

	<p>– фактор закрытости метода; фактор закрытости свойства; фактор наследования метода; фактор наследования свойства; фактор полиморфизма; фактор сцепления.</p> <p>Синтаксис языка Java, отношения между классами и особенности разработки; особенности наследования; исключительные ситуации и их обработка; механизмы ввода и вывода информации; понятие сериализации.</p> <p>Основы синтаксиса Java. Типы данных и литералы. Операторы. Работа со строками и массивами. Инструкции. Отношения между классами. Объектно-ориентированный дизайн. Отношения между классами. Наследование. Зависимость. Ассоциация. Агрегация. Композиция. Метакласс. Характеристики ООП дизайна приложений. Связность (coupling) и сфокусированность (cohesion). Принципы дизайна. Принципы SOLID, YAGNI, KISS. Возникновение ошибок и подходы к их обработке. Механизм обработки исключений. Классификация исключений. Объявляемые исключения и их особенности. Отлов исключений. Выбрасывание исключений. Создание типов исключений. Отладка приложений. Виды наследования. Расширение классов. Переопределение методов. Соккрытие полей. Завершенные и абстрактные методы и классы. Интерфейсы и их описание. Использование интерфейсов. Потoki данных и их виды. Иерархия и разновидности потоков данных. Понятие сериализации. Особенности сериализации и десериализации.</p> <p>Многопоточное программирование: общие принципы и реализация. Основы ООП проектирования сетевых приложений.</p> <p>Многопоточное программирование и его особенности Потoki и работа с ними. Группы потоков. Приоритеты потоков. Демон-потоки. Блокировки и синхронизация. Новые виды ошибок. Совместная работа с полями и переменными. Методы класса Object. Прерывание потоков. Высокоуровневые средства работы с потоками. Протоколы транспортного уровня</p> <p>Понятие сокета. Абстракция сокета. Понятие порта. Пакет java.net. Классы Socket и ServerSocket. Классы DatagramPacket и DatagramSocket. Класс URL</p> <p><i>1.2. Основные понятия паттернов (шаблонов) проектирования. Порождающие паттерны. Структурные паттерны проектирования. Поведенческие паттерны проектирования.</i></p> <p>Понятия паттернов проектирования. Преимущества и недостатки паттернов проектирования. Проблема повторяемости программного кода. Классификация паттернов (порождающие, поведенческие, структурные). Особенности порождающих шаблонов проектирования ПО. Abstract Factory (Абстрактная фабрика). Builder (Строитель). Factory Method (Фабричный метод). Prototype (Прототип). Singleton (Одиночка).</p> <p>Основные особенности структурных паттернов проектирования. Состав и описание основных структурных шаблонов. Adapter (Адаптер). Bridge (Мост). Facade (Фасад). Composite (Композит). Decorator (Декоратор). Flyweight (Легковес). Proxy (Прокси).</p> <p>Основные особенности и назначение поведенческих шаблонов проектирования. Состав и описание основных поведенческих шаблонов. Chain Of Responsibility (Цепочка обязанностей). Command (Команда). State (Состояние). Template Method (Шаблонный метод). Mediator (Посредник). Interpreter (Интерпретатор). Iterator (Итератор). Memento (Хранитель). Observer (Наблюдатель). Strategy (Стратегия). Visitor (Посетитель).</p>
2	<p>лабораторные работы 4 шт. по 2 часа:</p> <p><i>2.1. Разработка многопоточных приложений.</i></p> <p>Теоретическое введение. Класс Thread. Интерфейс Runnable. Синхронизация потоков. Приоритет потока. Примеры программирования. Создание многопоточного приложения на основе базового класса Thread. Создание многопоточного приложения с помощью интерфейса Runnable.</p>

	<p>2.2. Оценка качества объектно-ориентированных программ с помощью метрик Мартина и Чидамбера -Кемерера. Оценка качества объектно-ориентированных программ с помощью метрик Лоренца-Кидда и Абреу. Пример объектно-ориентированного приложения. Оценка характеристик программы «Геометрия точки, окружности и прямоугольника» с помощью метрик Мартина и Чидамбера-Кемерера. Метрики Мартина. Метрики Чидамбера и Кемерера. Оценка характеристик программы «Геометрия точки, окружности и прямоугольника» с помощью метрик Лоренца - Кидда и Абреу. Метрики Лоренца и Кидда. Метрики Абреу. Задания для самостоятельной работы.</p> <p>2.3. Порождающие паттерны проектирования. Структурные паттерны проектирования. Примеры практического использования порождающих паттернов проектирования. Паттерн проектирования Abstract Factory. Паттерн проектирования «Прототип» Prototype. Задания для самостоятельной работы. Примеры практического использования структурных паттернов проектирования. Паттерн Adapter (Адаптер). Паттерн Decorator (Декоратор). Задания для самостоятельной работы.</p> <p>2.4. Поведенческие паттерны проектирования. Применение шаблонов проектирования на практике. Примеры практического использования поведенческих паттернов проектирования. Паттерны Command (Комманда), State (Состояние), Template Method (Шаблонный метод). Mediator (Посредник). Задания для самостоятельной работы. Разработка сетевых клиент-серверных приложений с использованием паттернов проектирования. Выполнить объектно-ориентированное тестирование разработанного приложения. Задания для самостоятельной работы.</p>
3	практические занятия не предусмотрены в структуре дисциплины
4	курсовая работа не предусмотрена в структуре дисциплины
5	расчетно-графическая работа не предусмотрена в структуре дисциплины
6	<p>Самостоятельная работа студентов:</p> <p>6.1. 2 контрольных опроса после 1-й и 2-й лекций;</p> <p>6.2. Закрепление материала по тематике лекционных занятий: закрепление изучения материалов лекций 1.1-1.4 – основы объектно-ориентированного программирования на языке высокого уровня; классификация программного обеспечения; проектирование программного обеспечения на основе паттернов и объектно-ориентированном подходе к программированию; тестирование и отладка программных продуктов; оценка качества разработанных программных средств при помощи метрик.</p> <p>6.3. Подготовка к зачету по дисциплине (оценочные материалы приведены в разделе 6 настоящей РПД).</p>

Год начала подготовки (по учебному плану) 2026

Образовательный стандарт (СУОС) утвержденный ректором ФГБОУ ВО «НИУ «МЭИ» Н.Д. Роголевым 20.12.2023