Начальник
Управления
образовательной деятельности



С.Н. Синицын

Специалист
по учебно-методической работе
Учебно-методического отдела



Г.В. Казанкова

1. ОБЪЕМ ДИСЦИПЛИНЫ

Общая трудоемкость – 5 зач. ед. (180 акад. часа)

Виды учебной работы	Распределение трудоемкости по семестрам и видам учебных занятий (акад. часов)	
	Всего	1-й семестр
Аудиторные занятия:	72	72
Лекции	36	36
Лабораторные занятия	36	36
Самостоятельная работа:	108	108
Подготовка к экзамену	27	27
Прочие виды	81	81
Вид промежуточной аттестации		Экзамен
Общая трудоемкость	180	180

2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ

Дисциплина «Архитектуры информационных систем» включена в модуль «Информационно-вычислительные системы» обязательной части блока 1 учебного плана. Для освоения дисциплины необходимы компетенции в области проектирования программных систем, формируемые соответствующими дисциплинами программ бакалавриата или специалитета. Результаты изучения дисциплины используются при освоении профильных дисциплин, включенных в модули «Информационно-вычислительные системы», «Технологии распределённой обработки данных».

3. ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ ОБУЧЕНИЯ

Основная цель изучения дисциплины – формирование у обучающихся профессиональных знаний и практических навыков в области проектирования архитектур программного обеспечения, используя современные архитектурные паттерны и стили.

Задачи дисциплины:
изучение:

- общеизвестных архитектурных стилей и шаблонов, применяемых при проектировании монолитных программных систем;
- паттернов проектирования, используемых при разработки микросервисных архитектур.
- подходов к интеграции программного обеспечения.

практическое освоение:

- методов и средств проектирования программного обеспечения;
- программных фреймворков и библиотек, применяемых при разработке распределенного программного обеспечения.

Компетенции, формируемые в результате освоения дисциплины:

- способность разрабатывать оригинальные алгоритмы и программные средства, в том числе с использованием современных интеллектуальных технологий, для решения профессиональных задач (ОПК-2);
- способность разрабатывать и модернизировать программное и аппаратное обеспечение информационных и автоматизированных систем (ОПК-5);
- способность самостоятельно приобретать с помощью информационных технологий и использовать в практической деятельности новые знания и умения, в том числе в новых областях знаний, непосредственно не связанных со сферой деятельности (ОПК-7).

В результате освоения дисциплины обучающийся должен демонстрировать следующие **результаты обучения**:

Должен знать:

- общеизвестные паттерны, используемые при проектировании архитектур программного обеспечения (для ОПК-7).

Должен уметь:

- проектировать архитектуры программного обеспечения различного назначения и сложности (для ОПК-2, ОПК-5).

Должен владеть:

- программными средствами визуального проектирования архитектур программных систем (ОПК-5);
- навыками работы с программными средствами (фреймворками, библиотеками) при реализации распределенного программного обеспечения (для ОПК-7).

4. СОДЕРЖАНИЕ ДИСЦИПЛИНЫ

4.1 Учебно-тематический план

Разделы дисциплины		Часов контактной работы с преподавателем	
№	Наименование	Лекции	Лабор. занятия
1	Архитектуры монолитных систем	16	14
	Рубежный контроль №1	-	2
2	Сервис-ориентированные архитектуры	20	18
	Рубежный контроль №2	-	2
Всего по дисциплине:		36	36

4.2 Содержание лекционных занятий

Наименование и содержание лекции	Часов контактной работы с преподавателем
Раздел №1. АРХИТЕКТУРЫ МОНОЛИТНЫХ СИСТЕМ	
Лекция 1. Введение в проектирование архитектуры программных системам Понятие функционал, концепт, пакет, модуль, компонент, приложение, система. Определения архитектуры программного обеспечения. Признаки плохой архитектуры. Результаты применения плохой архитектуры. Расслоение систем. Многоуровневые архитектуры. Слой домена. Слой источника данных. Слой представления. Паттерн «Сценарий транзакций». Паттерны «Модель предметной области». Паттерн «Модуль таблицы».	2
Лекция 2. Архитектура EBI (Сущность — Граница — Интерактор) Описание структурных элементов архитектуры: Entity (Сущность), Interactor (Интерактор) и Boundary (Граница). Структурные диаграммы	2

архитектуры. Диаграмма взаимодействия уровней. Жизненный цикл объектов архитектуры EBI. Отличие EBI от MVC. Разбор примера приложения с архитектурой EBI.	
<p>Лекция 3. Предметно-ориентированное проектирование (DDD)</p> <p>Предметные области, подобласти, ограниченные контексты. Сущности, объекты-значения. Агрегаты. Службы. Фабрики. Хранилища. Интеграция ограниченных контекстов. Разбор примера приложения с архитектурой DDD.</p>	2
<p>Лекция 4. Гексагональная-архитектура</p> <p>Эволюция от многоуровневой архитектуры. Понятие порт и адаптер. Типы адаптеров. Изоляция механизмов доставки. Уровень домена. Уровень приложений. Уровень фреймворков. Связь между уровнями. Варианты использования/Команды. Командная шина. Разбор примера приложения с гексагональной архитектурой.</p>	2
<p>Лекция 5. Луковая архитектура</p> <p>Связь луковой архитектуры с многослойной архитектурой. Структурная диаграмма луковой архитектуры. Слои: слой домена, слой репозитория, слой услуг, слой пользовательского интерфейса. Взаимодействие слоев. Преимущества и недостатки луковой архитектуры. Разбор примера приложения с луковой архитектурой.</p>	2
<p>Лекция 6. Чистая архитектура</p> <p>Свойства хорошей архитектуры по Роберту Мартину. Варианты использования. Разделение уровней архитектуры и вариантов использования. Границы компонентов. Службы. Бизнес-правила: сущности и варианты использования. Кричащая архитектура. Общая диаграмма чистой архитектуры. Описание слоев. Изоляция слоев. Взаимодействие слоев. Слой фреймворков и устройств. Слой адаптеров (Контроллеры, Шлюзы, Презентеры). Слой вариантов использования. Слой сущностей. Связь с луковой, гексагональной архитектурой, MVC и EBI. Разбор примера приложения с чистой архитектурой.</p>	2
<p>Лекция 7. Событийно-ориентированная архитектура</p> <p>Понятие события. Способы уменьшения связности компонентов при использовании событий. Выполнение асинхронных задач. Паттерны: Event Notification, Event-Carried State Transfer, Event-Sourcing. Преимущества и недостатки паттернов. Разбор примера приложения, использующего паттерн Event-Sourcing.</p>	2
<p>Лекция 8. CQS и CQRS</p> <p>Понятия запроса и команды. Шаблон команды. Принцип разделения методов. Командная шина. Структурная диаграмма CQS. Достоинства и недостатки CQS. Структурная диаграмма CQRS и принцип ее работы. Сторона запроса. Хранения данных запроса. Сторона команд. Масштабирование запросов. Команды и валидация. Автономные компоненты. Модель домена в CQRS. Синхронизация хранилища запросов. Разбор примера приложения, использующий паттерн CQRS.</p>	2
Раздел №2. СЕРВИС-ОРИЕНТИРОВАННЫЕ АРХИТЕКТУРЫ	
<p>Лекция 9. Введение в сервис-ориентированные архитектуры</p> <p>Понятие сервиса. История развития: от CORBA к Web-сервисам, а далее к микросервисам. Очереди сообщений. Брокеры сообщений.</p>	2

Enterprise Service Bus (ESB). Микросервисы. Архитектура равиоли. SOAP-протокол. Архитектурный стиль REST. Концепции GraphQL. Разбор примеров приложений с сервис-ориентированной архитектурой.	
<p align="center">Лекция 10. Брокер сообщений RabbitMQ</p> <p>Введение в RabbitMQ, основные определения. Пример отправки и получения сообщения из очереди. Мониторинг очередей. Работа с очередями: круговая диспетчеризация, подтверждение получения сообщения, справедливая отправка сообщений. Обмены (Exchanges). Временные очереди. Привязки. Прямой обмен. Множественные привязки. Подписки. Маршрутизация сообщений. Ключи маршрутизации. Темы (Topic Exchange). Удаленный вызов процедур.</p>	2
<p align="center">Лекция 11. Введение в микросервисы. Шаблоны разбиения на микросервисы</p> <p>Введение в микросервисную архитектуру: основные понятия и определения. Недостатки монолитной архитектуры. Сравнение микросервисной архитектуры с другими видами сервис-ориентированных архитектур. Достоинства и недостатки микросервисной архитектуры. Краткий обзор шаблонов микросервисной архитектуры. Определение системных операций. Шаблон разбиения на микросервисы. Трудности разбиения на микросервисы.</p>	2
<p align="center">Лекция 12. Межпроцессное взаимодействие в микросервисной архитектуре</p> <p>Обзор способов межпроцессного взаимодействия. Описание API в микросервисной архитектуре. Форматы сообщений. Взаимодействие на основе удаленного вызова процедур: REST, gRPC. Шаблон «Предохранитель». Шаблоны обнаружение сервисов. Взаимодействие с помощью асинхронного обмена сообщениями. Брокеры сообщений. Транзакционный обмен сообщениями.</p>	2
<p align="center">Лекция 13. Управление транзакциями в микросервисной архитектуре</p> <p>Распределенные транзакции. Шаблон «Повествование». Задача координации повествований. Повествования на основе хореографии. Повествование на основе оркестрации. Проблема недостаточной изолированности.</p>	2
<p align="center">Лекция 14. Проектирование бизнес-логики в микросервисной архитектуре</p> <p>Шаблоны организации бизнес-логики. Шаблон «Сценарий транзакции». Шаблон «Доменная модель». Применение DDD в микросервисной архитектуре. Публикация доменных событий. Разработка бизнес-логики с использованием порождения событий. Реализация хранилища событий. Совместное использование повествований и поражения событий.</p>	4
<p align="center">Лекция 15. Реализация запросов в микросервисной архитектуре</p> <p>Шаблон «Объединение API». Преимущества и недостатки объединения API. Шаблон CQRS в микросервисной архитектуре. Шаблон «API-шлюз». Реализация API-шлюза.</p>	2
<p align="center">Лекция 16. Развертывание микросервисов</p> <p>Развертывание сервисов с использованием пакетов для конкретных языков. Развертывание в виде виртуальных машин. Развертывание сервисов в виде контейнеров (Docker). Развертывание сервисов с использованием Kubernetes.</p>	2

Лекция 17. <i>Переход от монолитной архитектуры к микросервисной.</i> Причины переход на микросервисы. Понятие «удушение монолита». Стратегии перехода на микросервисы. Проектирование взаимодействия между микросервисами. Реализация новой возможности в виде сервиса. Способы разбиения на части.	2
Всего часов лекционных занятий	36

4.3 Лабораторные работы

Наименование и содержание лабораторной работы	Часов контактной работы с преподавателем
Раздел №1. АРХИТЕКТУРЫ МОНОЛИТНЫХ СИСТЕМ	
Лабораторная работа № 1. <i>Реализация приложения с монолитной архитектурой</i> Разработка архитектуры приложения с использованием типовых архитектурных шаблонов и стилей. Реализация приложения. Защита проекта. Тема согласовывается с преподавателем.	14
Рубежный контроль №1.	2
Раздел №2. СЕРВИС-ОРИЕНТИРОВАННЫЕ АРХИТЕКТУРЫ	
Лабораторная работа № 2. <i>Реализация приложения с микросервисной архитектурой</i> Перевод архитектуры приложения из лабораторной работы №1 к микросервисной архитектуре. Реализация микросервисов и их способов взаимодействия. Развертывания приложения в виде контейнеров с использованием Docker и Kubernetes. Защита проекта.	18
Рубежный контроль №2.	2
Всего часов Лабораторных занятий	36

5 МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ОБУЧАЮЩИХСЯ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ

Конспект лекций (краткий обзор рассматриваемых на лекциях вопросов) представлен в формате мультимедийных презентаций и включен в состав учебно-методического комплекса дисциплины.

Основная цель проведения лабораторных занятий – получение практических навыков применения архитектурных шаблонов и стилей при проектировании программного обеспечения.

Программой изучения дисциплины предусмотрено выполнение двух лабораторных работ.

Самостоятельная работа обучающихся по освоению дисциплины включает подготовку к промежуточной аттестации (экзамену), подготовку к проведению

рубежного контроля, проработку лекционного материала, выполнение и оформление результатов лабораторных работ.

Для текущего контроля успеваемости по очной форме обучения преподавателем используется балльно-рейтинговая система контроля и оценки академической активности. Поэтому настоятельно рекомендуется тщательно прорабатывать материал дисциплины при самостоятельной работе, участвовать во всех формах обсуждения и взаимодействия, как на лекциях, так и на лабораторных занятиях в целях лучшего освоения материала и получения высокой оценки по результатам освоения дисциплины.

Рекомендуемое распределение трудоемкости самостоятельной работы приведено в таблице.

Таблица 5.1 – Рекомендуемая трудоемкость самостоятельной работы

Виды самостоятельной работы	Рекомендуемая трудоемкость, акад. часов
Самостоятельное изучение тем:	45
Брокер запросов Kafka	12
Программные средства тестирования ПО с микросервисной архитектурой	11
Использование сервисной шины BS для организации потоков данных	11
Программные средства развертывания и мониторинга микросервисов	11
Изучение материала лекционного курса:	27
Подготовка и выполнение лабораторных работ (по 2 ч. на каждое занятие)	32
Подготовка к рубежному контролю (по 2 часа на контроль)	4
Подготовка к экзамену	27
Всего:	108

6 ФОНД ОЦЕНОЧНЫХ СРЕДСТВ

6.1 Перечень оценочных средств

Фонд оценочных средств по дисциплине включает следующие компоненты, включенные в состав учебно-методического комплекса дисциплины:

1. Балльно-рейтинговая система контроля и оценки академической активности обучающихся в КГУ
2. Банк заданий к рубежным контролям №1, №2.
3. Вопросы для подготовки к экзамену по дисциплине.
4. Образцы отчетов по практическим заданиям.

6.2 Система бально-рейтинговой оценки работы обучающихся по дисциплину

Таблица 6.2 – Рейтинговые бальные оценки по дисциплине

Виды контроля/аттестации по дисциплине	Содержание	Максимальная оценка	
		За одну аттестацию	Всего
Текущий контроль	Посещение лекций	16x18	18
	Выполнения практических заданий	26x16	32
Рубежный контроль	№1. Тема: «Архитектуры монолитных приложений»	1	10
	№2. Тема: «Сервис-ориентированные архитектуры»	1	10
Промежуточная аттестация (экзамен)		30	30
Максимальная итоговая оценка, баллов			100

Пересчет 100-балльной рейтинговой оценки по дисциплине в традиционную (4-балльную) оценку и в оценку ECTS (Общеввропейская система учета учебной работы) производится в соответствии с таблицей 6.3.

Таблица 6.3 – Соответствие шкал оценивания

Рейтинговая оценка, баллов	Виды оценок промежуточной аттестации		
	Традиционная оценка		Оценка ECTS
91-100	Отлично (5)	Зачтено	A
84-90			Хорошо (4)
74-83	C		
68-73	Удовлетворительно (3)		
61-67			E
51-60	Неудовлетворительно (2)	Не зачтено	Fx
0-50			F

Для допуска к экзамену обучающийся должен набрать по итогам текущего и рубежного контроля не менее 50 баллов и при этом он должен выполнить и защитить все лабораторные работы.

Для получения оценки «удовлетворительно» автоматически (без сдачи экзамена) достаточно набрать 68 баллов по результатам текущего и рубежного контроля в течение семестра.

Обучающемуся, набравшему в течение семестра не менее 68 баллов, преподаватель вправе добавить до 30 дополнительных (бонусных) баллов за активность на учебных занятиях, оригинальность принимаемых решений при выполнении лабораторных работ и индивидуальных контрольных заданий и выставить «автоматически» оценку «хорошо» или «отлично».

В случае если по результатам текущего и рубежного контроля набрано менее 50 баллов, он может набрать недостающее количество баллов, выполнив дополнительные индивидуальные задания до конца зачетной недели семестра.

Ликвидация академических задолженностей, возникших из-за разности в учебных планах при переводе или восстановлении, также проводится путем выполнения дополнительных индивидуальных заданий.

Состав дополнительных заданий, их количество, формы выполнения и максимальные балльные оценки определяются преподавателем и доводятся до обучающихся в момент выдачи заданий.

6.3 Процедура оценивания результатов освоения дисциплины

Рубежный контроль №1 и №2 проводится в форме фронтального тестирования по теоретической части дисциплины (тематические разделы №1 и №2). Тест содержит 20 вопросов, расчетное время проведения тестирования – 40 минут. Оценивается количество правильных ответов на задания теста и соответственно начисляется балл. Обучающийся, ответивший правильно менее, чем на 8 заданий теста, считается не прошедшим тестирование и обязан повторно пройти этот тест во время консультации по дисциплине.

Экзамен по дисциплине проводится в традиционной форме: обучающийся выполняет задания билета, включающего два теоретических вопроса, и отвечает экзаменатору. Оценивается полнота и правильность ответов на вопросы билета, а также его эрудиция в смежных вопросах.

Результаты текущего контроля успеваемости и экзамена заносятся преподавателем в экзаменационную ведомость, которая сдается в организационный отдел института в день экзамена, а также выставляются в зачетную книжку обучающегося.

6.4 Примерные оценочных средств для рубежных контролей и экзамена

Примеры вопросов для проведения рубежного контроля №1

1. Что понимается под «Объектом-значением» в предметно-ориентированном проектировании (DDD) ?

а) Логически целостный объект, определяемый совокупностью индивидуальных черт.

б) Объект, который представляет описательный аспект предметной области и не имеет индивидуального существования, собственной идентичности.

в) Объект, ответственный за создание других объектов.

г) Контейнер объектов в виде концептуального множества, который управляет их жизненным циклом и предоставляет способы доступа клиентского кода к ним.

2. Какого слоя нет в луковой архитектуре?

а) Слой домена.

б) Слой репозитория.

в) Слой пользовательского интерфейса.

г) Слой презентера.

3. О чем гласит принцип CQS?

а) Метод должен быть либо командой, выполняющей какое-то действия, либо запросом, возвращающим данные, но не одновременно командой и запросом.

б) Направление взаимодействия в многоуровневой архитектуре должно идти от внешних слоев к центру.

в) Программные сущности (классы, модули, функции) должны быть открыты для расширения, но не для модификации.

г) Метод-команда должен возвращать всегда результат.

Примеры вопросов для проведения рубежного контроля №2

1. Какой общеизвестной стратегии создания микросервисной архитектуры не существует?
 - а) Разбиение на сервисы по бизнес-возможностям.
 - б) Разбиение на сервисы по проблемным областям.
 - в) Разбиение на сервисы по ролям.
2. Как называется аномалия, вызванная нехваткой изолированности, в микросервисной архитектуре, при которой одно повествование перезаписывает изменения, внесенные другим, не читая их при этом:
 - а) Потеря обновления.
 - б) «Грязное» чтение.
 - в) Нечеткое/неповторяемое чтение.
3. Какой стратегии перехода с монолитной архитектуры к микросервисной не существует?
 - а) Реализация новой возможности в виде сервисов.
 - б) Разделение уровня представления и внутренних компонентов.
 - в) Разбиение монолита путем оформления функциональности в виде сервисов.
 - г) Разбиение монолита на более мелкие монолиты путем использования агрегатов.

Примеры вопросов для подготовки к экзамену по дисциплине

1. Введение в проектирование архитектуры программных систем. Основные определения понятия. Принципы SOLID.
2. Архитектура EBI. Пример приложения с архитектурой EBI.
3. Принципы предметно-ориентированного проектирования.
4. Чистая архитектура. Пример приложения с чистой архитектурой.
5. Межпроцессорное взаимодействие в микросервисной архитектуре. Шаблон «Предохранитель».

6. Развертывание сервисов в виде контейнеров (Docker).
7. Переход от монолита к микросервисной архитектуре. Шаблон «Реализация новой возможности в виде сервисов».

6.5. Фонд оценочных средств

Полный банк заданий для текущего, рубежных контролей и промежуточной аттестации по дисциплине, показатели, критерии, шкалы оценивания компетенций, методические материалы, определяющие процедуры оценивания образовательных результатов, приведены в учебно-методическом комплексе дисциплины.

7 ОСНОВНАЯ И ДОПОЛНИТЕЛЬНАЯ УЧЕБНАЯ ЛИТЕРАТУРА

7.1 Основная литература

1. Мартин Р. Чистая архитектура. Искусство разработки программного обеспечения. — СПб.: Питер, 2018. — 352 с.
2. Эванс Э. Предметно-ориентированное проектирование (DDD): структуризация сложных программных систем. : Пер. с англ. — М.: "И .Д. Вильямс", 2011. — 448 с.
3. Стопфорд Б. Проектирование событийно-ориентированных систем: Концепции и шаблоны проектирования сервисов потоковой обработки данных с использованием Apache Kafka / ; Пер. с англ. Главный редактор: Анастасия Овсянникова; научные редакторы: Дмитрий Чумак, Иван Сидоров; Перевод: Владимир Жданов — 2-е изд., испр. — Иркутск : ITSumma Press, 2019. — 175 с.
5. Ричардсон К. Микросервисы. Паттерны разработки и рефакторинга. — СПб.: Питер, 2019. — 544 с.

7.2 Дополнительная литература

1. Хорсдал К. Микросервисы на платформе .NET. — СПб.: Питер, 2018. — 352 с.
2. Бёрнс Б. Распределенные системы. Паттерны проектирования. — СПб.: Питер, 2019. — 224 с.
3. Ньюмен С. Создание микросервисов. — СПб.: Питер, 2016. — 304 с.
4. Greg Y. CQRS Documents [Электронный документ] URL: https://cqrs.files.wordpress.com/2010/11/cqrs_documents.pdf

7.3 Информационно-справочные материалы

1. Хроники архитектур программного обеспечения
<https://herbertograca.com/2017/07/03/the-software-architecture-chronicles>

2. Блог Криса Фидао <https://fideloper.com/hexagonal-architecture>
3. Официальный сайт Джерфи Палермо <https://jeffreypalermo.com>
4. Официальный сайт Уди Дахама <https://udidahan.com>

8 МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

8.1 Техническое обеспечение

№	Наименование	Использование
1	Комплект: ноутбук, медиа-проектор, экран	Для демонстрации иллюстративного материала при чтении лекций.
2	Персональный компьютер стандартной комплектации	Используется в качестве инструмента и объекта исследования при выполнении лабораторных и контрольных работ.

8.2 Программное обеспечение

№	Наименование	Использование
1	Редактор разработки диаграмм UML.	При выполнении курсовой работы и для подготовки документации
2	Интегрированная среда разработки программных систем	
3	Текстовый редактор	

Аннотация
рабочей программы учебной дисциплины

АРХИТЕКТУРЫ
ИНФОРМАЦИОННО-ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ
образовательной программы высшего образования –
программы магистратуры

09.04.04 Программная инженерия
направленность

*Методы и алгоритмы интеллектуальной обработки данных
в информационно-вычислительных системах*

формы обучения – очная

Трудоемкость освоения дисциплины – 5 зач. ед. (180 акад. часов)

Семестры: 1-й

Форма промежуточной аттестации: экзамен

Содержание дисциплины

Раздел 1. Архитектуры монолитных систем

Раздел 2. Сервис-ориентированные архитектуры