

Министерство науки и высшего образования Российской Федерации  
федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«Курганский государственный университет»  
(КГУ)

Кафедра «**Программное обеспечение автоматизированных систем**»



УТВЕРЖДАЮ:  
Первый проректор  
/ Щербич С.Н. /

« 30 » августа 2019 г.

## Рабочая программа

Архитектура программных систем

образовательной программы высшего образования –  
программы бакалавриата

**09.03.04 Программная инженерия**

Направленность:

**Программное обеспечение автоматизированных систем**

Формы обучения: очная, заочная

Курган 2019

Рабочая программа дисциплины «Архитектура программных систем» составлена в соответствии с учебными планами по программе бакалавриата «Программная инженерия» (Программное обеспечение автоматизированных систем), утвержденными для очной формы обучения «29» августа 2019 года, для заочной формы обучения «29» августа 2019 года.

Рабочая программа дисциплины одобрена на заседании кафедры «Программное обеспечение автоматизированных систем» «30» августа 2019 года, протокол № 1.

Рабочую программу составили:

Доцент кафедры  
«Программное обеспечение  
автоматизированных систем»

О.С. Черепанов

Согласовано:

Заведующий кафедрой  
«Программное обеспечение  
автоматизированных систем»  
к.т.н., доцент

Т.Р. Змызгова

Специалист по учебно-методической  
работе Учебно-методического отдела

Г.В. Казанкова

Начальник Управления образовательной  
деятельности

С.Н. Синицын

## 1. ОБЪЕМ ДИСЦИПЛИНЫ

Всего: 4 зачетных единиц трудоемкости (144 академических часов)

### Очная форма обучения

Вид учебной работы	На всю дисциплину	Семестр
		7
<b>Аудиторные занятия (контактная работа с преподавателем), всего часов</b>	<b>48</b>	<b>48</b>
<b>в том числе:</b>		
Лекции	16	16
Практические работы	32	32
Аудиторные занятия в интерактивной форме, часов	-	-
<b>Самостоятельная работа, всего часов</b>	<b>96</b>	<b>96</b>
<b>в том числе:</b>		
Подготовка к экзамену	27	27
Другие виды самостоятельной работы	33	33
Курсовая работа	36	36
<b>Вид промежуточной аттестации</b>	<b>экзамен</b>	<b>экзамен</b>
<b>Общая трудоемкость дисциплины и трудоемкость по семестрам, часов</b>	<b>144</b>	<b>144</b>

### Заочная форма обучения

Вид учебной работы	На всю дисциплину	Семестр
		8
<b>Аудиторные занятия (контактная работа с преподавателем), всего часов</b>	<b>10</b>	<b>10</b>
<b>в том числе:</b>		
Лекции	4	4
Практические работы	6	6
Аудиторные занятия в интерактивной форме, часов	-	-
<b>Самостоятельная работа, всего часов</b>	<b>134</b>	<b>134</b>
<b>в том числе:</b>		
Подготовка к экзамену	27	27
Другие виды самостоятельной работы (самостоятельное изучение тем (разделов) дисциплины)	71	71
Курсовая работа	36	36
<b>Вид промежуточной аттестации</b>	<b>экзамен</b>	<b>экзамен</b>
<b>Общая трудоемкость дисциплины и трудоемкость по семестрам, часов</b>	<b>144</b>	<b>144</b>



## 2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ

Дисциплина «Архитектура программных систем» относится к вариативной части блока 1 модуль «Программное и аппаратное обеспечение информационно-коммуникационных систем».

Изучение дисциплины базируется на результатах обучения, сформированных при изучении следующих дисциплин:

- Информатика.
- Основы программирования.
- Алгоритмы и структуры данных.
- Объектно-ориентированное программирование.
- Конструирование программного обеспечения.
- Машинно-ориентированное программирование.

Результаты обучения по дисциплине необходимы для изучения дисциплин: «Технологии проектирования информационных систем», «Распределенные вычисления» и выполнения выпускной квалификационной работы.

## 3. ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ ОБУЧЕНИЯ

Целью освоения дисциплины «Администрировании программных систем» является формирование теоретических знаний и практических навыков проектирование архитектур программных систем.

Задачами дисциплины являются изучение принципов и архитектурных паттернов проектирования программных систем.

Компетенции, формируемые в результате освоения дисциплины:

- владение навыками использования операционных систем, сетевых технологий, систем управления базами данных (ПК-6);
- способность проводить установку, настройку и оптимизацию функционирования прикладного программного обеспечения (ПК-11).

В результате изучения дисциплины обучающийся должен

*Знать*

- принципы объектно-ориентированного проектирования SOLID (ПК-6);
- принцип расслоения систем, предназначение и взаимосвязь основных слоев (ПК-6).
- паттерны проектирования программных систем (ПК-11).

*Уметь*

- применять принципы и архитектурные паттерны при проектировании информационных систем (ПК-6).

*Владеть:*

- современными программными средствами проектирования архитектур программных систем (ПК-6).

## 4. СОДЕРЖАНИЕ ДИСЦИПЛИНЫ

### 4.1. Учебно-тематический план

#### Очная форма обучения

№	Наименование раздела	Количество часов контактной работы с преподавателем	
		Лекции	Практические работы
1	Принципы дизайна и организации компонентов программных систем	2	0
2	Паттерны проектирования программных систем	6	14
	Рубежный контроль 1		2
3	Архитектурные паттерны проектирования	8	14
	Рубежный контроль 2		2
<b>Всего:</b>		<b>16</b>	<b>32</b>

#### Заочная форма обучения

№	Наименование раздела	Количество часов контактной работы с преподавателем	
		Лекции	Практические работы
1	Принципы дизайна и организации компонентов программных систем	1	0
2	Паттерны проектирования программных систем	1	3
3	Архитектурные паттерны проектирования	2	3
<b>Всего:</b>		<b>4</b>	<b>6</b>



## 4.2. Содержание лекционных занятий

Наименование и содержание лекции	Часов контактной работы с преподавателем	
	Очная форма	Заочная форма
<b>Раздел 1. Принципы дизайна и организации компонентов программных систем</b>		
<p><b>Лекция 1. Введение в архитектуру программных систем. Принципы SOLID.</b></p> <p>Понятие архитектуры и дизайна. Последствия применения «слабой» архитектуры. Ретроспектива развития архитектурных паттернов и стилей. Принцип единственной ответственности. Принцип открытости/закрытости. Принцип подстановки Барбары Лисков. Принцип разделения интерфейсов. Принцип инверсии зависимости. Компоненты системы. Связанность компонентов. Сочетаемость компонентов.</p>	2	1
<b>Раздел 2. Паттерны проектирования программных систем</b>		
<p><b>Лекция 2. Введение в паттерны проектирования</b></p> <p>Понятие паттерны проектирования. Классификация паттернов. Порождающие паттерны. Фабричный метод. Абстрактная фабрика. Строитель. Прототип. Одиночка. Примеры реализации порождающих паттернов.</p>	2	1
<p><b>Лекция 3. Структурные паттерны</b></p> <p>Назначение, структура, реализация и примеры применения паттернов: адаптер, мост, компоновщик, декоратор, фасад, легковес и заместитель.</p>	2	
<p><b>Лекция 4. Поведенческие паттерны</b></p> <p>Назначение, структура, реализация и примеры применения паттернов: цепочка обязанностей, команда, итератор, посредник, снимок, наблюдатель, состояние, стратегия, шаблонный метод и посетитель.</p>	2	

<b>Раздел 3. Архитектурные паттерны проектирования</b>		
<p><b>Лекция 5. Расслоение системы. Слой бизнес-логики</b></p> <p>Концепция слоев системы. Развитие модели слоев в программных системах. Краткое описание основных слоев системы. Место функционирования слоев. Организация бизнес-логики. Сценарий транзакции. Модель предметной области. Модуль таблицы. Уровень служб.</p>	2	
<p><b>Лекция 6. Типовые решения источников данных</b></p> <p>Шлюз таблицы. Шлюз записи данных. Активная запись. Преобразователь данных. Типовые решения моделирования поведения: единица работы, коллекция объектов, загрузка по требованию. Паттерны моделирования структуры: поле идентификации, отображение внешних ключей, отображение с помощью таблицы ассоциации, отображение зависимых объектов. Реализация наследования в реляционной базе данных.</p>	2	2
<p><b>Лекция 7. Архитектурные паттерны.</b></p> <p>Структура и поведение паттернов Model-View-Controller, Model-View-Presenter и Model-View-ViewModel.</p>	2	
<p><b>Лекция 8. Сервис-ориентированные архитектуры</b></p> <p>История развития сервис-ориентированных архитектур. Понятие сервис и микросервис. Микросервисная архитектура. Шаблоны разбиения на микросервисы. Межпроцессорное взаимодействие в микросервисной архитектуре.</p>	2	0
<b>Итого:</b>	16	4



### 4.3. Практические работы

Но- мер раз- дела, темы	Наименование раздела	Наименование практи- ческой работы	Норматив времени, час.	
			Очная фор- ма обучения	Заочная форма обу- чения
1	Паттерны проектирования программных систем	Реализация паттернов проектирования	14	3
		Рубежный контроль №1	2	-
2	Архитектурные паттерны проектирования	Реализация программного обеспечения с архитектурой MVC, MVP или MVVM	14	3
		Рубежный контроль № 2	2	-
<b>Всего:</b>			<b>32</b>	<b>6</b>

#### 4.4 Курсовая работа (для заочной формы обучения)

В процессе выполнения курсовой работы студент выполняет самостоятельную разработку программного обеспечения в соответствии с индивидуальным заданием, согласованным и утвержденным в установленном порядке.

### 5. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ОБУЧАЮЩИХСЯ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ

Во время лекций по дисциплине студентам рекомендуется конспектировать теоретический материал, отмечая важные моменты, на которые заострил внимание преподаватель, участвовать в опросах и дискуссиях. Перед лекцией необходимо повторить выданный материал, зафиксировать непонятные моменты, чтобы обсудить их на занятии. Конспект лекций представлен в виде мультимедийных презентаций и включен в состав методического комплекса дисциплины.

Практические работы предусмотрены по двум разделам дисциплины: «Паттерны проектирования программных систем» и «Архитектурные паттерны проектирования». Все работы выполняются в соответствии с заданием, выданным преподавателем.

Самостоятельная работа студентов включает в себя проработку материала лекционного курса дисциплины, подготовку и выполнение практических работ, выполнение курсовой работы (для заочной формы обучения),



выполнение курсовой работы (для очной формы обучения), а также подготовку к рубежному контролю (для очной формы обучения) и к экзамену.

Для текущего контроля успеваемости обучения для очной формы обучения используется балльно-рейтинговая система контроля. Для получения высокой оценки настоятельно рекомендуется активно участвовать во время обсуждения материала дисциплины на лекционных и практических занятиях, а также тщательно его прорабатывать при самостоятельной работе.

### Рекомендуемый режим самостоятельной работы для очной формы обучения

Наименование вида самостоятельной работы	Рекомендуемая трудоемкость, акад. час.	
	Очная форма обучения	Заочная форма обучения
<b>Самостоятельное изучение тем дисциплины:</b>		
Принципы дизайна и организации компонентов программных систем	3	65
Паттерны проектирования программных систем	1	20
Архитектурные паттерны проектирования	1	20
<b>Подготовка к практическим работам (по 2 ч. на каждое занятие)</b>	1	25
<b>Подготовка к рубежным контролям (по 1 часу на каждый рубеж)</b>	28	6
<b>Выполнение курсовой работы</b>	2	-
<b>Подготовка к экзамену</b>	36	36
	27	27
<b>Всего:</b>	<b>96</b>	<b>134</b>

## 6. ФОНД ОЦЕНОЧНЫХ СРЕДСТВ ДЛЯ АТТЕСТАЦИИ ПО ДИСЦИПЛИНЕ

### 6.1. Перечень оценочных средств

1. Балльно-рейтинговая система контроля и оценки академической активности студентов в КГУ (для очной формы обучения).
2. Отчеты студентов по практическим занятиям.
3. Банк заданий к рубежным контролям № 1, № 2 (для очной формы обучения).
4. Курсовая работа.
6. Вопросы к экзамену.

## 6.2. Система балльно-рейтинговой оценки работы студентов по дисциплине

№	Наименование	Содержание					
<b>Очная форма обучения</b>							
1	Распределение баллов за семестры по видам учебной работы, сроки сдачи учебной работы (доводятся до сведения студентов на первом учебном занятии)	Распределение баллов за 7 семестр					
		Вид учебной работы:	Посещение лекций	Выполнение и защита результатов практических работ	Рубежный контроль №1	Рубежный контроль №2	Экзамен
		Балльная оценка:	16 x 8 = 8 б	3 б + 14 занятий = 42 б	10 б	10 б	30 б
2	Критерий пересчета баллов в традиционную оценку по итогам работы в семестре и экзамена	60 и менее баллов – неудовлетворительно; 61...73 – удовлетворительно; 74...90 – хорошо; 91...100 – отлично.					
3	Критерии допуска к промежуточной аттестации, возможности получения, возможность получения бонусных баллов	<p>Для допуска к промежуточной аттестации (экзамену) студент должен набрать не менее 50 баллов и выполнить все практические работы и курсовую работу.</p> <p>Для получения «автоматически» оценки «удовлетворительно» студенту необходимо набрать 68 баллов.</p> <p>По согласованию с преподавателем студенту, набравшему 68 баллов, могут быть добавлены дополнительные (бонусные) баллы за активность на консультациях, активное участие в научной и методической работе, оригинальность принятых решений в ходе выполнения лабораторных работ, за участие в значимых учебных и внеучебных мероприятиях кафедры и выставлена оценка «хорошо» или «отлично» автоматически.</p>					



4	Формы и виды учебной работы для неуспевающих (восстановившихся на курсе обучения) студентов для получения недостающих баллов в конце семестра	<p>В случае если к промежуточной аттестации (экзамену) набрана сумма менее 50 баллов, студенту необходимо набрать недостающее количество баллов за счет выполнения дополнительных заданий, до конца последней (зачетной) недели семестра. При этом необходимо проработать материал всех пропущенных практических работ.</p> <p>Формы дополнительных заданий (назначаются преподавателем):</p> <ul style="list-style-type: none"> <li>- выполнение и защита пропущенной практической работы (при невозможности дополнительного проведения практической работы преподаватель устанавливает форму дополнительного задания по тематике пропущенной практической работы самостоятельно) – до 8 баллов.</li> </ul> <p>Ликвидация академических задолженностей, возникших из-за разности в учебных планах при переводе или восстановлении, проводится путем выполнения дополнительных заданий, форма и объем которых определяется преподавателем.</p>															
5	Распределение баллов за семестры по видам учебной работы, сроки сдачи учебной работы ( <i>доводятся до сведения студентов на первом учебном занятии</i> )	<table border="1" style="width: 100%; text-align: center;"> <tr> <th colspan="4">Курсовая работа</th> </tr> <tr> <th>Качество представленной документации</th> <th>Качество выполненной работы</th> <th>Качество защиты</th> <th>Всего</th> </tr> <tr> <td>20 б</td> <td>50 б</td> <td>30 б</td> <td>100 б</td> </tr> </table>				Курсовая работа				Качество представленной документации	Качество выполненной работы	Качество защиты	Всего	20 б	50 б	30 б	100 б
Курсовая работа																	
Качество представленной документации	Качество выполненной работы	Качество защиты	Всего														
20 б	50 б	30 б	100 б														

### 6.3. Процедура оценивания результатов освоения дисциплины

Рубежные контроли (для очной формы обучения) проводятся в форме письменного тестирования, экзамен в виде ответа на вопросы.

Перед проведением рубежного контроля преподаватель прорабатывает со студентами основной материал соответствующих разделов дисциплины в форме краткой лекции-дискуссии.

Варианты заданий для рубежных контролей № 1, № 2 состоят из 10 вопросов.

На каждую подготовку к рубежному контролю студенту отводится 1 академический час.

Преподаватель оценивает в баллах результаты рубежных контролей каждого студента по количеству правильных ответов и заносит в ведомость учета текущей успеваемости.

На экзамене студенту предлагается ответить на 2 вопроса. Вопросы к экзамену доводятся до студентов на последней лекции в семестре. Каждый вопрос оценивается в 15 баллов. На подготовку ответа студенту отводится 1 астрономический час.

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

### 6.4. Примеры оценочных средств для рубежных контролей и экзамена

### **6.4.1 Примеры заданий для рубежного контроля №1**

1. Какой паттерн определяет общий интерфейс для создания объектов в суперклассе, позволяя своим подклассам конкретизировать тип создаваемого объекта в рамках общей иерархии?

- а) Фабричный метод.
- б) Абстрактная фабрика.
- в) Одиночка.
- г) Команда.

2. За что отвечает класс Director в паттерне «Строитель»?

- а) Определяет порядок вызова этапов построения сложного объекта.
- б) Определяет этапы построения объекта.
- в) Определяет тип строящегося объекта.
- г) Такого класса в паттерне не предусмотрено.

3. Какой структурный паттерн проектирования предоставляет упрощенный интерфейс к сложной системе классов, библиотеке или фреймворку?

- а) Фасад.
- б) Адаптер.
- в) Мост.
- г) Заместитель.

4. Какой структурный паттерн проектирования позволяет подставить вместо реальных объектов специальные объекты, которые перехватывают вызовы к оригинальному объекту, позволяя внедрить новый функционал?

- а) Заместитель.
- б) Фасад.
- в) Мост.
- г) Адаптер.

5. Какой паттерн позволяет сохранять и восстанавливать прошлые состояния объектов, не раскрывая подробностей их реализации.

- а) Снимок.
- б) Стратегия.
- в) Команда.
- г) Шаблонный метод.

### **6.4.2 Примеры заданий для рубежного контроля №2**

1. Слои «Домен» отвечает за:

- а) бизнес-логику приложений;
- б) представление услуг, отображение данных, обработку событий пользовательского интерфейса;



в) обращение к базе данных, обмену сообщениями, управление транзакциями.

2. Какого паттерна слоя «Источник данных» не существует?

- а) шлюз записи;
- б) активная записи;
- в) преобразователь данных;
- г) шлюз к хранилищу данных.

3. Какой паттерн предполагает размещение всей иерархии классов в одной таблице базы данных?

- а) Single Table Inheritance;
- б) Class Table Inheritance;
- в) Concrete Table Inheritance.

4. Представление в архитектурном паттерне «Model-View-Controller» отвечает за:

- а) представление данных и обработку команд контроллера;
- б) отображение данных модели, реагируя на ее изменение;
- в) интерпретацию действий пользователя, оповещая модель об необходимости изменений;
- г) такого компонента нет.

#### ***6.4.3 Примерный перечень вопросов для экзамена***

1. Определение архитектуры ПО и дизайна ПО. Последствия неудачного выбора архитектуры.

2. Принципы SOLID. Принцип единственной ответственности и принцип открытости/закрытости.

3. Принципы SOLID. Принцип подстановки Барбары Лисков и принцип разделения интерфейсов.

4. Принципы SOLID. Принцип инверсии зависимости.

5. Принципы организации компонентов. Понятие компонент системы. Связанность компонентов.

6. Принципы организации компонентов. Сочетаемость компонентов системы.

7. Понятие «Паттерн проектирования». Классификация паттернов. Взаимосвязь паттернов.

8. Порождающие паттерны. Фабричный метод и абстрактная фабрика.

9. Порождающие паттерны. Строитель, прототип и одиночка.

10. Структурные паттерны. Адаптер и мост.

11. Структурные паттерны. Компоновщик и декоратор.

12. Структурные паттерны. Фасад и легковес.

13. Поведенческие паттерны. Цепочка обязанностей и команда.

14. Поведенческие паттерны. Итератор и посредник.

15. Поведенческие паттерны. Снимок и наблюдатель.
16. Поведенческие паттерны. Состояние и стратегия.
17. Поведенческие паттерны. Шаablонный метод и посетитель.
18. Расслоение системы. Концепция слоев системы. Краткое описание основных слоев системы. Место функционирования слоев.
19. Организация бизнес-логики. Сценарий транзакции. Модель предметной области. Модуль таблицы. Уровень служб.
20. Типовые решения источников данных. Шлюз таблицы. Шлюз записи данных.
21. Типовые решения источников данных. Активная запись. Преобразователь данных.
22. Типовые решения моделирования поведения: Единица работы. Коллекция объектов. Загрузка по требованию.
23. Паттерны моделирования структуры. Поле идентификации. Отображение внешних ключей. Отображение с помощью таблицы ассоциации. Отображение зависимых объектов.
24. Реализация наследования в реляционных базах данных.
25. Архитектурный паттерн Model-View-Controller.
26. Архитектурный паттерн Model-View-Presenter.
27. Архитектурный паттерн Model-View-ViewModel.
28. История развития сервис-ориентированных архитектур. Понятие сервис и микросервис. Микросервисная архитектура.
29. Шаablоны разбиения на микросервисы.
30. Межпроцессорное взаимодействие в микросервисной архитектуре.

#### ***6.4.4 Примерные темы курсовой работы***

1. Программная реализация web-приложения по продаже товаров с применением паттерны MVVM.
2. Разработка медицинской информационной системы на базе микросервисной архитектуры с реализацией транзакций между микросервисами.

#### **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.



Аннотация  
рабочей программы учебной дисциплины

**АРХИТЕКТУРА ПРОГРАММНЫХ СИСТЕМ**  
образовательной программы высшего образования –  
программы бакалавриата

**09.03.04 Программная инженерия**

направленность

*Программное обеспечение автоматизированных систем*

формы обучения – очная и заочная

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

Семестры: 7-й для очной и 8-й для заочной формы обучения

Промежуточная аттестация: экзамен в 7-м семестре для очной и 8 заоч-  
ной формы обучения

Содержание дисциплины

- Раздел 1. Принципы дизайна и организации компонентов программных систем
- Раздел 2. Паттерны проектирования программных систем
- Раздел 3. Архитектурные паттерны проектирования