

МИНОБРНАУКИ РОССИИ  
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ  
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
«ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ» (ФГБОУ ВО «ВГУ»)

УТВЕРЖДАЮ

Заведующий кафедрой

ПиИТ

*наименование кафедры, отвечающей за реализацию дисциплины*



проф. Махортов С.Д.,

*подпись, расшифровка подписи*

11.03.2022 г.

**РАБОЧАЯ ПРОГРАММА УЧЕБНОЙ ДИСЦИПЛИНЫ**

**Б1.В.ДВ.02.01 Проектирование архитектуры программных систем**

*Код и наименование дисциплины в соответствии с учебным планом*

**1. Код и наименование направления подготовки/специальности:**

09.04.04 Программная инженерия

**2. Профиль подготовки/специализация:** Системное программирование

**3. Квалификация выпускника:** магистр

**4. Форма обучения:** очная

**5. Кафедра, отвечающая за реализацию дисциплины:** ПиИТ

**6. Составители программы:** Вахтин А.А., к.ф.-м.н., доц.

---

**7. Рекомендована:** НМС ФКН протокол № 3 от 25.02.2022 г. *(наименование  
рекомендующей структуры, дата, номер протокола,*

---

*отметки о продлении вносятся вручную)*

**8. Учебный год:** 2022/2023

**Семестр(ы):** 2

## 9. Цели и задачи учебной дисциплины

Целями освоения учебной дисциплины являются: изучение современных методов проектирования архитектуры сложных программных систем, комплексов и инструментов для разработки и поддержки документации программной системы.

Задачи учебной дисциплины: научить студентов навыкам группового программирования и конструирования архитектуры сложных программ в соответствии с современными подходами и требованиями, с возможностью дальнейшего расширения и сопровождения программ.

**10. Место учебной дисциплины в структуре ООП:** учебная дисциплина относится к части блока Б1, формируемой участниками образовательных отношений, дисциплина по выбору. К началу изучения предмета студенты должны иметь навык программирования, построения и анализа алгоритмов.

**11. Планируемые результаты обучения по дисциплине/модулю (знания, умения, навыки), соотнесенные с планируемыми результатами освоения образовательной программы (компетенциями) и индикаторами их достижения:**

Код	Название компетенции	Код(ы)	Индикатор(ы)	Планируемые результаты обучения
ПК-10	Способен определять варианты структур программного обеспечения информационных систем (программного средства), необходимые информационные потоки и исследовать варианты структур с использованием моделей различного уровня	ПК-10.2	Проводит формирование вариантов структуры системы (программного средства) и разрабатывает варианты реализации их реализации в рамках предлагаемых алгоритмических и программных решений	знать: Современные методы проектирования архитектуры программных систем. Языки и среды программирования. Системы контроля версий и технологии тестирования ПО. уметь: Проектировать архитектуру программных систем в коллективе. Тестировать и документировать разрабатываемое ПО. владеть (иметь навык(и)): Навыками программирования, сохранения программ в системах контроля версиями, тестирования и отладки разрабатываемого ПО.
		ПК-10.3	Проводит исследование альтернативных вариантов построения системы (программного средства) с использованием моделей различного уровня	знать: Современные методы проектирования архитектуры программных систем. Языки и среды программирования. Системы контроля версий и технологии тестирования ПО. уметь: Проектировать архитектуру программных систем в коллективе. Тестировать и документировать разрабатываемое ПО. владеть (иметь навык(и)): Навыками программирования, сохранения программ в системах контроля версиями, тестирования и отладки разрабатываемого ПО.

ПК-12	Способен выполнять разработку и изменение архитектуры программного обеспечения	ПК-12.1	Использует существующие типовые решения и шаблоны проектирования программного обеспечения	<p>знать:</p> <p>Современные методы проектирования архитектуры программных систем. Языки и среды программирования. Системы контроля версий и технологии тестирования ПО.</p> <p>уметь:</p> <p>Проектировать архитектуру программных систем в коллективе. Тестировать и документировать разрабатываемое ПО.</p> <p>владеть (иметь навык(и)):</p> <p>Навыками программирования, сохранения программ в системах контроля версиями, тестирования и отладки разрабатываемого ПО.</p>
ПК-13	Способен выполнять проектирование структур данных и баз данных	ПК-13.1	Применяет методы и средства проектирования программного обеспечения, структур данных, баз данных	<p>знать:</p> <p>Современные методы проектирования архитектуры программных систем. Языки и среды программирования. Системы контроля версий и технологии тестирования ПО.</p> <p>уметь:</p> <p>Проектировать архитектуру программных систем в коллективе. Тестировать и документировать разрабатываемое ПО.</p> <p>владеть (иметь навык(и)):</p> <p>Навыками программирования, сохранения программ в системах контроля версиями, тестирования и отладки разрабатываемого ПО.</p>
ПК-14	Способен проектировать архитектуру программного средства	ПК-14.1	Определяет состав компонентов программного средства	<p>знать:</p> <p>Современные методы проектирования архитектуры программных систем. Языки и среды программирования. Системы контроля версий и технологии тестирования ПО.</p> <p>уметь:</p> <p>Проектировать архитектуру программных систем в коллективе. Тестировать и документировать разрабатываемое ПО.</p> <p>владеть (иметь навык(и)):</p> <p>Навыками программирования, сохранения программ в системах контроля версиями, тестирования и отладки разрабатываемого ПО.</p>
		ПК-14.2	Определяет способы взаимодействия между программными подсистемами программного средства	<p>знать:</p> <p>Современные методы проектирования архитектуры программных систем. Языки и среды программирования. Системы контроля версий и технологии тестирования ПО.</p> <p>уметь:</p> <p>Проектировать архитектуру программных систем в коллективе. Тестировать и документировать разрабатываемое ПО.</p> <p>владеть (иметь навык(и)):</p> <p>Навыками программирования, сохранения программ в системах контроля версиями, тестирования и отладки разрабатываемого ПО.</p>

12. Объем дисциплины в зачетных единицах/час — 3/108.

## Форма промежуточной аттестации экзамен

### 13. Трудоемкость по видам учебной работы

Вид учебной работы		Трудоемкость			
		Всего	По семестрам		
			3 семестр		
Аудиторные занятия		32	32		
в том числе:	лекции	16	16		
	практические	0	0		
	лабораторные	16	16		
Самостоятельная работа		76	76		
в том числе: курсовая работа (проект)		0	0		
Форма промежуточной аттестации (экзамен – __ час.)		0	0		
Итого:		108	108		

#### 13.1. Содержание дисциплины

п/п	Наименование раздела дисциплины	Содержание раздела дисциплины	Реализация раздела дисциплины с помощью онлайн-курса, ЭУМК
<b>1. Лекции</b>			
1.1	Понятие архитектуры программных систем	Разработка. Развертывание. Эффективность работы. Сопровождение. Сохранение разнообразия вариантов. Независимость от устройств. Варианты использования. Эффективность работы. Независимость разработки. Сохранение разнообразия вариантов. Уровни. Дублирование.	
1.2	Уровни архитектуры программных систем	Проблемы монолитных программных систем. Компоненты развертывания. Поток выполнения. Локальные процессы. Службы. Уровни в архитектуре программных систем. Управление сложностью.	
1.3	Бизнес-правила	Сущности. Варианты использования. Модели запросов и ответов.	
1.4	Чистая архитектура программных систем	Правило независимостей. Типичный сценарий. Шаблоны. Шлюзы к базам данных. Преобразователи данных.	
1.5	Другие архитектуры программных систем	Тема архитектуры. Цель архитектуры. Фреймворки. Тестируемые архитектуры.	
1.6	Главный компонент архитектуры программных систем	Фасады. Границы. Пересечение потоков. Разбиение потоков. Привязка к оборудованию.	
1.7	Архитектура службы	Сервисная архитектура. Преимущества служб. Службы на основе компонентов. Сквозные задачи.	
1.8	Тесты	Тесты как компоненты системы. Проектирование для простоты тестирования. Программный интерфейс для тестирования. Безопасность.	

2. Лабораторные занятия			
2.1	Понятие архитектуры программных систем	Разработка. Развертывание. Эффективность работы. Сопровождение. Сохранение разнообразия вариантов. Независимость от устройств. Варианты использования. Эффективность работы. Независимость разработки. Сохранение разнообразия вариантов. Уровни. Дублирование.	
2.2	Уровни архитектуры программных систем	Проблемы монолитных программных систем. Компоненты развертывания. Потoki выполнения. Локальные процессы. Службы. Уровни в архитектуре программных систем. Управление сложностью.	
2.3	Бизнес-правила	Сущности. Варианты использования. Модели запросов и ответов.	
2.4	Чистая архитектура программных систем	Правило зависимостей. Типичный сценарий. Шаблоны. Шлюзы к базам данных. Преобразователи данных.	
2.5	Другие архитектуры программных систем	Тема архитектуры. Цель архитектуры. Фреймворки. Тестируемые архитектуры.	
2.6	Главный компонент архитектуры программных систем	Фасады. Границы. Пересечение потоков. Разбиение потоков. Привязка к оборудованию.	
2.7	Архитектура службы	Сервисная архитектура. Преимущества служб. Службы на основе компонентов. Сквозные задачи.	
2.8	Тесты	Тесты как компоненты системы. Проектирование для простоты тестирования. Программный интерфейс для тестирования. Безопасность.	

### 13.2. Темы (разделы) дисциплины и виды занятий

№ п/п	Наименование темы (раздела) дисциплины	Виды занятий (количество часов)				Всего
		Лекции	Практические	Лабораторные	Самостоятельная работа	
1.	Понятие архитектуры программных систем	2		2	10	14
2.	Уровни архитектуры программных систем	2		2	10	14
3.	Бизнес-правила	2		2	10	14
4.	Чистая архитектура программных систем	2		2	10	14
5.	Другие архитектуры программных систем	2		2	10	14
6.	Главный компонент архитектуры программных систем	2		2	8	12
7.	Архитектура службы	2		2	8	12
8.	Тесты	2		2	10	14
	Итого:	16		16	76	108

### 14. Методические указания для обучающихся по освоению дисциплины

Выполнение заданий не зависит от выбора языка программирования. Для практики группового программирования, и отслеживания процесса выполнения заданий используется система контроля версиями GitHub.

## 15. Перечень основной и дополнительной литературы, ресурсов интернет, необходимых для освоения дисциплины

а) основная литература:

№ п/п	Источник
1.	<i>Мартин Р. Чистая архитектура. Искусство разработки программного обеспечения. / Р. Мартин. — СПб.: Питер, 2018. — 352 с.</i>

б) дополнительная литература:

№ п/п	Источник
1.	<i>Макконнелл С. Совершенный код. Мастер-класс = Code Complete : пер. с англ. / С. Макконнелл. — М.: Русская редакция, 2012. — XX, 867 с.</i>

в) информационные электронно-образовательные ресурсы (официальные ресурсы интернет):

№ п/п	Ресурс
1.	<a href="https://lib.vsu.ru/?p=4&amp;t=8">https://lib.vsu.ru/?p=4&amp;t=8</a>
2.	<a href="https://github.com/Software-engineering-Labs">https://github.com/Software-engineering-Labs</a>

## 16. Перечень учебно-методического обеспечения для самостоятельной работы

№ п/п	Источник
1	<i>Макконнелл С. Совершенный код. Мастер-класс = Code Complete : пер. с англ. / С. Макконнелл. — М.: Русская редакция, 2012. — XX, 867 с.</i>
2	<i>Мартин Р. Чистая архитектура. Искусство разработки программного обеспечения. / Р. Мартин. — СПб.: Питер, 2018. — 352 с.</i>
3	<a href="https://github.com/Software-engineering-Labs">https://github.com/Software-engineering-Labs</a>

## 17. Образовательные технологии, используемые при реализации учебной дисциплины, включая дистанционные образовательные технологии (ДОТ), электронное обучение (ЭО), смешанное обучение):

На лабораторных и практических занятиях допускается использовать студентами любой язык программирования и любую среду программирования. Разрабатываемые и сопровождаемые проекты следует сохранять в виде репозитория в системе контроля версий GitHub в специально зарегистрированном для этого разделе.

## 18. Материально-техническое обеспечение дисциплины:

Компьютеры с установленной средой программирования и доступом в интернет.

## 19. Оценочные средства для проведения текущей и промежуточной аттестаций

Порядок оценки освоения обучающимися учебного материала определяется содержанием следующих разделов дисциплины:

№ п/п	Наименование раздела дисциплины (модуля)	Компетенция(и)	Индикатор(ы) достижения компетенции	Оценочные средства
-------	--	----------------	-------------------------------------	--------------------

1.	Понятие архитектуры программных систем	ПК-10 ПК-12 ПК-13 ПК-14	ПК-10.2, ПК-10.3 ПК-12.1 ПК-13.1 ПК-14.1, ПК-14.2	Лабораторная работа
2.	Уровни архитектуры программных систем	ПК-10 ПК-12 ПК-13 ПК-14	ПК-10.2, ПК-10.3 ПК-12.1 ПК-13.1 ПК-14.1, ПК-14.2	Лабораторная работа
3.	Бизнес-правила	ПК-10 ПК-12 ПК-13 ПК-14	ПК-10.2, ПК-10.3 ПК-12.1 ПК-13.1 ПК-14.1, ПК-14.2	Лабораторная работа
4.	Чистая архитектура программных систем	ПК-10 ПК-12 ПК-13 ПК-14	ПК-10.2, ПК-10.3 ПК-12.1 ПК-13.1 ПК-14.1, ПК-14.2	Лабораторная работа
5.	Другие архитектуры программных систем	ПК-10 ПК-12 ПК-13 ПК-14	ПК-10.2, ПК-10.3 ПК-12.1 ПК-13.1 ПК-14.1, ПК-14.2	Лабораторная работа
6.	Главный компонент архитектуры программных систем	ПК-10 ПК-12 ПК-13 ПК-14	ПК-10.2, ПК-10.3 ПК-12.1 ПК-13.1 ПК-14.1, ПК-14.2	Лабораторная работа
7.	Архитектура службы	ПК-10 ПК-12 ПК-13 ПК-14	ПК-10.2, ПК-10.3 ПК-12.1 ПК-13.1 ПК-14.1, ПК-14.2	Лабораторная работа
8.	Тесты	ПК-10 ПК-12 ПК-13 ПК-14	ПК-10.2, ПК-10.3 ПК-12.1 ПК-13.1 ПК-14.1, ПК-14.2	Лабораторная работа
Промежуточная аттестация форма контроля – зачет с оценкой				По результатам лабораторных работ

## 20 Типовые оценочные средства и методические материалы, определяющие процедуры оценивания

**20.1 Текущий контроль успеваемости** Контроль успеваемости по дисциплине осуществляется с помощью контроля и оценки следующих лабораторных заданий:

### 20.1.1 Перечень лабораторных заданий

1. Банковская система. Необходимо реализовать возможность регистрации клиента, открытие кредитных и депозитных счетов. Подсчет начисленных процентов, комиссий и т.п. Формирование отчетов по клиентам и делам банка. Предусмотреть возможность открытие счетов по заданным тарифам, корректировку тарифов, создание индивидуального тарифа.
2. Программное обеспечение для поликлиники или диагностического центра. Предусмотреть возможности регистрации пациента. Просмотр и редактирование расписания врачей, запись пациента на прием. Ввод результатов обследования и заключения врача. Распечатка результатов анализов, заключений врача. Реализовать возможность доступа к данным анализов и заключений врача через личный кабинет пациента.

3. Интернет-магазин (автотоваров, бытовой химии, товаров для ремонта и т.п.). Продумать разделение товаров по категориям, поиск товара по параметрам. Формирование заказа, обработка заказа и т.п.
4. Система управления с помощью вызванных зрительных потенциалов. Продумать подключаемый модуль получения данных вызванных потенциалов с ЭЭГ, модуль обработки сигналов и модуль управления на основе команд, полученных из модуля обработки сигнала.
5. Система управления с помощью миоэлектрических потенциалов. Продумать подключаемый модуль получения данных вызванных потенциалов с ЭЭГ, модуль обработки сигналов и модуль управления на основе команд, полученных из модуля обработки сигнала.

Описание технологии проведения Группа студентов занимается на практике и лабораторных работах выданной им задачей. Промежуточные результаты сохраняют в системе контроля версий <https://github.com/Softwareengineering-Labs>, созданный ими код проверяется и оценивается.

Оценка знаний, умений и навыков, характеризующая этапы формирования компетенций в рамках изучения дисциплины осуществляется в ходе текущей и промежуточной аттестаций.

Текущая аттестация проводится в соответствии с Положением о текущей аттестации обучающихся по программам высшего образования Воронежского государственного университета. Текущая аттестация проводится на основе оценки выполнения практических задач.

## **20.2 Промежуточная аттестация**

Промежуточная аттестация по дисциплине осуществляется по итогам оценок лабораторных работ в текущей аттестации. Перечень заданий приведен выше.

Промежуточная аттестация проводится в соответствии с Положением о промежуточной аттестации обучающихся по программам высшего образования. При оценивании используются качественные шкалы оценок.