



## **1. Цель и планируемые результаты освоения дисциплины**

Целью освоения дисциплины является формирование следующих компетенций:

- ПК-1 – способность проектировать программное обеспечение;
- ПК-2 – способность архитектурный проект программного средства;
- ПК-8 – способность руководить проектами по созданию комплексных систем

искусственного интеллекта

Результатами освоения дисциплины являются следующие индикаторы достижения компетенций:

ИПК-2.3. Определяет ключевые сценарии для архитектуры программного средства.

ИПК-2.2. Определяет цели архитектуры программного средства.

ИПК-2.1. Оценивает возможность создания архитектурного проекта программного средства.

ИПК-1.3. Знает методы и средства проектирования программного обеспечения, методы и средства проектирования баз данных.

ИПК-1.2. Применяет методы и средства проектирования программного обеспечения, структур данных, баз данных, программных интерфейсов.

ИПК-1.1. Использует существующие типовые решения и шаблоны проектирования программного обеспечения.

ИПК-8.1. Руководит разработкой архитектуры комплексных систем искусственного интеллекта

## **2. Задачи освоения дисциплины**

Изучение основ архитектурного проектирования, изучение подходов к разработке сложных программных систем, методов организации устойчивого процесса разработки.

- Изучить основные понятия и процессы, связанные с разработкой архитектур программных систем;
- Освоить инструменты разработки программных систем;
- Научиться применять типовые шаблоны архитектурного проектирования;

## **3. Место дисциплины в структуре образовательной программы**

Дисциплина относится к части образовательной программы, формируемой участниками образовательных отношений, предлагается обучающимся на выбор. Дисциплина входит в модуль Специализация.

## **4. Семестр(ы) освоения и форма(ы) промежуточной аттестации по дисциплине**

Второй семестр, зачет

## **5. Входные требования для освоения дисциплины**

Для успешного освоения дисциплины требуются результаты обучения по следующим дисциплинам: «Введение в программную инженерию» .

## **6. Язык реализации**

Русский

## **7. Объем дисциплины**

Общая трудоемкость дисциплины составляет 2 з.е., 72 часов, из которых:

-лекции: 16 ч.

-лабораторные: 16 ч.

в том числе практическая подготовка: 0 ч.

Объем самостоятельной работы студента определен учебным планом.

## **8. Содержание дисциплины, структурированное по темам**

Тема 1. Введение в архитектуру программных систем.

Понятие архитектуры программной системы. Требования к квалификации архитектора программного обеспечения (software architect).

Тема 2. Практика архитектурного проектирования.

Критерии качества кода. Повышение культуры разработки. Архитектура и объектно-ориентированное проектирование (ООП). Абстракции как метод борьбы со сложностью. Архитектурные паттерны. Шаблоны проектирования SOLID. Шаблоны управления зависимостями в коде. Внедрение зависимостей, принцип инверсии контроля. Разбор типовых архитектур и примеров.

Тема 3. Управленческие аспекты разработки архитектуры.

Организация процесса разработки программного обеспечения, методологии разработки. Место архитектора в этих процессах. Управление техническим долгом.

## **9. Текущий контроль по дисциплине**

Текущий контроль по дисциплине проводится путем контроля посещаемости, проведения контрольных работ, проверки лабораторных работ. По итогу лабораторной работы студентом оформляется отчет и защищается перед преподавателем. По результатам выполнения работ и итоговой защиты, предоставляется аттестация.

## **10. Порядок проведения и критерии оценивания промежуточной аттестации**

Промежуточная аттестация проводится в форме зачета во втором семестре. Студент получает зачет в случае, если он сдал все лабораторные работы, проверяющие компетенции ИПК-2.1, ИПК-2.2 и ИПК-2.3 и прошел итоговое тестирование, проверяющее компетенции ИПК-1.1, ИПК-1.2, ИПК-1.3, ИПК-8.1.

Примерный перечень теоретических вопросов:

1. Смысл и практическое применение паттернов SOLID.
2. Критерии качества кода.
3. Именованые сущностей.
4. Принцип DRY.
5. Зацепление и связность.
6. Внедрение зависимостей (через конструктор, свойство, параметр метода).
7. Как разделять код на уровни абстракции.
8. Для чего нужны ревью кода и как они организуются в команде.
9. Различие функциональных и нефункциональных требований.
10. Понятие технического долга и основные подходы к управлению им.

## **11. Учебно-методическое обеспечение**

а) Электронный учебный курс по дисциплине в электронном университете «Moodle».

б) Оценочные материалы текущего контроля и промежуточной аттестации по дисциплине.

## **12. Перечень учебной литературы и ресурсов сети Интернет**

а) основная литература:

- Мартин, Роберт Чистый код: создание, анализ и рефакторинг / Роберт Мартин. — М. : Питер, 2018. — 464 с.
- Мартин, Роберт Чистая архитектура. Искусство разработки программного обеспечения / Роберт Мартин. — М. : Питер, 2018. — 352 с.
- Макконнелл, Стив Совершенный код / Стив Макконнелл. — М. : БХВ-Петербург, 2017. — 896 с.
- Fairbanks, G. H. Just Enough Software Architecture: A Risk-Driven Approach / G. H. Fairbanks. — 1. — Marshall & Brainerd, 2010. — 376 с.
- Хант, Эндрю Программист-прагматик / Эндрю Хант, Дэвид Томас. — М. : Вильямс, 2020. — 368 с.
- Симан, Марк Внедрение зависимостей на платформе .NET / Марк Симан, Стивен ван Дерсен. — М. : Питер, 2021. — 608 с.

### **13. Перечень информационных технологий**

- а) лицензионное и свободно распространяемое программное обеспечение:  
 – MS Windows или Linux; IDE JetBrains Rider

### **14. Материально-техническое обеспечение**

Аудитории для проведения занятий лекционного типа.

Аудитории для проведения лабораторных занятий, индивидуальных и групповых консультаций, текущего контроля и промежуточной аттестации.

Помещения для самостоятельной работы, оснащенные компьютерной техникой и доступом к сети Интернет, в электронную информационно-образовательную среду и к информационным справочным системам.

### **15. Информация о разработчиках**

Шестаков Николай Александрович – канд. техн. наук, доцент кафедры теоретических основ информатики ТГУ.