

МИНОБРНАУКИ РОССИИ

**Федеральное государственное автономное образовательное
учреждение высшего образования "Пермский
государственный национальный исследовательский
университет"**

Институт компьютерных наук и технологий

Авторы-составители: **Огурецкая Анна Игоревна**

Рабочая программа дисциплины

**ТРЕК "ИНЖЕНЕРИЯ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ (АРХИТЕКТУРА И
ПРОЕКТИРОВАНИЕ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ)"**

Код УМК 100580

Утверждено
Протокол №1
от «28» июня 2024 г.

Пермь, 2024

1. Наименование дисциплины

Трек "Инженерия программного обеспечения (Архитектура и проектирование программного обеспечения)"

2. Место дисциплины в структуре образовательной программы

Дисциплина входит в вариативную часть Блока « Б.1 » образовательной программы по направлениям подготовки (специальностям):

Направление подготовки: **01.03.02** Прикладная математика и информатика
направленность Инженерия программного обеспечения

3. Планируемые результаты обучения по дисциплине

В результате освоения дисциплины **Трек "Инженерия программного обеспечения (Архитектура и проектирование программного обеспечения)"** у обучающегося должны быть сформированы следующие компетенции:

01.03.02 Прикладная математика и информатика (направленность : Инженерия программного обеспечения)

УК.1 Способен осуществлять поиск, анализ и синтез информации, применять системный подход для разрешения проблемных ситуаций

Индикаторы

УК.1.2 Работает с противоречивой информацией из разных источников, находит пробелы в необходимой для разрешения проблемы информации, определяет варианты устранения пробелов

ПК.5 Способен разрабатывать требования и проектировать программное обеспечение

Индикаторы

ПК.5.2 Разрабатывает, изменяет архитектуру компьютерного программного обеспечения; проектирует структуры данных, базы данных, алгоритмы, программные интерфейсы

4. Объем и содержание дисциплины

Направление подготовки	01.03.02 Прикладная математика и информатика (направленность: Инженерия программного обеспечения)
форма обучения	очная
№№ семестров, выделенных для изучения дисциплины	5
Объем дисциплины (з.е.)	4
Объем дисциплины (ак.час.)	144
Контактная работа с преподавателем (ак.час.), в том числе:	52
Проведение лекционных занятий	18
Проведение практических занятий, семинаров	34
Самостоятельная работа (ак.час.)	92
Формы текущего контроля	Итоговое контрольное мероприятие (1) Письменное контрольное мероприятие (2)
Формы промежуточной аттестации	Экзамен (5 семестр)

5. Аннотированное описание содержания разделов и тем дисциплины

Усовершенствование программного кода

Переменные

Общие принципы использования переменных. Грамотное объявление переменных. Области видимости. Инициализация переменных. Единственность цели каждой переменной. Конвенции именования переменных.

Основные типы данных

Целые числа. Числа с плавающей запятой. Символы и строки. Логические переменные. Перечисляемые типы. Именованные константы. Массивы. Структуры. Указатели. Глобальные данные.

Форматирование и стиль

Самодокументирующийся код

Качество ПО

Методики повышения качества ПО. Совместное конструирование. Тестирование, выполняемое разработчиками. Отладка. Стратегии оптимизации кода. Методики оптимизации кода.

Принципы рефакторинга и проектирование

Дублирование кода. Длинный метод. Большой класс. Длинный список параметров. Расходящиеся модификации. Параллельные иерархии наследования. Ленивый класс. Выделение метода. Выделение поясняющей переменной. Замещение алгоритма. Перемещение метода. Выделение класса. Удаление посредника. Организация данных. Замена массива объектом. Замена значения ссылкой. Инкапсуляция поля. Замена записи классом данных. Упрощение условных выражений. Удаление управляющего флага. Упрощение вызовов метода.

Разработка юнит-тестов

Ценность самотестирующегося кода. Фреймворки тестирования unittest, JUnit, NUnit.

Паттерны проектирования для ООП в схеме MVC

Порождающие паттерны

Паттерн "Строитель". Паттерн "Фабрика". Паттерн "Одиночка". Паттерн "Шаблонный метод".

Структурные паттерны

Паттерны "Адаптер" и "Фасад". Паттерн "Декоратор". Паттерн "Мост". Паттерны "Итератор" и "Компоновщик".

Паттерны поведения

Паттерн "Цепочка обязанностей". Паттерн "Наблюдатель". Паттерн "Интерпретатор". Паттерн "Команда". Паттерн "Состояние". Паттерн "Заместитель". Паттерн "Приспособленец". Паттерн "Посредник". Паттерн "Хранитель".

Проектирование микросервисной архитектуры

Микросервисы и их моделирование

Основные понятия микросервисной архитектуры. Основные преимущества микросервисов: устойчивость, масштабирование, простота развёртывания, технологическая разнородность. Слабая связанность и сильное зацепление микросервисов.

Интеграция микросервисов

Сохранение технологической независимости применяемых API. Совместно используемая база данных. Сравнение синхронного и асинхронного стилей. Удаленные вызовы процедуры. REST и HTTP. JSON и XML. DRY и риски повторного использования кода. Управление версиями микросервисов. Пользовательские интерфейсы. Интеграция с программами сторонних разработчиков.

Разбиение монолитного ПО на части

Мотивы для разбиения монолита на части. Запутанные зависимости. Перенастройка баз данных. Создание отчетов.

Развёртывание

Сборочные конвейеры и непрерывная доставка. Настраиваемые образы. Конфигурация сервиса. Docker. Vagrant.

Мониторинг и безопасность

Размещение одного сервиса на нескольких серверах. Журналирование. Отслеживание показателей нескольких сервисов. Каскадные сбои. Реализация семантического мониторинга. Аутентификация и авторизация. Технологии единого входа. Стандарт HTTP(S) Basic Authentication. Использование SAML или OpenID Connect. HMAC через HTTP. API-ключи.

6. Методические указания для обучающихся по освоению дисциплины

Освоение дисциплины требует систематического изучения всех тем в той последовательности, в какой они указаны в рабочей программе.

Основными видами учебной работы являются аудиторные занятия. Их цель - расширить базовые знания обучающихся по осваиваемой дисциплине и систему теоретических ориентиров для последующего более глубокого освоения программного материала в ходе самостоятельной работы. Обучающемуся важно помнить, что контактная работа с преподавателем эффективно помогает ему овладеть программным материалом благодаря расстановке необходимых акцентов и удержанию внимания интонационными модуляциями голоса, а также подключением аудио-визуального механизма восприятия информации.

Самостоятельная работа преследует следующие цели:

- закрепление и совершенствование теоретических знаний, полученных на лекционных занятиях;
- формирование навыков подготовки текстовой составляющей информации учебного и научного назначения для размещения в различных информационных системах;
- совершенствование навыков поиска научных публикаций и образовательных ресурсов, размещенных в сети Интернет;
- самоконтроль освоения программного материала.

Обучающемуся необходимо помнить, что результаты самостоятельной работы контролируются преподавателем во время проведения мероприятий текущего контроля и учитываются при промежуточной аттестации.

Обучающимся с ОВЗ и инвалидов предоставляется возможность выбора форм проведения мероприятий текущего контроля, альтернативных формам, предусмотренным рабочей программой дисциплины. Предусматривается возможность увеличения в пределах 1 академического часа времени, отводимого на выполнение контрольных мероприятий.

Процедура оценивания результатов обучения инвалидов и лиц с ограниченными возможностями здоровья по дисциплине предусматривает предоставление информации в формах, адаптированных к ограничениям их здоровья и восприятия информации.

При проведении текущего контроля применяются оценочные средства, обеспечивающие передачу информации, от обучающегося к преподавателю, с учетом психофизиологических особенностей здоровья обучающихся.

7. Перечень учебно-методического обеспечения для самостоятельной работы обучающихся по дисциплине

При самостоятельной работе обучающимся следует использовать:

- конспекты лекций;
- литературу из перечня основной и дополнительной учебной литературы, необходимой для освоения дисциплины (модуля);
- текст лекций на электронных носителях;
- ресурсы информационно-телекоммуникационной сети "Интернет", необходимые для освоения дисциплины;
- лицензионное и свободно распространяемое программное обеспечение из перечня информационных технологий, используемых при осуществлении образовательного процесса по дисциплине;
- методические указания для обучающихся по освоению дисциплины.

8. Перечень основной и дополнительной учебной литературы

Основная:

1. Павлова, Е. А. Технологии разработки современных информационных систем на платформе Microsoft.NET : учебное пособие / Е. А. Павлова. — 4-е изд. — Москва : Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2024. — 128 с. — ISBN 978-5-4497-2463-2. — Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. <https://www.iprbookshop.ru/133986.html>
2. Вязовик, Н. А. Программирование на Java : учебное пособие / Н. А. Вязовик. — 3-е изд. — Москва : Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2021. — 601 с. — ISBN 978-5-4497-0852-6. — Текст : электронный // Электронно-библиотечная система IPR BOOKS : [сайт]. <http://www.iprbookshop.ru/102048>

Дополнительная:

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. — ISBN 2227-8397. — Текст : электронный // Электронно-библиотечная система IPR BOOKS : [сайт]. <http://www.iprbookshop.ru/74412.html>

9. Перечень ресурсов сети Интернет, необходимых для освоения дисциплины

<https://refactoring.guru/ru/design-patterns> Паттерны/шаблоны проектирования

<https://habr.com/ru/post/136766/> Шпаргалка по паттернам проектирования

10. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине

Образовательный процесс по дисциплине **Трек "Инженерия программного обеспечения (Архитектура и проектирование программного обеспечения)"** предполагает использование следующего программного обеспечения и информационных справочных систем:

- 1) презентационные материалы (слайды по темам лекционных и практических занятий);
 - 2) доступ в режиме on-line в Электронную библиотечную систему (ЭБС);
 - 3) доступ в электронную информационно-образовательную среду университета;
 - 4) интернет-сервисы и электронные ресурсы (поисковые системы, электронная почта);
- Перечень необходимого лицензионного и (или) свободно распространяемого программного обеспечения

Среда разработки Visual Studio Code;

Платформа для визуальной совместной работы Miro;

Интерпретатор языка программирования Python 3.9;

Платформа Java, Standard Edition Development Kit 11;

Платформа .NET 5;

Операционная система ALT Linux;

Офисный пакет приложений «LibreOffice».

При освоении материала и выполнения заданий по дисциплине рекомендуется использование материалов, размещенных в Личных кабинетах обучающихся ЕТИС ПГНИУ (**student.psu.ru**).

При организации дистанционной работы и проведении занятий в режиме онлайн могут использоваться:

система видеоконференцсвязи на основе платформы BigBlueButton (<https://bigbluebutton.org/>).

система LMS Moodle (<http://e-learn.psu.ru/>), которая поддерживает возможность использования текстовых материалов и презентаций, аудио- и видеоконтент, а так же тесты, проверяемые задания, задания для совместной работы.

система тестирования Indigo (<https://indigotech.ru/>).

11. Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине

Для проведения лекционных занятий:

Аудитория, оснащенная презентационной техникой (проектор, экран, компьютер/ноутбук) с соответствующим программным обеспечением, меловой (и) или маркерной доской.

Для проведения лабораторных занятий – Компьютерный класс, оснащенный персональными ЭВМ и соответствующим программным обеспечением. Состав оборудования определен в Паспорте Компьютерного класса.

Аудитории для проведения текущего контроля;

Компьютерный класс, оснащенный персональными ЭВМ и соответствующим программным обеспечением. Состав оборудования определен в Паспорте компьютерного класса.

Аудитории для групповых (индивидуальных) консультаций;

Аудитория, оснащенная презентационной техникой (проектор, экран, компьютер/ноутбук) с соответствующим программным обеспечением, меловой (и) или маркерной доской.

Аудитория для самостоятельной работы:

Аудитория оснащенная компьютерной техникой с возможностью подключения к сети «Интернет», обеспеченная доступом в электронную информационно-образовательную среду университета.

Помещения Научной библиотеки ПГНИУ

Помещения научной библиотеки ПГНИУ для обеспечения самостоятельной работы обучающихся:

1. Научно-библиографический отдел, корп.1, ауд. 142. Оборудован 3 персональными компьютера с доступом к локальной и глобальной компьютерным сетям.

2. Читальный зал гуманитарной литературы, корп. 2, ауд. 418. Оборудован 7 персональными компьютерами с доступом к локальной и глобальной компьютерным сетям.

3. Читальный зал естественной литературы, корп.6, ауд. 107а. Оборудован 5 персональными компьютерами с доступом к локальной и глобальной компьютерным сетям.

4. Отдел иностранной литературы, корп.2 ауд. 207. Оборудован 1 персональным компьютером с доступом к локальной и глобальной компьютерным сетям.

5. Библиотека юридического факультета, корп.9, ауд. 4. Оборудована 11 персональными компьютерами с доступом к локальной и глобальной компьютерным сетям.

6. Читальный зал географического факультета, корп.8, ауд. 419. Оборудован 6 персональными компьютерами с доступом к локальной и глобальной компьютерным сетям.

Все компьютеры, установленные в помещениях научной библиотеки, оснащены следующим программным обеспечением:

Операционная система ALT Linux;

Офисный пакет Libreoffice.

Справочно-правовая система «КонсультантПлюс»

Фонды оценочных средств для аттестации по дисциплине
Трек "Инженерия программного обеспечения (Архитектура и проектирование
программного обеспечения)"

Планируемые результаты обучения по дисциплине для формирования компетенции.
Индикаторы и критерии их оценивания

ПК.5

Способен разрабатывать требования и проектировать программное обеспечение

Компетенция (индикатор)	Планируемые результаты обучения	Критерии оценивания результатов обучения
ПК.5.2 Разрабатывает, изменяет архитектуру компьютерного программного обеспечения; проектирует структуры данных, базы данных, алгоритмы, программные интерфейсы	Способен разрабатывать и изменять архитектуру компьютерного программного обеспечения, проектировать структуры данных, базы данных, алгоритмы и программные интерфейсы с использованием современных методов и инструментов. Умеет анализировать процессы взаимодействия компонентов программного обеспечения, оптимизировать их работу, создавать документацию на разработанное программное обеспечение и обучать других разработчиков его использованию.	Неудовлетворител Студент не способен использовать современные методы и инструменты для проектирования элементов программного обеспечения, или испытывает значительные затруднения в процессе работы. Может изменить существующую архитектуру программного обеспечения только с направляющим руководством. Плохо разбирается в процессах взаимодействия компонентов программного обеспечения. Студент может создавать документацию на разработанное программное обеспечение, но допускает неточности. Архитектура программного обеспечения разработана и реализована не качественно или не соответствует требованиям потенциального заказчика. В процессе работы возможны сложности, которые студент не способен устранить самостоятельно. Удовлетворительн Студент способен использовать современные методы и инструменты для проектирования элементов программного обеспечения, но испытывает некоторые сложности в процессе работы. Может изменить существующую архитектуру программного обеспечения с целью повышения его эффективности и надежности, но нуждается в направляющем руководстве. Умеет анализировать процессы взаимодействия компонентов программного обеспечения и оптимизировать их работу.

Компетенция (индикатор)	Планируемые результаты обучения	Критерии оценивания результатов обучения
		Удовлетворительн Студент может создавать документацию на разработанное программное обеспечение. Архитектура программного обеспечения разработана и реализована качественно, соответствует требованиям потенциального заказчика. В процессе работы возможны некоторые сложности, которые студент способен устранить самостоятельно. Хорошо Студент использует современные методы и инструменты для проектирования элементов программного обеспечения. Может изменить существующую архитектуру программного обеспечения с целью повышения его эффективности и надежности. Умеет анализировать процессы взаимодействия компонентов программного обеспечения и оптимизировать их работу. Студент может создавать документацию на разработанное программное обеспечение и обучать других разработчиков его использованию. Архитектура программного обеспечения разработана и реализована качественно, соответствует требованиям потенциального заказчика и современным технологиям. В процессе работы возможны некоторые сложности, которые студент способен устранить самостоятельно. Отлично Студент эффективно использует современные методы и инструменты для проектирования элементов программного обеспечения. Может самостоятельно изменить существующую архитектуру программного обеспечения с целью повышения его эффективности и надежности. Умеет анализировать процессы взаимодействия компонентов программного обеспечения и оптимизировать их работу. Студент может самостоятельно создавать документацию на разработанное

Компетенция (индикатор)	Планируемые результаты обучения	Критерии оценивания результатов обучения
		Отлично программное обеспечение и обучать других разработчиков его использованию. Архитектура программного обеспечения разработана и реализована качественно, соответствует требованиям потенциального заказчика и современным технологиям.

УК.1

Способен осуществлять поиск, анализ и синтез информации, применять системный подход для разрешения проблемных ситуаций

Компетенция (индикатор)	Планируемые результаты обучения	Критерии оценивания результатов обучения
УК.1.2 Работает с противоречивой информацией из разных источников, находит пробелы в необходимой для разрешения проблемы информации, определяет варианты устранения пробелов	знает способы работы с противоречивой информацией в области информационных технологий; умеет работать с противоречивой информацией из разных источников при анализе и находить пробелы в информации в области информационных технологий, определять варианты устранения этих пробелов	Неудовлетворител не знает способы работы с противоречивой информацией в области информационных технологий; не умеет работать с противоречивой информацией из разных источников при анализе и находить пробелы в информации в области информационных технологий, определять варианты устранения этих пробелов Удовлетворительн показывает общие, но не структурированные знания о способах работы с противоречивой информацией в области информационных технологий; демонстрирует частично сформированное умение работать с противоречивой информацией из разных источников при анализе и показывает фрагментарное применение навыков находить пробелы в информации в области информационных технологий, определять варианты устранения этих пробелов Хорошо показывает сформированные, но содержащие отдельные недочеты знания способов работы с противоречивой информацией в области информационных технологий; демонстрирует в целом успешные, но содержащие отдельные пробелы умения работать с противоречивой информацией из разных

Компетенция (индикатор)	Планируемые результаты обучения	Критерии оценивания результатов обучения
		Хорошо источников при анализе и навыков находить пробелы в информации в области информационных технологий, определять варианты устранения этих пробелов Отлично показывает сформированные систематические знания способов работы с противоречивой информацией в области информационных технологий; демонстрирует сформированное умение работать с противоречивой информацией из разных источников при анализе и успешное систематическое применение навыков находить пробелы в информации в области информационных технологий, определять варианты устранения этих пробелов

Оценочные средства текущего контроля и промежуточной аттестации

Схема доставки : Базовая

Вид мероприятия промежуточной аттестации : Экзамен

Способ проведения мероприятия промежуточной аттестации : Оценка по дисциплине в рамках промежуточной аттестации определяется на основе баллов, набранных обучающимся на контрольных мероприятиях, проводимых в течение учебного периода.

Максимальное количество баллов : 100

Конвертация баллов в отметки

«отлично» - от 81 до 100

«хорошо» - от 61 до 80

«удовлетворительно» - от 43 до 60

«неудовлетворительно» / «незачтено» менее 43 балла

Компетенция (индикатор)	Мероприятие текущего контроля	Контролируемые элементы результатов обучения
УК.1.2 Работает с противоречивой информацией из разных источников, находит пробелы в необходимой для разрешения проблемы информации, определяет варианты устранения пробелов ПК.5.2 Разрабатывает, изменяет архитектуру компьютерного программного обеспечения; проектирует структуры данных, базы данных, алгоритмы, программные интерфейсы	Разработка юнит-тестов Письменное контрольное мероприятие	Способен проводить рефакторинг программного кода и поддерживать качество ПО за счет использования системы unit-тестов.

Компетенция (индикатор)	Мероприятие текущего контроля	Контролируемые элементы результатов обучения
УК.1.2 Работает с противоречивой информацией из разных источников, находит пробелы в необходимой для разрешения проблемы информации, определяет варианты устранения пробелов ПК.5.2 Разрабатывает, изменяет архитектуру компьютерного программного обеспечения; проектирует структуры данных, базы данных, алгоритмы, программные интерфейсы	Паттерны поведения Письменное контрольное мероприятие	Способен применять базовые паттерны проектирования ПО при разработке приложения на объектно-ориентированном языке программирования (C#, Java, Python).
УК.1.2 Работает с противоречивой информацией из разных источников, находит пробелы в необходимой для разрешения проблемы информации, определяет варианты устранения пробелов ПК.5.2 Разрабатывает, изменяет архитектуру компьютерного программного обеспечения; проектирует структуры данных, базы данных, алгоритмы, программные интерфейсы	Мониторинг и безопасность Итоговое контрольное мероприятие	Способен провести рефакторинг монолитной архитектуры в микросервисная архитектуру.

Спецификация мероприятий текущего контроля

Разработка юнит-тестов

Продолжительность проведения мероприятия промежуточной аттестации: **2 часа**

Условия проведения мероприятия: **в часы самостоятельной работы**

Максимальный балл, выставляемый за мероприятие промежуточной аттестации: **30**

Проходной балл: **13**

Показатели оценивания	Баллы
В ходе рефакторига предлагает корректные способы выделения новых сущностей (классов, методов, переменных).	6
Определяет классы, методы, переменные, повторяющиеся части кода, для которых необходимо произвести рефакторинг.	6
Соблюдает формальные и неформальные (соглашения) правила для форматирования и	

стиля программного кода. Способен писать самодокументируемый код.	6
Разрабатывает систему unit-тестов, необходимую для минимального покрытия функционала приложения.	6
Проводит мониторинг качества ПО на основе разработанной системы unit-тестов.	6

Паттерны поведения

Продолжительность проведения мероприятия промежуточной аттестации: **2 часа**

Условия проведения мероприятия: **в часы самостоятельной работы**

Максимальный балл, выставаемый за мероприятие промежуточной аттестации: **30**

Проходной балл: **13**

Показатели оценивания	Баллы
Демонстрирует использование не менее одного паттерна поведения при проектировании приложения.	10
Демонстрирует использование не менее одного структурного паттерна при проектировании приложения.	10
Демонстрирует использование не менее одного порождающего паттерны при проектировании приложения.	10

Мониторинг и безопасность

Продолжительность проведения мероприятия промежуточной аттестации: **2 часа**

Условия проведения мероприятия: **в часы самостоятельной работы**

Максимальный балл, выставаемый за мероприятие промежуточной аттестации: **40**

Проходной балл: **17**

Показатели оценивания	Баллы
В ходе рефакторинга монолитной архитектуры способен выделить базовые сущности микросервисов.	10
Способен разработать RESTful API для микросервисов.	10
Способен организовать мониторинг работы микросервисов.	10
Способен автоматизировать непрерывное развертывание и доставку микросервисов в docker-контейнерах.	10