

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

2019 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 – заочная

Рабочая программа дисциплины «Архитектура информационно-вычислительных систем» составлена в соответствии с учебными планами программы магистратуры Программная инженерия (Методы и алгоритмы интеллектуальной обработки данных в информационно-вычислительных системах) заочной форм обучения, утвержденными 29.08.2019 г.

Рабочая программа одобрена на заседании кафедры Программного обеспечения автоматизированных систем 30.08.2019 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 – 4 зач. ед. (144 акад. часа)

Виды учебной работы	Распределение трудоемкости по семестрам и видам учебных занятий (акад. часов)	
	Всего	4-й семестр
Аудиторные занятия:	18	18
Лекции	10	10
Практические занятия	8	8
Самостоятельная работа:	126	126
Контрольная работа	18	18
Подготовка к экзамену	27	27
Прочие виды	81	81
Вид промежуточной аттестации		Экзамен
Общая трудоемкость	144	144

2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ

Дисциплина «Архитектуры информационных систем» включена в модуль «Программные системы и сети» 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 «Высокопроизводительные системы обработки данных», «Программные системы и сети».

3. ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ ОБУЧЕНИЯ

Основная цель изучения дисциплины – формирование у студентов 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 и модернизировать программное и аппаратное обеспечение информационных и автоматизированных систем (ОПК-5);
- способность применять при решении профессиональных задач методы и средства получения, хранения, переработки и трансляции информации посредством современных компьютерных технологий, в том числе, в глобальных компьютерных сетях (ОПК-7).

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

Должен знать:

- общеизвестные паттерны, используемые при проектировании архитектур программного обеспечения (ОПК-5, ОПК-7).

Должен уметь:

- проектировать архитектуры программного обеспечения различного назначения и сложности (ОПК-7).

Должен владеть:

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

4. СОДЕРЖАНИЕ ДИСЦИПЛИНЫ

4.1 Учебно-тематический план

Разделы дисциплины		Часов контактной работы с преподавателем	
№	Наименование	Лекции	Прак. занятия
		1	Архитектуры монолитных систем
2	Сервис-ориентированные архитектуры	4	4
Всего по дисциплине:		10	8

4.2 Содержание лекционных занятий

Наименование и содержание лекции	Часов контактной работы с преподавателем
Раздел №1. АРХИТЕКТУРЫ МОНОЛИТНЫХ СИСТЕМ	
<p>Лекция 1. Введение в проектирование архитектуры программных систем. Основные архитектуры. Часть 1.</p> <p>1) Введение в архитектуру программных систем Понятие функционал, концепт, пакет, модуль, компонент, приложение, система. Определения архитектуры программного обеспечения. Признаки плохой архитектуры. Результаты применения плохой архитектуры. Расслоение систем. Многоуровневые архитектуры.</p> <p>2) Архитектура EBI (Сущность — Граница — Интерактор). Описание структурных элементов архитектуры: Entity (Сущность), Interactor (Интерактор) и Boundary (Граница). Структурные диаграммы архитектуры. Диаграмма взаимодействия уровней. Жизненный цикл объектов архитектуры EBI. Отличие EBI от MVC. Разбор примера приложения с архитектурой EBI.</p>	2

<p>3) Предметно-ориентированное проектирование (DDD). Предметные области, подобласти, ограниченные контексты. Сущности, объекты-значения. Агрегаты. Службы. Фабрики. Хранилища. Интеграция ограниченных контекстов. Разбор примера приложения с архитектурой DDD.</p>	
<p style="text-align: center;">Лекция 2. Основные архитектуры. Часть 2.</p> <p>1) Гексагональная архитектура. 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> <p>2) Луковая архитектура 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> <p>3) Чистая архитектура 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.</p>	2
<p style="text-align: center;">Лекция 3. Основные архитектуры. Часть 3.</p> <p>1) Событийно-ориентированная архитектура 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> <p>2) CQS и CQRS 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 style="text-align: center;">Лекция 4. Введение в сервис-ориентированные архитектуры. Микросервисная архитектура. Шаблоны разбиения на микросервисы</p> <p>Понятие сервиса. История развития: от CORBA к Web-сервисам, а далее к микросервисам. Очереди сообщений. Брокеры сообщений. Enterprise Service Bus (ESB). Микросервисы. Архитектура равиоли. SOAP-протокол. Архитектурный стиль REST. Концепции GraphQL. Разбор примеров приложений с сервис-ориентированной архитектурой. Введение в микросервисную архитектуру: основные понятия и</p>	2

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>Лекция 5. Межпроцессное взаимодействие в микросервисной архитектуре. Проектирование бизнес-логики в микросервисной архитектуре. Реализация запросов 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. 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. Шаблон «Объединение API». Преимущества и недостатки объединения API. Шаблон CQRS в микросервисной архитектуре. Шаблон «API-шлюз». Реализация API-шлюза.</p>	2
Всего часов лекционных занятий	10

4.3 Практические работы

Наименование и содержание практической работы	Часов контактной работы с преподавателем
Раздел №1. АРХИТЕКТУРЫ МОНОЛИТНЫХ СИСТЕМ	
<p>Практическая работа № 1. Реализация приложения с монолитной архитектурой</p> <p>Разработка архитектуры приложения с использованием типовых архитектурных шаблонов и стилей. Реализация приложения. Защита проекта. Тема согласовывается с преподавателем.</p>	4
Раздел №2. СЕРВИС-ОРИЕНТИРОВАННЫЕ АРХИТЕКТУРЫ	
<p>Практическая работа № 2. Реализация приложения с микросервисной архитектурой</p> <p>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.</p>	4
Всего часов Практических занятий	8

4.4 Контрольная работа

Контрольная работа выполняется студентами заочной формы обучения и предполагает подготовку реферата по заданной теме.

5 МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ СТУДЕНТОВ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ

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

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

Программой изучения дисциплины предусмотрено выполнение двух лабораторных работ по соответствующим разделам дисциплины.

Для текущего контроля успеваемости 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 практических занятиях в целях лучшего освоения материала и получения высокой оценки по результатам освоения дисциплины.

Самостоятельная работа студентов по освоению дисциплины включает подготовку к промежуточной аттестации (экзамену), выполнение контрольной работы (для студентов заочной формы обучения), проработку лекционного материала, выполнение и оформление результатов практических работ. Рекомендуемое распределение трудоемкости самостоятельной работы приведено в таблице 5.1.

Таблица 5.1 – Рекомендуемая трудоемкость самостоятельной работы

Виды самостоятельной работы	Рекомендуемая трудоемкость, акад. часов
Изучение материала лекционного курса	36
Подготовка и выполнение лабораторных работ	8
Самостоятельное изучение тем разделов:	37
Брокеры сообщений	12
Шаблоны разбиения на микросервисы	12
Управление транзакциями в микросервисной архитектуре	13
Выполнение контрольной работы	18
Подготовка к экзамену	27
Всего:	126

6 ФОНД ОЦЕНОЧНЫХ СРЕДСТВ

6.1 Перечень оценочных средств

1. Балльно-рейтинговая система контроля и оценки академической активности обучающихся в КГУ.
2. Вопросы для подготовки к экзамену по дисциплине.

3. Задания для выполнения контрольной работы.
4. Образцы отчетов по практическим заданиям.

6.2 Система балльно-рейтинговой оценки работы обучающихся по дисциплине

Таблица 6.1 – Рейтинговые балльные оценки по дисциплине

Вид учебной работы	Количество баллов
Посещение лекций	$36 \times 5 = 156$
Выполнение практической работы	$206 + 206 = 406$
Контрольная работа	156
Экзамен	306

Таблица 6.2 – Соответствие шкал оценивания

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

Для допуска к экзамену студент должен набрать по итогам текущего контроля не менее 50 баллов и при этом он должен выполнить и защитить все практические работы и контрольную работу.

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

Студенту, 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 оценку «автоматически» «хорошо» или «отлично»..

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

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

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

6.3 Процедура оценивания результатов освоения дисциплины

Экзамен по дисциплине проводится в традиционной форме: студент выполняет задания билета, включающего два теоретических вопроса, и отвечает экзаменатору. Оценивается полнота и правильность ответов студента на вопросы билета, а также его эрудиция в смежных вопросах. Каждый вопрос оценивается в 15 баллов. Результаты текущего контроля и экзамена заносится преподавателем в экзаменационную ведомость, которая сдается в организационный отдел института в день экзамена, а также выставляется в зачетную книжку обучающегося.

6.4. Примеры оценочных средств для экзамена

Примерные вопросы для экзамена

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».

Примерные темы контрольной работы

1. Проектирование бизнес-логики в микросервисной архитектуре.
2. Программные средства развертывания микросервисов.
3. Реализация запросов в микросервисной архитектуре.

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

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

7.3 Информационно-справочные материалы

10. Хроники архитектур программного обеспечения
<https://herbertograca.com/2017/07/03/the-software-architecture-chronicles>
11. Блог Криса Фидао <https://fideloper.com/hexagonal-architecture>
12. Официальный сайт Джерфи Палермо <https://jeffreypalermo.com>
13. 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	RabbitMQ или Apache Kafka	Фреймворк для реализации взаимодействия микросервисов.
2	Docker	Программное обеспечение для развертывания приложения с микросервисной архитектурой.
3	Kubernetes	Программное обеспечение для управления контейнеризированными приложениями.

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 – заочная

Трудоемкость освоения дисциплины – 4 зач. ед. (144 акад. часов)

Семестры: 4-й

Форма промежуточной аттестации: экзамен

Содержание дисциплины

Раздел 1. Архитектуры монолитных систем

Раздел 2. Сервис-ориентированные архитектуры