Министерство науки и высшего образования Российской Федерации

Муромский институт (филиал)

федерального государственного бюджетного образовательного учреждения высшего образования

«Владимирский государственный университет имени Александра Григорьевича и Николая Григорьевича Столетовых»

(МИ ВлГУ)

Кафедра *ПИн*

«УТВЕРЖДАЮ»
Заместитель директора по УР
_____Д.Е. Андрианов
19.05.2026

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

Архитектура и шаблоны проектирования

Направление подготовки *09.04.04 Программная инженерия*

Профиль подготовки *Технологии разработки интеллектуальных систем*

Семестр	Трудоемкость, час./зач. ед.	Лекции, час.	Практические занятия, час.	Лабораторные работы, час.	Консультация, час.	Контроль, час.	Всего (контактная работа), час.	СРС, час.	Форма промежуточного контроля (экз., зач., зач. с оц.)
2	**252 / 7**	**32**	**10**	**48**	**3,2**	**2,35**	**95,55**	**120,8**	**Экз.(35,65)**
Итого	**252 / 7**	**32**	**10**	**48**	**3,2**	**2,35**	**95,55**	**120,8**	**35,65**

Муром, 2026 г.

1. Цель освоения дисциплины

Целью освоения дисциплины является формирование у магистрантов системного понимания принципов, шаблонов и стилей архитектурного проектирования программного обеспечения, а также практических навыков применения современных методов разработки масштабируемых, гибких и поддерживаемых систем. Задачи дисциплины включают: изучение принципов SOLID и способов создания абстракций, устойчивых к изменениям требований; освоение стратегий инверсии управления (IoC) и разрешения зависимостей; приобретение умений применять шаблоны «Адаптер», «Мост», «Итератор», «Цепочка обязанностей», «Интерпретатор» и расширяемую фабрику; формирование компетенций в области Domain-Driven Design, интеграции программных решений через системы обмена сообщениями, а также в анализе проблем вертикального масштабирования и проектировании микросервисной архитектуры.

2. Место дисциплины в структуре ОПОП ВО

Дисциплина базируется на знаниях принципов объектно-ориентированного программирования, полученных на бакалавриате.

3. Планируемые результаты обучения по дисциплине

Планируемые результаты обучения по дисциплине, соотнесенные с планируемыми результатами освоения ОПОП (компетенциями и индикаторами достижения компетенций)

Формируемые компетенции (код, содержание компетенции)	Планируемые результаты обучения по дисциплине, в соответствии с индикатором достижения компетенции		Наименование оценочного средства
	Индикатор достижения компетенции	Результаты обучения по дисциплине	
ОПК-1 Способен самостоятельно приобретать, развивать и применять математические, естественнонаучные, социально-экономические и профессиональные знания для решения нестандартных задач, в том числе в новой или незнакомой среде и в междисциплинарном контексте	ОПК-1.1 Формулирует задачи и определяет пути их решения на основе оценки эффективности выбора с учетом специфики научных исследований в сфере обработки и передачи информации	основные критерии оценки эффективности архитектурных решений применительно к задачам обработки и передачи информации (ОПК-1.1) формулировать архитектурные задачи и обосновывать выбор способа их решения с учётом специфики научных исследований (ОПК-1.1) методами сравнительного анализа альтернативных архитектурных вариантов для выбора оптимального подхода в научно-исследовательских проектах (ОПК-1.1)	Вопросы к опросу
ОПК-4 Способен применять на практике новые научные принципы и методы исследований	ОПК-4.1 Применяет современные методы научного исследования и разработки программного обеспечения	современные методы научного исследования в области ПО (ОПК-4.1) применять методы эмпирического исследования для проверки гипотез о применимости шаблонов проектирования и архитектурных стилей (ОПК-4.1) навыками планирования и проведения научных экспериментов с архитектурой ПО (ОПК-4.1)	Вопросы к опросу

ОПК-7 Способен 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-7.1 Применяет лучшие практики, шаблоны и стили архитектурного проектирования	лучшие практики применения SOLID, шаблонов проектирования и архитектурных стилей (ОПК-7.1) выбирать и адаптировать подходящие архитектурные шаблоны и стили для решения конкретных задач проектирования с учётом контекста (ОПК-7.1) навыками интеграции принципов SOLID, расширяемой фабрики, стратегий разрешения зависимостей и Domain-Driven Design в промышленную архитектуру приложений (ОПК-7.1)	Вопросы к опросу

4. Структура и содержание дисциплины

Общая трудоемкость дисциплины составляет 7 зачетных единиц, 252 час.

4.1. Форма обучения: очная

Уровень базового образования: высшее.
Срок обучения 2г.

4.1.1. Структура дисциплины

№ п\п	Раздел (тема) дисциплины	Семестр	Контактная работа обучающихся с педагогическим работником							Самостоятельная работа	Форма текущего контроля успеваемости (по неделям семестра), форма промежуточной аттестации(по семестрам)
			Лекции	Практические занятия	Лабораторные работы	Контрольные работы	КП / КР	Консультация	Контроль		
1	Принципы проектирования и шаблоны	2	16	6	24					24	Устный опрос
2	Архитектура, интеграция и масштабирование	2	16	4	24					96,8	Устный опрос
Всего за семестр		252	32	10	48		+	3,2	2,35	120,8	Экз.(35,65)
Итого		252	32	10	48			3,2	2,35	120,8	35,65

4.1.2. Содержание дисциплины

4.1.2.1. Перечень лекций

Семестр 2

Раздел 1. Принципы проектирования и шаблоны

Лекция 1.

Схема применения SOLID принципов (2 часа).

Лекция 2.

Определение абстракций, устойчивых к изменениям требований (2 часа).

Лекция 3.

Расширяемая фабрика и IoC (2 часа).

Лекция 4.

Стратегии разрешения зависимостей IoC (2 часа).

Лекция 5.

Адаптер и мост (2 часа).

Лекция 6.

Domain Driving Design (2 часа).

Лекция 7.

Применение SOLID в промышленных фреймворках (2 часа).

Лекция 8.

Итератор (2 часа).

Раздел 2. Архитектура, интеграция и масштабирование

Лекция 9.

Понятие и задачи архитектуры решений (2 часа).

Лекция 10.

Жизненный цикл архитектуры ПО и полезные артефакты (2 часа).

Лекция 11.

Проблема вертикального масштабирования и синхронизация (2 часа).

Лекция 12.

Интеграция программного обеспечения (2 часа).

Лекция 13.

Системы обмена сообщениями (2 часа).

Лекция 14.

Построение архитектуры приложения, построенного на системе обмена сообщениями (2 часа).

Лекция 15.

Методологии разработки ПО (2 часа).

Лекция 16.

Микросервисная архитектура (2 часа).

4.1.2.2. Перечень практических занятий

Семестр 2

Раздел 1. Принципы проектирования и шаблоны

Практическое занятие 1

Заместитель (2 часа).

Практическое занятие 2

Декоратор и Посетитель (2 часа).

Практическое занятие 3

Шаблонный метод (2 часа).

Раздел 2. Архитектура, интеграция и масштабирование

Практическое занятие 4

Чистый код и рефакторинг (2 часа).

Практическое занятие 5

Создание микросервиса (2 часа).

4.1.2.3. Перечень лабораторных работ

Семестр 2

Раздел 1. Принципы проектирования и шаблоны

Лабораторная 1.

Команда (4 часа).

Лабораторная 2.

Расширяемая фабрика и IoC (4 часа).

Лабораторная 3.

Адаптер и мост (4 часа).

Лабораторная 4.

Состояние (4 часа).

Лабораторная 5.

Цепочка обязанностей (4 часа).

Лабораторная 6.

Интерпретатор (4 часа).

Раздел 2. Архитектура, интеграция и масштабирование

Лабораторная 7.

Определение абстракций, устойчивых к изменениям требований (4 часа).

Лабораторная 8.

SOLID и исключения (4 часа).

Лабораторная 9.

Проблема вертикального масштабирования и синхронизация (4 часа).

Лабораторная 10.

Системы обмена сообщениями (4 часа).

Лабораторная 11.

Создание микросервиса (4 часа).

Лабораторная 12.

Микросервисная архитектура (4 часа).

4.1.2.4. Перечень тем и учебно-методическое обеспечение самостоятельной работы

Перечень тем, вынесенных на самостоятельное изучение:

1. Принцип подстановки Лисков.
2. Инверсия зависимостей на практике.
3. Паттерн Стратегия и масштабирование.
4. Декоратор в каналах обмена.
5. Саги для микросервисных транзакций.
6. CQRS: разделение команд и запросов.
7. Event Sourcing и хранение состояния.
8. Репозиторий в предметно-ориентированном дизайне.
9. Антикоррупционный слой при интеграции.
10. Балансировка запросов в микросервисах.

Для самостоятельной работы используются методические указания по освоению дисциплины и издания из списка приведенной ниже основной и дополнительной литературы.

4.1.2.5. Перечень тем контрольных работ, рефератов, ТР, РГР, РПР

Не планируется.

4.1.2.6. Примерный перечень тем курсовых работ (проектов)

1. Проектирование расширяемой фабрики с IoC-контейнером.
2. Реализация адаптера и моста для масштабируемой системы.
3. Архитектура микросервиса на базе обмена сообщениями.
4. Применение цепочки обязанностей и интерпретатора в обработке данных.
5. Управление состоянием при вертикальном масштабировании и синхронизации.

5. Образовательные технологии

В процессе изучения дисциплины применяется контактная технология преподавания (за исключением самостоятельно изучаемых студентами вопросов). При проведении практических работ применяется имитационный или симуляционный подход. Шаги решения задач студентам демонстрируются при помощи мультимедийной техники. В дальнейшем студенты самостоятельно решают аналогичные задания.

6. Оценочные средства для текущего контроля успеваемости, промежуточной аттестации по итогам освоения дисциплины.

Фонды оценочных материалов (средств) приведены в приложении.

7. Учебно-методическое и информационное обеспечение дисциплины.

7.1. Основная учебно-методическая литература по дисциплине

1. Алексеев, В. А. Паттерны проектирования программных систем : методические указания к проведению лабораторных работ по курсу «Архитектура программных систем» / В. А. Алексеев. — Липецк : Липецкий государственный технический университет, ЭБС АСВ, 2016. — 33 с. — Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. — URL: https://www.iprbookshop.ru/74412.html (дата обращения: 15.05.2026). — Режим доступа: для авторизир. пользователей - https://www.iprbookshop.ru/74412.html

2. Минакова, О. В. Технологии программирования: паттерны проектирования в реализации JavaFX приложений : практикум / О. В. Минакова. — Воронеж : Воронежский государственный технический университет, ЭБС АСВ, 2020. — 156 с. — ISBN 978-5-7731-0911-2. — Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. — URL: https://www.iprbookshop.ru/111505.html (дата обращения: 15.05.2026). — Режим доступа: для авторизир. пользователей - https://www.iprbookshop.ru/111505.html

3. Аникеев, Д. В. Архитектура информационных систем : учебное пособие / Д. В. Аникеев. — Рязань : Рязанский государственный радиотехнический университет, 2022. — 72 с. — Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. — URL: https://www.iprbookshop.ru/137312.html (дата обращения: 15.05.2026). — Режим доступа: для авторизир. пользователей - https://www.iprbookshop.ru/137312.html

7.2. Дополнительная учебно-методическая литература по дисциплине

1. Кукарцев, В. В. Проектирование и архитектура информационных систем : учебник / В. В. Кукарцев, Р. Ю. Царев, О. А. Антамошкин. — Красноярск : Сибирский федеральный университет, 2019. — 192 с. — ISBN 978-5-7638-3620-2. — Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. — URL: https://www.iprbookshop.ru/100091.html (дата обращения: 15.05.2026). — Режим доступа: для авторизир. пользователей - https://www.iprbookshop.ru/100091.html

7.3. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине, включая перечень программного обеспечения и информационных справочных систем

В образовательном процессе используются информационные технологии, реализованные на основе информационно-образовательного портала института (www.mivlgu.ru/iop), и инфокоммуникационной сети института:

- предоставление учебно-методических материалов в электронном виде;
- взаимодействие участников образовательного процесса через локальную сеть института и Интернет;
- предоставление сведений о результатах учебной деятельности в электронном личном кабинете обучающегося.

Информационные справочные системы:

Электронная библиотека ВлГУ - http://e.lib.vlsu.ru/

электронная библиотечная системы "IPRBooks" (http://www.iprbookshop.ru/);

Программное обеспечение:

Google Chrome (Лицензионное соглашение Google)

PostgreSQL (Лицензия PostgreSQL)

IntelliJ IDEA (Apache License 2.0)

7.4. Перечень ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины

iprbookshop.ru

e.lib.vlsu.ru

mivlgu.ru/iop

8. Материально-техническое обеспечение дисциплины

Лаборатория программирования и баз данных

12 шт. компьютеров Intel Core i5-10150 3,70 GHz/ 16Gb(DDr4)/ SSD-150Gb / Haff 23,8’; проектор ACER P1100 DLP Projector EMEA; экран проекционный настенный DRAPPER Apex STAR; маршрутизатор Gigabit Switch TEG-S16S; плоттер HP Design Jet T610. Маркерная доска. Доступ к сети Интернет.

9. Методические указания по освоению дисциплины

Для успешного освоения теоретического материала обучающийся: знакомится со списком рекомендуемой основной и дополнительной литературы; уточняет у преподавателя, каким дополнительным пособиям следует отдать предпочтение; ведет конспект лекций и прорабатывает лекционный материал, пользуясь как конспектом, так и учебными пособиями.

На практических занятиях пройденный теоретический материал подкрепляется решением задач по основным темам дисциплины. Занятия проводятся в компьютерном классе, используя специальное программное обеспечение. Каждой подгруппе обучающихся преподаватель выдает задачу, связанную с разработкой и программной реализацией алгоритмов обработки информации. В конце занятия обучающие демонстрируют полученные результаты преподавателю и при необходимости делают работу над ошибками.

До выполнения лабораторных работ обучающийся изучает соответствующий раздел теории. Перед занятием студент знакомится с описанием заданий для выполнения работы, внимательно изучает содержание и порядок проведения лабораторной работы. Лабораторная работа проводятся в компьютерном классе. Обучающиеся выполняют индивидуальную задачу компьютерного моделирования в соответствии с заданием на лабораторную работу. Полученные результаты исследований сводятся в отчет и защищаются по традиционной методике в классе на следующем лабораторном занятии. Необходимый теоретический материал, индивидуальное задание, шаги выполнения лабораторной работы и требование к отчету приведены в методических указаниях, размещенных на информационно-образовательном портале института.

Самостоятельная работа оказывает важное влияние на формирование личности будущего специалиста, она планируется обучающимся самостоятельно. Каждый обучающийся самостоятельно определяет режим своей работы и меру труда, затрачиваемого на овладение учебным содержанием дисциплины. Он выполняет внеаудиторную работу и изучение разделов, выносимых на самостоятельную работу, по личному индивидуальному плану, в зависимости от его подготовки, времени и других условий.

Курсовая работа выполняется в соответствии с методическими указаниями на курсовую работу. Обучающийся выбирает одну из указанных в перечне тем курсовых работ, исходя из своих интересов, наличия соответствующих литературных и иных источников. В ходе выполнения курсовой работы преподаватель проводит консультации обучающегося. На заключительном этапе обучающийся оформляет пояснительную записку к курсовой работе и выполняет ее защиту в присутствии комиссии из преподавателей кафедры.

Форма заключительного контроля при промежуточной аттестации – экзамен. Для проведения промежуточной аттестации по дисциплине разработаны фонд оценочных средств и балльно-рейтинговая система оценки учебной деятельности студентов. Оценка по дисциплине выставляется в информационной системе и носит интегрированный характер, учитывающий результаты оценивания участия студентов в аудиторных занятиях, качества и своевременности выполнения заданий в ходе изучения дисциплины и промежуточной аттестации.

Программа составлена в соответствии с требованиями ФГОС ВО по направлению *09.04.04 Программная инженерия* и профилю подготовки *Технологии разработки интеллектуальных систем*
Рабочую программу составил *к.т.н., доцент Кульков Я.Ю.*_____

Программа рассмотрена и одобрена на заседании кафедры *ПИн*

протокол № 28 от 05.05.2026 года.
Заведующий кафедрой *ПИн* _____*Кульков Я.Ю.*
(Подпись)

Рабочая программа рассмотрена и одобрена на заседании учебно-методической комиссии факультета

протокол № 9 от 12.05.2026 года.
Председатель комиссии ФИТР _____ *Кутарова Е.И.*
(Подпись) (Ф.И.О.)

Приложение

Фонд оценочных материалов (средств) по дисциплине
Архитектура и шаблоны проектирования

1. Оценочные материалы для проведения текущего контроля успеваемости по дисциплине

Вопрос 1

Какая из перечисленных стратеций разрешения зависимостей в IoC предполагает внедрение зависимости через аргумент конструктора?

- Внедрение через свойство
- Внедрение через метод
- Внедрение через конструктор
- Внедрение через сервис-локатор

Вопрос 2

Какой принцип SOLID требует, чтобы абстракции не зависели от деталей, а детали зависели от абстракций?

- Принцип единственной ответственности
- Принцип открытости/закрытости
- Принцип подстановки Барбары Лисков
- Принцип инверсии зависимостей

Вопрос 3

Для какой цели в архитектуре приложений применяется шаблон «Адаптер»?

- Для разделения абстракции и реализации
- Для преобразования интерфейса одного класса в интерфейс, ожидаемый клиентом
- Для организации очереди сообщений между микросервисами
- Для последовательного обхода элементов коллекции без раскрытия её внутренней структуры

Вопрос 4

Что из перечисленного является характерным признаком событийно-ориентированной архитектуры (EDA) на основе систем обмена сообщениями?

- Синхронный вызов каждого сервиса с блокировкой потока
- Прямое соединение клиент-сервер через HTTP без посредников
- Асинхронная публикация событий в брокер и подписка заинтересованных сервисов
- Общая база данных для всех сервисов

Вопрос 5

В контексте Domain-Driven Design (DDD) что описывает термин «Антикоррупционный слой»?

- Слой, который предотвращает потерю данных при отказе сети
- Слой, который изолирует доменную модель от внешних систем, преобразуя чужие модели в свои
- Слой, который шифрует все сообщения между сервисами
- Слой, который агрегирует события из нескольких источников

Вопрос 6

Какая проблема решается использованием шаблона «Мост» (Bridge)?

- Необходимость создания большого числа подклассов при комбинации нескольких независимых измерений
- Управление транзакциями в распределённой системе
- Отложенная инициализация тяжёлого объекта
- Обеспечение единственного экземпляра класса в системе

Вопрос 7

При переходе от монолита к микросервисной архитектуре какой из перечисленных аспектов обычно становится наиболее сложным?

- Управление состоянием сессии и распределёнными транзакциями
- Написание кода на разных языках программирования
- Выбор интегрированной среды разработки
- Тестирование отдельного класса

Вопрос 8

Какое требование необходимо выполнить, чтобы правильно реализовать шаблон «Итератор»?

- Все элементы коллекции должны быть одного конкретного типа
- Коллекция должна быть неизменяемой в процессе итерации
- Предоставить механизм последовательного доступа к элементам без раскрытия внутреннего представления
- Гарантировать уникальность каждого элемента в коллекции

Вопрос 9

Что из перечисленного относится к недостаткам архитектуры, построенной исключительно на синхронных HTTP-вызовах (без обмена сообщениями)?

- Отсутствие поддержки SSL/TLS
- Трудность реализации циклических зависимостей между сервисами
- Жёсткая связанность, проблемы с масштабированием и низкая отказоустойчивость
- Невозможность передавать бинарные данные

Вопрос 10

В каком случае применение вертикального масштабирования (scale-up) предпочтительнее горизонтального (scale-out)?

- Когда требуется абсолютная отказоустойчивость на уровне дата-центров
- Когда приложение содержит глобальную блокирующую синхронизацию, не выносимую на несколько узлов
- Когда нагрузка носит случайный и резко переменный характер
- Когда бюджет на оборудование минимален

Общее распределение баллов текущего контроля по видам учебных работ для студентов

Рейтинг-контроль 1	устный опрос	20
Рейтинг-контроль 2	устный опрос	20
Рейтинг-контроль 3	устный опрос	20
Посещение занятий студентом		0
Дополнительные баллы (бонусы)		0
Выполнение семестрового плана самостоятельной работы		0

2. Промежуточная аттестация по дисциплине

Перечень вопросов к экзамену / зачету / зачету с оценкой. Перечень практических задач / заданий к экзамену / зачету / зачету с оценкой (при наличии)

1. Чем отличается архитектура решения от архитектуры программного обеспечения в целом?
2. Назовите основные артефакты, создаваемые на разных этапах жизненного цикла архитектуры ПО.
3. Объясните на примере, как применение принципа открытости/закрытости (OCP) помогает создавать абстракции, устойчивые к изменениям требований.
4. В чём разница между шаблонами «Адаптер» и «Мост»? Когда предпочтительно использовать каждый из них?
5. Опишите, как работает стратегия внедрения зависимостей через IoC-контейнер. Какие способы разрешения зависимостей вы знаете?
6. Какие проблемы возникают при вертикальном масштабировании приложения с интенсивной синхронизацией потоков?
7. Как системы обмена сообщениями (брокеры) помогают решать проблему интеграции в микросервисной архитектуре?
8. Что такое агрегат в терминах Domain-Driven Design? Приведите пример.
9. Назовите два способа реализации итератора для обхода составного объекта. В чём их отличие?
10. В каком случае цепочка обязанностей (Chain of Responsibility) предпочтительнее, чем использование условных операторов (if-else)?

Методические материалы, характеризующие процедуры оценивания

Промежуточная аттестация проводится в форме устного экзамена: каждый билет включает два теоретических вопроса и одно практическое задание (анализ или проектирование архитектуры). На подготовку отводится до 30 минут, после чего студент устно отвечает на вопросы билета, при необходимости преподаватель задаёт уточняющие вопросы. Оценка выставляется с учётом полноты ответа, корректности решения практической задачи, владения терминологией и аргументации; семестровый рейтинг может учитываться при спорной оценке.

Максимальная сумма баллов, набираемая студентом по дисциплине равна 100.

Оценка в баллах	Оценка по шкале	Обоснование	***Уровень сформированности компетенций***
Более 80	«Отлично»	Содержание курса освоено полностью, без пробелов, необходимые практические навыки работы с освоенным материалом сформированы, все предусмотренные программой обучения учебные задания выполнены, качество их выполнения оценено числом баллов, близким к максимальному	***Высокий уровень***
66-80	«Хорошо»	Содержание курса освоено полностью, без пробелов, некоторые практические навыки работы с освоенным материалом	***Продвинутый уровень***

		сформированы недостаточно, все предусмотренные программой обучения учебные задания выполнены, качество выполнения ни одного из них не оценено минимальным числом баллов, некоторые виды заданий выполнены с ошибками	
50-65	«Удовлетворительно»	Содержание курса освоено частично, но пробелы не носят существенного характера, необходимые практические навыки работы с освоенным материалом в основном сформированы, большинство предусмотренных программой обучения учебных заданий выполнено, некоторые из выполненных заданий, возможно, содержат ошибки	***Пороговый уровень***
Менее 50	«Неудовлетворительно»	Содержание курса не освоено, необходимые практические навыки работы не сформированы, выполненные учебные задания содержат грубые ошибки	***Компетенции не сформированы***

3. Задания в тестовой форме по дисциплине

Примеры заданий:
ОПК-1.1
ОПК-4.1
ОПК-7.1

Вопрос 1

Какая из перечисленных стратеций разрешения зависимостей в IoC предполагает внедрение зависимости через аргумент конструктора?

- Внедрение через свойство
- Внедрение через метод
- Внедрение через конструктор
- Внедрение через сервис-локатор

Вопрос 2

Какой принцип SOLID требует, чтобы абстракции не зависели от деталей, а детали зависели от абстракций?

- Принцип единственной ответственности
- Принцип открытости/закрытости
- Принцип подстановки Барбары Лисков
- Принцип инверсии зависимостей

Вопрос 3

Для какой цели в архитектуре приложений применяется шаблон «Адаптер»?

- Для разделения абстракции и реализации
- Для преобразования интерфейса одного класса в интерфейс, ожидаемый клиентом
- Для организации очереди сообщений между микросервисами
- Для последовательного обхода элементов коллекции без раскрытия её внутренней структуры

Вопрос 4

Что из перечисленного является характерным признаком событийно-ориентированной архитектуры (EDA) на основе систем обмена сообщениями?

- Синхронный вызов каждого сервиса с блокировкой потока
- Прямое соединение клиент-сервер через HTTP без посредников
- Асинхронная публикация событий в брокер и подписка заинтересованных сервисов
- Общая база данных для всех сервисов

Вопрос 5

В контексте Domain-Driven Design (DDD) что описывает термин «Антикоррупционный слой»?

- Слой, который предотвращает потерю данных при отказе сети
- Слой, который изолирует доменную модель от внешних систем, преобразуя чужие модели в свои
- Слой, который шифрует все сообщения между сервисами
- Слой, который агрегирует события из нескольких источников

Вопрос 6

Какая проблема решается использованием шаблона «Мост» (Bridge)?

- Необходимость создания большого числа подклассов при комбинации нескольких независимых измерений
- Управление транзакциями в распределённой системе
- Отложенная инициализация тяжёлого объекта
- Обеспечение единственного экземпляра класса в системе

Вопрос 7

При переходе от монолита к микросервисной архитектуре какой из перечисленных аспектов обычно становится наиболее сложным?

- Управление состоянием сессии и распределёнными транзакциями
- Написание кода на разных языках программирования
- Выбор интегрированной среды разработки
- Тестирование отдельного класса

Вопрос 8

Какое требование необходимо выполнить, чтобы правильно реализовать шаблон «Итератор»?

- Все элементы коллекции должны быть одного конкретного типа
- Коллекция должна быть неизменяемой в процессе итерации
- Предоставить механизм последовательного доступа к элементам без раскрытия внутреннего представления
- Гарантировать уникальность каждого элемента в коллекции

Вопрос 9

Что из перечисленного относится к недостаткам архитектуры, построенной исключительно на синхронных HTTP-вызовах (без обмена сообщениями)?

- Отсутствие поддержки SSL/TLS
- Трудность реализации циклических зависимостей между сервисами
- Жёсткая связанность, проблемы с масштабированием и низкая отказоустойчивость
- Невозможность передавать бинарные данные

Вопрос 10

В каком случае применение вертикального масштабирования (scale-up) предпочтительнее горизонтального (scale-out)?

- Когда требуется абсолютная отказоустойчивость на уровне дата-центров
- Когда приложение содержит глобальную блокирующую синхронизацию, не выносимую на несколько узлов
- Когда нагрузка носит случайный и резко переменный характер
- Когда бюджет на оборудование минимален

Полный перечень тестовых заданий с указанием правильных ответов, размещен в банке вопросов на информационно-образовательном портале института по ссылке https://www.mivlgu.ru/iop/course/view.php?id=4568

Оценка рассчитывается как процент правильно выполненных тестовых заданий из их общего числа.