

АННОТАЦИЯ

рабочей программы дисциплины «Объектно-ориентированное проектирование и программирование»

по направлению подготовки 09.03.03 Прикладная информатика (уровень бакалавриата)

Направленность (профиль) образовательной программы:
Прикладная информатика в юриспруденции

Цель освоения дисциплины «Объектно-ориентированное проектирование и программирование» – формирование у обучающихся теоретических знаний и практических навыков в области объектно-ориентированного анализа, проектирования и программирования программного обеспечения, развитие компетенций, необходимых для руководства разработкой, интеграцией и сопровождением программных продуктов в условиях современных методологий разработки.

Место дисциплины в структуре образовательной программы бакалавриата

Дисциплина «Объектно-ориентированное проектирование и программирование» относится к части формируемой участниками образовательных отношений.

Освоение дисциплины предполагает наличие у обучающихся знаний, умений и навыков по дисциплинам: «Информационные технологии», «Профессиональные информационные системы и базы данных», «Теория вероятностей и математическая статистика».

Знания и навыки, полученные в результате освоения дисциплины «Объектно-ориентированное проектирование и программирование», являются базовыми для прохождения учебной и производственной практик, а также для изучения дисциплин: «Проектирование и разработка web сайтов», «Визуальное программирование на языке Python», «Визуальное программирование на языке Java».

Объем, структура и содержание дисциплины «Объектно-ориентированное проектирование и программирование» составляет 4 зачетных единиц, 144 академических часа.

Содержание дисциплины «Объектно-ориентированное проектирование и программирование»

Раздел 1. Введение в ООП и основы проектирования

Тема 1. Эволюция парадигм программирования. Принципы ООП. Эволюция парадигм программирования: процедурное, объектно-ориентированное, функциональное. Четыре основных принципа ООП. Преимущества ООП для разработки масштабируемых систем. Анализ предметной области: выявление сущностей, атрибутов, связей.

Тема 2. Анализ предметной области. Диаграммы прецедентов. Диаграммы прецедентов (Use Case): акторы, сценарии, отношения. Инструменты визуализации (StarUML, Draw.io). Практика построения диаграмм для типового приложения.

Раздел 2. Базовые концепции ООП

Тема 3. Классы и объекты. Инкапсуляция и абстракция. Класс как шаблон объекта. Поля, методы, конструкторы. Инкапсуляция: модификаторы доступа, геттеры/сеттеры. Абстракция: выделение существенных характеристик.

Тема 4. Наследование и полиморфизм. Наследование: базовые и производные классы, переопределение методов. Полиморфизм: статический и динамический, виртуальные методы.

Тема 5. Интерфейсы и абстрактные классы. Интерфейсы: контракты поведения. Абстрактные классы: частичная реализация. Сравнение интерфейсов и абстрактных классов.

Раздел 3. Проектирование классов

Тема 6. Диаграммы классов. Ассоциации, агрегация, композиция . Диаграммы классов: ассоциации, агрегация, композиция, зависимости. Кардинальность связей. Практика проектирования модели данных.

Тема 7. Принципы SOLID. Принципы SOLID: SRP, OCP, LSP, ISP, DIP. Примеры нарушений и их устранение.

Тема 8. Рефакторинг кода . Рефакторинг: техники (извлечение метода, замена условного оператора полиморфизмом), инструменты IDE.

Раздел 4. Продвинутое проектирование ООП

Тема 9. Шаблоны проектирования: порождающие.

Порождающие паттерны: Фабрика, Абстрактная фабрика, Строитель, Одиночка.

Тема 10. Шаблоны проектирования: структурные. Структурные паттерны: Адаптер, Декоратор, Фасад, Компоновщик.

Тема 11. Шаблоны проектирования: поведенческие. Поведенческие паттерны: Наблюдатель, Стратегия, Команда, Состояние.

Раздел 5. Инструменты и методологии

Тема 12. UML-моделирование. Диаграммы последовательности и состояний. Диаграммы последовательности: объекты, сообщения, активации. Диаграммы состояний: состояния, переходы, события.

Тема 13. Системы контроля версий (Git). Работа с репозиториями Git: основные команды, ветвление, слияние, разрешение конфликтов. Работа с GitHub/GitLab.

Тема 14. Введение в CI/CD. Автоматизация сборки и тестирования. Работа с репозиториями. Работа с репозиториями. CI/CD: концепция, инструменты (Jenkins, GitLab CI). Автоматизация тестирования и деплоя.

Раздел 6. Практическое применение

Тема 15. Разработка многослойного приложения (кейс). Разработка консольного/веб-приложения с применением ООП, паттернов, работы с репозиториями.

Тема 16. Подготовка технической документации. Составление технического задания, диаграмм классов, описания архитектуры. (Содержание для заочной формы обучения адаптировано с акцентом на самостоятельное изучение и подготовку рефератов (Р) по темам)