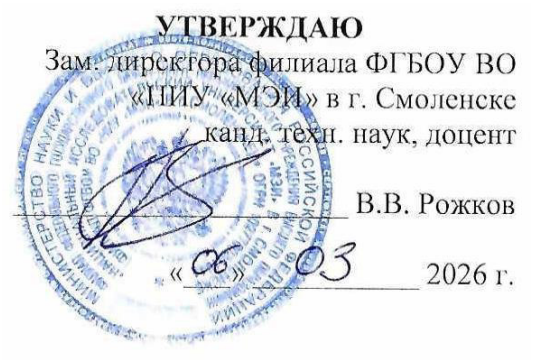
Филиал федерального государственного бюджетного образовательного учреждения высшего образования
«Национальный исследовательский университет «МЭИ»
в г. Смоленске



РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

ТЕХНОЛОГИИ ОБЪЕКТНОГО ПРОГРАММИРОВАНИЯ

(НАИМЕНОВАНИЕ ДИСЦИПЛИНЫ)

Направление подготовки: 09.03.01 Информатика и вычислительная техника

Профиль «Программное обеспечение средств вычислительной техники и автоматизированных систем»

Уровень высшего образования: бакалавриат

Нормативный срок обучения: 4 года 11 месяцев

Форма обучения: заочная

Год набора: 2026

Смоленск

Программа составлена с учетом ОС ВО – бакалавриата по направлению подготовки 09.03.01 «Информатика и вычислительная техника», утвержденного ректором ФГБОУ ВО «НИУ «МЭИ» Н.Д. Рогалевым 20.12.2023.

Программу составил:
к.т.н., доцент кафедры «Вычислительная техника»

Я.А. Федулов

«16» февраля 2026 г.

Программа обсуждена и одобрена на заседании кафедры «Вычислительная техника»
«18» февраля 2026 г., протокол № 5.

Заведующий кафедрой вычислительной техники
д.т.н., профессор

В.В. Борисов

« 05 » марта 2026 г.

РПД адаптирована для лиц с ограниченными возможностями здоровья и инвалидов

Ответственный в филиале по работе
с ЛОВЗ и инвалидами

Е.В. Зуева

« 05 » марта 2026 г.

1. ЦЕЛИ И ЗАДАЧИ ОСВОЕНИЯ ДИСЦИПЛИНЫ

Целью освоения дисциплины «Технологии объектного программирования» является формирование знаний, умений и навыков в определении требований к информационным системам и программному обеспечению и проектировании его на основе современных технологий программирования.

Задачами дисциплины являются: изучение основных понятий и определений, классификаций программного обеспечения; получение знаний об основных этапах создания программного обеспечения в рамках жизненного цикла, при современном состоянии технологий разработки программный продуктов; выработка умений в анализе исходную документации и разработки документации по информационным системам и программному обеспечению; получение навыков оценки качества процессов создания программного обеспечения; выработка практических навыков проектирования программного обеспечения и расчета его надежности.

2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОП ВО

Дисциплина «Технологии объектного программирования» относится *к части, формируемой участниками образовательных отношений* профессионального цикла Б1.В.15 основной образовательной программы подготовки бакалавров по направлению «09.03.01 «Информатика и вычислительная техника».

Перечень последующих дисциплин, для которых необходимы знания, умения и навыки, формируемые данной дисциплиной:

Б1.В.14 Тестирование программного обеспечения (ПК-8)
Б1.В.ДВ.05.01 Трансляторы программных языков (ПК-8)
Б1.В.ДВ.05.02 Теория формальных грамматик (ПК-8)

3. ТРЕБОВАНИЯ К РЕЗУЛЬТАТАМ ОСВОЕНИЯ ДИСЦИПЛИНЫ

Освоение дисциплины направлено на формирование элементов следующих компетенций в соответствии с ФГОС ВО и ОП ВО по данному направлению подготовки (*специальности*):

ПК-8 - Способен выполнять работы по проектированию, интеграции и тестированию компонентов системного программного обеспечения.

Перечень планируемых результатов обучения по дисциплине (модулю), соотнесенных с индикаторами достижения компетенций

Компетенция	Индикаторы достижения компетенций	Результаты обучения
ПК-8. Способен выполнять работы по проектированию, интеграции и тестированию компонентов системного программного обеспечения	*ПК-8.1. Выполняет работы по проектированию компонентов системного программного обеспечения*	*Знает: возможности информационных систем и баз данных; устройство и функционирование современных информационных систем; современные стандарты информационного взаимодействия систем; инструменты и методы анализа требований; характеристики различных систем обеспечения безопасности, влияющие на производительность информационных систем и баз данных.* *Умеет: анализировать исходную документацию и разрабатывать документацию по информационным системам и базам данных; настраивать параметры инструментов системы безопасности в соответствии с установленными критериями.* *Владеет: навыками анализа функциональных и нефункциональных требований к информационным системам и базам данных; навыками документирования и верификации требований к информационным системам и базам данных.*
	ПК-8.2. Выполняет работы по интеграции компонентов системного программного обеспечения	*Знает: возможности информационных систем и баз данных; устройство и функционирование современных информационных систем; современные стандарты информационного взаимодействия систем; инструменты и методы анализа требований; характеристики различных систем обеспечения безопасности, влияющие на производительность*

		информационных систем и баз данных. *Умеет: анализировать исходную документацию и разрабатывать документацию по информационным системам и базам данных; настраивать параметры инструментов системы безопасности в соответствии с установленными критериями.* *Владеет: навыками анализа функциональных и нефункциональных требований к информационным системам и базам данных; навыками документирования и верификации требований к информационным системам и базам данных.*
	ПК-8.3. Выполняет работы по тестированию компонентов системного программного обеспечения	*Знает: возможности информационных систем и баз данных; устройство и функционирование современных информационных систем; современные стандарты информационного взаимодействия систем; инструменты и методы анализа требований; характеристики различных систем обеспечения безопасности, влияющие на производительность информационных систем и баз данных.* *Умеет: анализировать исходную документацию и разрабатывать документацию по информационным системам и базам данных; настраивать параметры инструментов системы безопасности в соответствии с установленными критериями.* *Владеет: навыками анализа функциональных и нефункциональных требований к информационным системам и базам данных; навыками документирования и верификации требований к информационным системам и базам данных.*

4. СТРУКТУРА И СОДЕРЖАНИЕ ДИСЦИПЛИНЫ

№	Индекс	Наименование	Сессия 1									Сессия 2									Сессия 3									Итого за курс											Каф.	Курсы
			Контроль	Академических часов								Дней	Контроль	Академических часов								Дней	Контроль	Академических часов								Дней	Контроль	Академических часов								з.е.	Недель		
				Всего	Кон такт.	Лек	Лаб	Пр	КРП	СР	Конт роль			Всего	Кон такт.	Лек	Лаб	Пр	КРП	СР	Конт роль			Всего	Кон такт.	Лек	Лаб	Пр	КРП	СР	Конт роль			Всего	Кон такт.	Лек	Лаб	Пр	КРП	СР	Конт роль	Всего			
7	Б1.В.15	Технологии объектного программирования																					ЗаО	**144**	12	4	8			128	4		ЗаО	**144**	12	4	8			128	4	4		15	4

ОБОЗНАЧЕНИЯ:
Виды промежуточной аттестации (виды контроля):
Экз - экзамен;
ЗаО - зачет с оценкой;
За – зачет;

Виды работ:
Контакт. – контактная работа обучающихся с преподавателем;
Лек. – лекционные занятия;
Лаб.– лабораторные работы;
Пр. – практические занятия;
КРП – курсовая работа (курсовой проект);
РГР – расчетно-графическая работа (реферат);
СР – самостоятельная работа студентов;
з.е. – объем дисциплины в зачетных единицах.

Содержание дисциплины:

№	Наименование видов занятий и тематик, содержание
1	лекционные занятия 2 шт. по 2 часа: *1.1. Основные понятия и принципы объектно-ориентированного программирования (на примере языка программирования Java).* Введение в объектно-ориентированное программирование. Общие сведения о Java. Пакеты и имена в Java. Описание классов в Java. Реализация инкапсуляции. Достоинства и недостатки ООП. Особенности синтаксиса ООП в Java. Конструкторы и деструкторы. Модификаторы доступа к членам классов. UML- стандартный язык описания разработки программных продуктов с использованием объектного подхода. Диаграммы вариантов использования. Описание вариантов использования. Виды отношений между вариантами использования – ассоциация, расширение (extend), включение (include), обобщение. Диаграмма классов. Построение концептуальной модели предметной области. Класс: имя класса, атрибуты класса, операции. Отношения между классами – отношение зависимости, ассоциации, агрегации, композиции, обобщения. Интерфейсы. Объекты. Параметризованные классы (шаблоны). Метрические особенности объектно-ориентированных программ. Набор метрик Мартина – центростремительное сцепление, центробежное сцепление, нестабильность, абстрактность. Построение главной последовательности Дополнительные метрики Мартина – расстояние до главной последовательности, нормализованное расстояние до главной последовательности. Набор метрик Чидамбера-Кемерера – взвешенные методы на класс, глубина дерева наследования, количество потомков, связанность между классами объектов, количество откликов на класс, отсутствие сцепления в методах. Метрики Лоренца Кидда – размер класса; количество операций, переопределенных подклассом; количество операций, добавленных подклассом; индекс специализации, средний размер операции, сложность операции, среднее количество параметров на операцию, количество описаний сценариев, количество ключевых классов, количество подсистем. Метрики Фернандо Абреу – фактор закрытости метода; фактор закрытости свойства; фактор наследования метода; фактор наследования свойства; фактор полиморфизма; фактор сцепления. Синтаксис языка Java, отношения между классами и особенности разработки; особенности наследования; исключительные ситуации и их обработка; механизмы ввода и вывода информации; понятие сериализации. 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. Сокрытие полей. Завершенные и абстрактные методы и классы. Интерфейсы и их описание. Использование интерфейсов. Потоки данных и их виды. Иерархия и разновидности потоков данных. Понятие сериализации. Особенности сериализации и десериализации. Многопоточное программирование: общие принципы и реализация. Основы ООП проектирования сетевых приложений. Многопоточное программирование и его особенности Потоки 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 Понятие сокета. Абстракция сокета. Понятие порта. Пакет java.net. Классы Socket и ServerSocket. Классы DatagramPacket и DatagramSocket. Класс URL *1.2. Основные понятия паттернов (шаблонов) проектирования. Порождающие паттерны. Структурные паттерны проектирования. Поведенческие паттерны проектирования.* 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). 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). Основные особенности и назначение поведенческих шаблонов проектирования. Состав и описание основных поведенческих шаблонов. Chain Of Responsibility (Цепочка обязанностей). Command (Комманда). State (Состояние). Template Method (Шаблонный метод). Mediator (Посредник). Interpreter (Интерпретатор). Iterator (Итератор). Memento (Хранитель). Observer (Наблюдатель). Strategy (Стратегия). Visitor (Посетитель).
2	лабораторные работы 4 шт. по 2 часа: *2.1. Разработка многопоточных приложений.* Теоретическое введение. Класс Thread. Интерфейс Rannable. Синхронизация потоков. Приоритет потока. Примеры программирования. Создание многопоточного приложения на основе базового класса Thread. Создание многопоточного приложения с помощью интерфейса Rannable. *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. *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. *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.

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	Самостоятельная работа студентов: 6.1. 2 контрольных опроса после 1-й и 2-й лекций; 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. 6.3. Подготовка к зачету по дисциплине (оценочные материалы приведены в разделе 6 настоящей РПД).

5. ОБРАЗОВАТЕЛЬНЫЕ ТЕХНОЛОГИИ

Таблица - Образовательные технологии, используемые при реализации различных видов учебных занятий по дисциплине

№ п/п	Виды учебных занятий	Образовательные технологии
1	Лекции	Классическая (традиционная, информационная) лекция. Интерактивная лекция (лекция-визуализация). Индивидуальные и групповые консультации по дисциплине.
2	Лабораторная работа	Технология выполнения лабораторных заданий индивидуально. Технология выполнения лабораторных заданий в малой группе (в бригаде). Технология проблемного обучения на основе анализа результатов лабораторной работы: индивидуальный опрос, собеседование в малой группе (бригаде), обсуждение результатов командной работы, групповая дискуссия, метод «круглого стола», представление студентом или группой студентов (бригадой) результатов лабораторной работы в форме отчета и мультимедийной презентации. Проектная технология.
3	Самостоятельная работа студентов (внеаудиторная)	Информационно-коммуникационные технологии (доступ к ЭИОС филиала, к ЭБС филиала, доступ к информационно-методическим материалам по дисциплине).
4	Контроль (промежуточная аттестация: зачет или экзамен)	Технология письменного контроля, в том числе тестирование. Рейтинговая система контроля.

6. ОЦЕНОЧНЫЕ МАТЕРИАЛЫ ДЛЯ ТЕКУЩЕГО КОНТРОЛЯ И ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ – ДЛЯ ОЦЕНКИ КАЧЕСТВА ОСВОЕНИЯ ДИСЦИПЛИНЫ

К промежуточной аттестации студентов по дисциплине могут привлекаться представители

работодателей, преподаватели последующих дисциплин, заведующие кафедрами.

Оценка качества освоения дисциплины включает как текущий контроль успеваемости, так и промежуточную аттестацию.

Оценочные средства текущего контроля успеваемости:

Примеры вопросов к контрольному опросу после 1-й лекции:

1. Укажите принципы объектно-ориентированного программирования:

a) Ассоциация, агрегация, композиция. b) Зависимость, обобщение, расширение. c) Инкапсуляция, наследование, полиморфизм.

2. Инкапсуляция – это…

a) Такое свойство, при котором классы вызывают операции других классов. b) Такое свойство, при котором классы содержат описание атрибутов и действий одновременно. c) Такое свойство, при котором не обязательно создание экземпляров классов.

3. Наследование – это…

a) Такой метод определения классов, при котором классы наследуют свойства и методы своих потомков. b) Такой метод определения классов, при котором производные классы наследуют свойства и методы своих предков. c) Такой метод определения классов, при котором производные классы наследуют только свойства своих предков.

4. Полиморфизм – это…

a) Такое свойство классов, при котором действие с одинаковыми именами вызывает одинаковое поведение для экземпляров различных классов. b) Такое свойство классов, при котором действие с одинаковыми именами вызывает различное поведение для различных классов. c) Такое свойство классов, при котором изменение свойств предка вызывает изменение аналогичных свойств потомка.

5. Класс, обладающий свойством инкапсуляции, характеризуется следующими параметрами:

a) Уникальное имя, набор полей и свойств. b) Уникальное имя, набор атрибутов, набор действий для описания своего поведения. c) набор атрибутов, набор действий для описания своего поведения.

6. Атрибуты класса - это…

a) Данные, характеризующие состояние класса. b) Данные, характеризующие состояние экземпляра класса. c) Данные, характеризующие состояние и поведение класса

7. Изменение состояния объекта в ответ на какое-либо действие – это…

a) свойство. b) событие. c) сообщение.

8. Действие, которое может выполнить объект – это…

a) атрибут. b) метод. c) свойство.

9. Объектно-ориентированные технологии имеют следующие преимущества:

a) Уменьшенную связность между модулями. b) Повышенное качество кода. c) Высокий уровень абстракции. d) Все из вышеперечисленного.

10. В Объектно-ориентированной Технологии слово «UML» означает

a) Unified Module Language. b) Unified Modeling Language. c) Universal Module Leveling. d) Universal Module Language.

11 Объект имеет

a) Поведение. b) Состояние. c) Атрибуты. d) Все из вышеперечисленного.

12 Какое из следующих утверждений верно?

a) Объект - это экземпляр класса. b) Класс - это абстрактное определение для множества объектов. c) Объект может быть более чем в одном классе. d) Объект имеет линию жизни. e) Все из вышеперечисленного.

13 Состояние объекта определяется

a) Значением всех его атрибутов. b) Его связями с другими объектами. c) Его поведением в любой данный момент времени. d) Операциями, которые он может выполнять. e) Ответы b и d. f) Ответы a и b. g) Ответы a, c и d. h) Ответы a, b и c.

14 Структура класса

a) Представлена в коде. b) Представлена атрибутами и связями. c) Представлена операциями. d) Представлена взаимодействиями объектов. e) Ответы a и b. f) Ответы b и c.

15 Какое из следующих утверждений правильное:

Классы на диаграммах классов могут быть сгруппированы в пакеты, чтобы показать общую организацию модели (архитектуру системы). b) На диаграмме объектов наименования экземпляров должны быть выделены наклонным шрифтом. c) Если пакет B зависит от пакета A, то любые изменения в A вызывают изменения в пакете B. d) Диаграммы объектов и диаграммы классов взаимозаменяемы.

16 Какое из следующих утверждений о кратности роли в ассоциации является неверным?

a) “1” - именно один и только один. b) “0..n” - любое вещественное число, включая ноль. c) “0..1” - ноль или один. d) “3..7” - 3 или 7. e) “3, 7” - 3 или 7

17 На диаграмме прецеденты (UseCase diagram) есть три основных элемента, это:

a) Объекты, действия и связи. b) Субъекты (акторы), сообщения и связи. c) Объекты, прецеденты и связи. d) Субъекты (акторы), прецеденты и связи

18 Операция в классе «нечто» вызывает операцию в классе «что-то». Других отношений между двумя классами нет. Каким типом отношений является данный?

a) Ассоциация. b) Агрегация. c) Наследование. d) Реализация. e) Зависимость

19 Машина имеет 4 колеса. Какой тип отношений имеют класс “Машина” и класс “Колесо”?

a) Ассоциация. b) Агрегация. c) Наследование. d) Реализация

20 Студент посещает несколько аудиторий. Аудитория может вмещать несколько студентов. Какой тип отношений существует между студентом и аудиторией?

a) Ассоциация. b) Агрегация. c) Наследование. d) Реализация. e) Зависимость

21 Выберите отношение, которое может иметь кратность, отличную от “1”.

a) Ассоциация. b) Агрегация. c) Наследование. d) Реализация. e) Зависимость. f) Все из вышеперечисленного. g) Ответы a, c и d. h) Ответы a и b. i) Ответы c, d и e

22 Какое из следующих утверждений является неверным?

a) Класс может иметь отношение к самому себе. b) Объект может иметь отношение к другим объектам того же класса. c) Класс может иметь только одно отношение к другому классу. d) Отношение может существовать с мощностью “0’. e) Класс может существовать без каких- либо отношений к другим объектам.

23 Какой тип отношений к самому себе может иметь класс?

a) Ассоциация. b) Агрегация. c) Наследование. d) Реализация. e) Зависимость. f) Ответы a и b. g) Ответы c и e. h) Ответы d и e. i) Ничего из вышеперечисленного. j) Что такое «актор» в анализе варианта использования? k) Любая сущность, которая взаимодействует с системой. l) Термин для бизнес - объектов, которые присутствуют в системе. m) То, что реагирует на внешние воздействия системы. n) В анализе варианта использования этого термина нет.

24 Диаграмма Последовательности содержит

a) Объекты. b) Сообщения. c) Видимость объекта. d) Временные ограничения. e) Все из вышеперечисленного. f) Все из вышеперечисленного за исключением c. g) Все из вышеперечисленного за исключением a. h) Диаграмма Состояний содержит. i) Состояния класса. j) Переходы между состояниями. k) Действия, выполняемые классом. l) События, которые являются действиями в классе. m) Все из вышеперечисленного.

25 Диаграмма Состояний может иметь только одно (один)

a) Конечное состояние. b) Начальное состояние. c) Действие в состоянии. d) Переход из состояния.

e) Все из вышеперечисленного.

26 Какое из следующих утверждений является неверным?

a) В какой-либо момент времени объект может быть более чем в одном состоянии. b) Автомат может иметь множество конечных состояний. c) Автомат может запоминать, какое под состояние было последним состоянием. d) Переход может срабатывать условно.

27 Какое из следующих утверждений о диаграмме состояний является правильным:

a) Все действия на диаграмме состояний связаны с переходами. b) Событие может явиться причиной того, что объект останется в том же состоянии, предшествующем событию. c) Если объект выходит из состояния, он не может вернуться в это состояние. d) Два различных перехода из одного состояния могут перекрывать друг друга (они могут быть вызваны одним и тем же событием).

28 Названия прецедентов должны начинаться:

a) с глагола или существительного. b) с глагола. c) с прилагательного.d) с глагола или прилагательного.

Примеры вопросов к контрольному опросу после 2-й лекции:

1. Назначение паттерна делегирования (Delegation).

a) Использует делегирование, а не наследование, как универсальный способ расширения классов. b) Используется для того, чтобы избежать зависимости классов, когда один использует другой. Делает такую зависимость косвенной. c) Гарантирует согласованное поведение концептуально связанных классов, задавая для них общий абстрактный суперкласс.

2. Назначение паттерна интерфейс (Interface)

a) Используется для того, чтобы избежать зависимости классов, когда один использует другой. Делает такую зависимость косвенной. b) Использует делегирование, а не наследование, как универсальный способ расширения классов. c) Гарантирует согласованное поведение концептуально связанных классов, задавая для них общий абстрактный суперкласс.

3. Назначение паттерна абстрактный суперкласс (Abstract Superclass).

a) Гарантирует согласованное поведение концептуально связанных классов, задавая для них общий абстрактный суперкласс. b) Использует делегирование, а не наследование, как универсальный способ расширения классов. c) Используется для того, чтобы избежать зависимости классов, когда один использует другой. Делает такую зависимость косвенной.

4. Назначение паттерна Неизменный (Immutable).

a) Повышает надежность объектов, которые совместно используют ссылки на один и тот же объект, и уменьшает затраты на параллельный доступ к объекту. Это достигается путем наложения запрета на изменение содержимого совместно используемого объекта после того, как объект уже был создан. b) Гарантирует согласованное поведение концептуально связанных классов, задавая ДЛЯ них общий абстрактный суперкласс. c) Использует делегирование, а не наследование, как универсальный способ расширения классов.

5. Назначение паттерна Заместитель (Proxy).

a) Предоставляет замену другого объекта для контроля доступа к нему. b) Повышает надежность объектов, которые совместно используют ссылки на один и тот же объект, и уменьшает затраты на параллельный доступ к объекту. Это достигается путем наложения запрета на изменение содержимого совместно используемого объекта после того, как объект уже был создан. c) Используется для того, чтобы избежать зависимости классов, когда один использует другой. Делает такую зависимость косвенной.

6. Назначение паттерна Наблюдатель (Observer)

a) Определяет зависимость «один ко многим» между объектами так, что когда один объект меняет свое состояние, все зависимые объекты оповещаются и обновляются автоматически. b) Инкапсулирует запрос в виде объекта, позволяя передавать его клиентам в качестве параметров, ставить в очередь, отслеживать, а также поддерживать отмену операций. c) Не нарушая

инкапсуляцию, определяет и сохраняет внутреннее состояние объекта и позволяет позже восстановить объект в этом состоянии.

7. Назначение паттерна Цепочка обязанностей (Chain of Responsibility).

a) Избегает связывания отправителя запроса с его получателем, давая возможность обрабатывать запрос более чем одному объекту. Связывает объекты получатели и передает запрос по цепочке, пока объект не обработает его. b) Позволяет объекту изменять свое поведение в зависимости от внутреннего состояния. c) Представляет собой операцию, которая будет выполнена над объектами группы классов. Дает возможность определить новую операцию без изменения кода классов, над которыми эта операция проводится.

8. Назначение паттерна Команда (Command).

a) Инкапсулирует запрос в виде объекта, позволяя передавать его клиентам в качестве параметров, ставить в очередь, отслеживать, а также поддерживать отмену операций. b) Определяет зависимость «один ко многим» между объектами так, что когда один объект меняет свое состояние, все зависимые объекты оповещаются и обновляются автоматически. c) Определяет объект, инкапсулирующий способ взаимодействия объектов. Обеспечивает слабую связь, избавляя объекты от необходимости прямо ссылаться друг на друга и дает возможность независимо изменять их взаимодействие.

9. Назначение паттерна Посредник (Mediator).

a) Определяет объект, инкапсулирующий способ взаимодействия объектов. Обеспечивает слабую связь, избавляя объекты от необходимости прямо ссылаться друг на друга и дает возможность независимо изменять их взаимодействие. b) Позволяет объекту изменять свое поведение в зависимости от внутреннего состояния. c) Определяет алгоритм, некоторые этапы которого делегируются подклассам. Позволяет подклассам переопределить эти этапы, не меняя структуру алгоритма

10. Назначение паттерна Состояние (State).

a) Позволяет объекту изменять свое поведение в зависимости от внутреннего состояния. b) Представляет собой операцию, которая будет выполнена над объектами группы классов. Дает возможность определить новую операцию без изменения кода классов, над которыми эта операция проводится. c) Избегает связывания отправителя запроса с его получателем, давая возможность обрабатывать запрос более чем одному объекту. Связывает объекты получатели и передает запрос по цепочке пока объект не обработает его.

11. Назначение паттерна Стратегия (Strategy).

a) Определяет группу алгоритмов, инкапсулирует их и делает взаимозаменяемыми. Позволяет изменять алгоритм не зависимо от клиентов, его использующих. b) Представляет собой операцию, которая будет выполнена над объектами группы классов. Дает возможность определить новую операцию без изменения кода классов, над которыми эта операция проводится. c) Определяет зависимость «один ко многим» между объектами так, что когда один объект меняет свое состояние, все зависимые объекты оповещаются и обновляются автоматически.

12. Назначение паттерна Посетитель (Visitor).

a) Представляет собой операцию, которая будет выполнена над объектами группы классов. Дает возможность определить новую операцию без изменения кода классов, над которыми эта операция проводится. b) Определяет алгоритм, некоторые этапы которого делегируются подклассам. Позволяет подклассам переопределить эти этапы, не меняя структуру алгоритма. c) Избегает связывания отправителя запроса с его получателем, давая возможность обрабатывать запрос более чем одному объекту. Связывает объекты получатели и передает запрос по цепочке, пока объект не обработает его.

13. Назначение паттерна Шаблонный метод (Template Method).

a) Определяет алгоритм, некоторые этапы которого делегируются подклассам. Позволяет подклассам переопределить эти этапы, не меняя структуру алгоритма. b) Не нарушая инкапсуляцию, определяет и сохраняет внутреннее состояние объекта и позволяет позже восстановить объект в

этом состоянии. c) Представляет собой операцию, которая будет выполнена над объектами группы классов. Дает возможность определить новую операцию без изменения кода классов, над которыми эта операция проводится.

14. Назначение паттерна Хранитель (Memento).

a) Не нарушая инкапсуляцию, определяет и сохраняет внутреннее состояние объекта и позволяет позже восстановить объект в этом состоянии. b) Определяет группу алгоритмов, инкапсулирует их и делает взаимозаменяемыми. Позволяет изменять алгоритм не зависимо от клиентов, его использующих. c) Инкапсулирует запрос в виде объекта, позволяя передавать его клиентам в качестве параметров, ставить в очередь, определять, а также поддерживать отмену операций.

Примеры алгоритма самостоятельной работы по закреплению материала по тематике лекционных занятий:

В ходе изучения дисциплины «Технологии объектного программирования» студенты могут посещать аудиторные занятия (лекции, лабораторные занятия, консультации). Особенность изучения дисциплины состоит в выполнении комплекса лабораторных работ, главной задачей которого является получение навыков самостоятельной работы на компьютерах с использованием современных компьютерных программ, предназначенных для решения определенного круга профессиональных задач.

Важное место в овладении тем данной дисциплины отводится самостоятельной работе, при этом во время аудиторных занятий могут быть рассмотрены и проработаны наиболее важные и трудные вопросы по той или иной теме дисциплины, а более легкие вопросы могут быть изучены студентами самостоятельно.

Методика закрепления материалов лекционных занятий 1.1-1.8:

Закрепление знаний в области данной дисциплины, приобретение практических навыков проектирования программных систем с использованием шаблонного и объектно-ориентированного подходов осуществляется путем разработки программных средств по заданной предметной области:

1. Анализ требований на разработку.

Формирование требований к разрабатываемой системе; обследование объекта и обоснование необходимости создания программных средств; формирование требований пользователя к разрабатываемой системе; оформление отчета о выполнении работ и заявки на разработку программных средств; разработка концепции программного обеспечения; изучение объекта проектирования; Проведение необходимых исследовательских работ; разработка вариантов концепции и выбор варианта концепции программных средств, удовлетворяющего требованиям пользователей; оформление отчета о проделанной работе.

2. Разработка и утверждение технического задания.

Разработка и утверждение технического задания включает в себя: определение требований к программе; разработку технического обоснования разработки программы; определение стадий, этапов и сроков разработки программы и документации на нее; выбор языков программирования; определение необходимости проведения исследовательских работ на последующих стадиях; согласование и утверждение технического задания.

Результатом данной фазы является общее описание системы, включающее в себя требования к программе и требования к надежности программы. Требования к программному обеспечению включают: описание входных данных (корректных и ошибочных); роли при использовании программного обеспечения, и их интерфейсы (средств общения с пользователями); упрощения, предположения и допущения по отношению к программам; описание выходных данных; требования к надежности функционирования программы. Определяются ошибки, которые необходимо выявлять, и сообщения, которые желательно выдавать пользователю при наличии ошибок. Пере-

числяются все особые ситуации, требующие дополнительного учета и специального рассмотрения.

3. Проектирование программного обеспечения.

Проектирование включает в себя два этапа: эскизный проект и технический проект. Эскизный проект заключается в предварительной разработке структуры входных и выходных данных и общего описания алгоритма решения задачи. При разработке технического проекта структуры входных и выходных данных уточняются и определяются их формы представления. Уточняется алгоритм решения задачи, определяется семантика и синтаксис языка программирования и разрабатывается структура программы. Оба этапа сопровождаются пояснительной запиской и технико-экономическим обоснованием. В соответствии с технологией нисходящего структурного программирования программный комплекс разбивается на небольшие части – программные модули. Каждая отдельная подзадача должна быть относительно независимой и представлять собой некоторый законченный модуль программы. По итогу этапа должны быть спроектированы: словарь терминов, функциональные диаграммы IDEF0 или диаграммы потоков данных (DFD). Не менее трех уровней иерархии диаграмм (лучше больше). Оценка качества проектирования (коэффициент декомпозиции, коэффициент сбалансированности); диаграммы переходов состояний STD; схемы алгоритмов; диаграммы Джексона, структурные карты Константайна.

4. Реализация рабочего проекта.

Рабочий проект включает в себя разработку программы и программной документации, а также испытание программы. Этап кодирования алгоритмов заключается в переводе алгоритмов, разработанных для каждого программного модуля, в программы на конкретном языке программирования. Кодирование должно быть простым. Изощренное программирование может обойтись слишком дорого при отладке и модификации программы. Необычное кодирование (например, использование скрытых возможностей машины) часто препятствует отладке программы и, конечно, затрудняет ее использование другими программистами. Входные форматы должны быть разработаны с учетом максимального удобства для пользователя и минимальной возможности ошибок. Порядок переменных и форматы данных, привычные для пользователя, помогут избежать ошибок и облегчат использование программ. Форматы выходных данных могут сильно различаться. Иногда даются четкие инструкции, и выходные данные подгоняются под определенный стандарт. Результаты расчета должны быть удобочитаемыми и понятными для непрограммиста. По итогу этапа должны быть разработаны: структурные и функциональные схемы; описание разработанных процедур и функций (название, входные данные, выходные данные, ограничения); структурное тестирование (тестирование базового пути, тестирование условий, тестирование циклов), функциональное тестирование (разбиение на классы эквивалентности, анализ граничных значений, анализ причинно-следственных связей); реализация пользовательского интерфейса (построение графа диалога).

В заключении поводится оценка качества разработанного программного обеспечения.

Оценочные средства для промежуточной аттестации:

Примеры вопросов к зачету по дисциплине:

1. UML- стандартный язык описания разработки программных продуктов с использованием объектного подхода.
2. Основные компоненты языка UML.
3. Определение вариантов использования. Диаграммы вариантов использования.
4. Отношения на диаграмме вариантов использования.
5. Пример построения диаграммы вариантов использования.
6. Диаграмма классов.
7. Отношения между классами.
8. Интерфейсы. Объекты. Параметризованные классы.

9. Пример построения диаграммы классов.
10. Диаграммы состояний объекта.
11. Переходы на диаграмме состояния.
12. Диаграммы состояния. Последовательные состояния. Параллельные состояния.
13. Сложные переходы на диаграмме состояния.
14. Диаграмма деятельности. Состояние действия. Переходы. Дорожки. Объекты.
15. Диаграмма последовательности.
16. Сообщения на диаграмме последовательности.
17. Пример построения диаграммы последовательности.
18. Диаграмма пакетов.
19. Диаграмма кооперации.
20. Связи и сообщения на диаграмме кооперации.
21. Пример построения диаграммы кооперации.
22. Диаграмма компонентов. Виды компонентов. Интерфейсы. Зависимости.
23. Диаграмма размещения.
24. Пример построения диаграммы размещения.
25. Объектно-ориентированное тестирование. Разработка CRC-карт.
26. Объектно-ориентированное тестирование. Тестирование, основанное на ошибках.
27. Объектно-ориентированное тестирование. Тестирование, основанное на сценариях.
28. Стохастическое тестирование класса.
29. Тестирование разбиений на уровне класса.
30. Характеристики ООП дизайна приложений. Связность (coupling) и сфокусированность (cohesion).
31. Принципы дизайна. Принципы SOLID, YAGNI, KISS.
32. Оценка характеристик программ на основе объектно-ориентированных меток. Метрики Мартина
33. Пример использования метрик Мартина.
34. Оценка характеристик программ на основе объектно-ориентированных меток. Метрики Абреу.
35. Пример использования метрик Абреу.
36. Оценка характеристик программ на основе объектно-ориентированных меток. Метрики Чидамбера и Кемерера.
37. Пример использования метрик Чидамбера и Кемерера.
38. Оценка характеристик программ на основе объектно-ориентированных меток. Метрики Лоренца и Кидда.
39. Пример использования метрик Лоренца и Кидда.
40. Понятие паттернов проектирования, принципы классификации.
41. Порождающие паттерны проектирования. Паттерн Абстрактная фабрика (Abstract Factory).
42. Порождающие паттерны проектирования. Паттерн Прототип (Prototype).
43. Порождающие паттерны проектирования. Паттерн Одиночка (Singleton).
44. Структурные паттерны проектирования. Паттерн Фасад (Facade).
45. Структурные паттерны проектирования. Паттерн Адаптер (Adapter).
46. Структурные паттерны проектирования. Паттерн Декоратор (Decorator).
47. Поведенческие паттерны проектирования. Паттерн Observer (Наблюдатель).
48. Поведенческие паттерны проектирования. Паттерн Итератор (Iterator).
49. Поведенческие паттерны проектирования. Паттерн Стратегия (Strategy).
50. Понятие сериализации. Особенности сериализации и десериализации.

В филиале используется система с традиционной шкалой оценок – «отлично», «хорошо», «удовлетворительно», «неудовлетворительно», «зачтено», «не зачтено» (далее - пятибалльная система).

Форма промежуточной аттестации по настоящей дисциплине – **зачет с оценкой в 6-м семестре.**

Применяемые критерии оценивания по дисциплинам (в соответствии с инструктивным письмом НИУ МЭИ от 14 мая 2012 года № И-23):

Оценка по дисциплине	Критерии оценки результатов обучения по дисциплине
«отлично»/ «зачтено (отлично)»/ «зачтено»	Выставляется обучающемуся, обнаружившему всестороннее, систематическое и глубокое знание материалов изученной дисциплины, умение свободно выполнять задания, предусмотренные программой, усвоивший основную и знакомый с дополнительной литературой, рекомендованной рабочей программой дисциплины; проявившему творческие способности в понимании, изложении и использовании материалов изученной дисциплины, безупречно ответившему не только на вопросы билета, но и на дополнительные вопросы в рамках рабочей программы дисциплины, правильно выполнившему практическое задание. Оценка по дисциплине выставляются обучающемуся с учётом результатов текущего контроля. Компетенции, закреплённые за дисциплиной, сформированы на уровне – «эталонный».
«хорошо»/ «зачтено (хорошо)»/ «зачтено»	Выставляется обучающемуся, обнаружившему полное знание материала изученной дисциплины, успешно выполняющему предусмотренные задания, усвоившему основную литературу, рекомендованную рабочей программой дисциплины; показавшему систематический характер знаний по дисциплине, ответившему на все вопросы билета, правильно выполнивший практическое задание, но допустивший при этом непринципиальные ошибки. Оценка по дисциплине выставляются обучающемуся с учётом результатов текущего контроля. Компетенции, закреплённые за дисциплиной, сформированы на уровне – «продвинутый».
«удовлетворительно»/ «зачтено (удовлетворительно)»/ «зачтено»	Выставляется обучающемуся, обнаружившему знание материала изученной дисциплины в объеме, необходимом для дальнейшей учебы и предстоящей работы по профессии, справляющемуся с выполнением заданий, знакомому с основной литературой, рекомендованной рабочей программой дисциплины; допустившему погрешность в ответе на теоретические вопросы и/или при выполнении практических заданий, но обладающему необходимыми знаниями для их устранения под руководством преподавателя, либо неправильно выполнившему практическое задание, но по указанию преподавателя выполнившему другие практические задания из того же раздела дисциплины.. Компетенции, закреплённые за дисциплиной, сформированы на уровне – «пороговый».
«неудовлетворительно»/ не зачтено	Выставляется обучающемуся, обнаружившему серьезные пробелы в знаниях основного материала изученной дисциплины, допустившему принципиальные ошибки в выполнении заданий, не ответившему на все вопросы билета и дополнительные вопросы и неправильно выполнившему практическое задание (неправильное выполнение только практического задания не является однозначной причиной для выставления оценки «неудовлетворительно»). Как правило, оценка «неудовлетворительно ставится студентам, которые не могут продолжить обуче-

Оценка по дисциплине	Критерии оценки результатов обучения по дисциплине
	ние по образовательной программе без дополнительных занятий по соответствующей дисциплине. Оценка по дисциплине выставляются обучающемуся с учётом результатов текущего контроля. Компетенции на уровне «пороговый», закреплённые за дисциплиной, не сформированы.

7. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

Учебное и учебно-лабораторное оборудование

Учебная аудитория для проведения занятий семинарского типа, групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации, оснащенная: специализированной мебелью; доской аудиторной; демонстрационным оборудованием: персональным компьютером (ноутбуком); переносным (стационарным) проектором.

Учебная аудитория для лабораторных работ, выполняемых в компьютерном классе, оснащенная: специализированной мебелью; доской аудиторной; персональными компьютерами с подключением к сети "Интернет" и доступом в ЭИОС филиала.

Для самостоятельной работы обучающихся по дисциплине используется помещение для самостоятельной работы обучающихся, оснащенное: специализированной мебелью; доской аудиторной; персональным компьютерами с подключением к сети «Интернет» и доступом в ЭИОС филиала.

Программное обеспечение (Операционная система OS Windows, офисный пакет Microsoft Office, Microsoft Visual Studio 2019 Community, Draw.io, Java Eclipse IDE)

8. ОБЕСПЕЧЕНИЕ ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА ДЛЯ ЛИЦ С ОГРАНИЧЕННЫМИ ВОЗМОЖНОСТЯМИ ЗДОРОВЬЯ И ИНВАЛИДОВ

В ходе реализации дисциплины используются следующие дополнительные методы обучения, текущего контроля успеваемости и промежуточной аттестации обучающихся в зависимости от их индивидуальных особенностей:

для слепых и слабовидящих:

- лекции оформляются в виде электронного документа, доступного с помощью компьютера со специализированным программным обеспечением;
- письменные задания выполняются на компьютере со специализированным программным обеспечением, или могут быть заменены устным ответом;
- обеспечивается индивидуальное равномерное освещение не менее 300 люкс;
- для выполнения задания при необходимости предоставляется увеличивающее устройство; возможно также использование собственных увеличивающих устройств;
- письменные задания оформляются увеличенным шрифтом;
- экзамен и зачёт проводятся в устной форме или выполняются в письменной форме на компьютере.

для глухих и слабослышащих:

- лекции оформляются в виде электронного документа;
- письменные задания выполняются на компьютере в письменной форме;
- экзамен и зачёт проводятся в письменной форме на компьютере; возможно проведение в форме тестирования.

для лиц с нарушениями опорно-двигательного аппарата:

- лекции оформляются в виде электронного документа, доступного с помощью компьютера со специализированным программным обеспечением;
- письменные задания выполняются на компьютере со специализированным программным обеспечением;
- экзамен и зачёт проводятся в устной форме или выполняются в письменной форме на компьютере;
- используется специальная учебная аудитория для лиц с ЛОВЗ – ауд. 106 главного учебного корпуса по адресу 214013, г. Смоленск, Энергетический пр-д, д.1, здание энергетического института (основной корпус).

При необходимости предусматривается увеличение времени для подготовки ответа.

Процедура проведения промежуточной аттестации для обучающихся устанавливается с учётом их индивидуальных психофизических особенностей. Промежуточная аттестация может проводиться в несколько этапов.

При проведении процедуры оценивания результатов обучения предусматривается использование технических средств, необходимых в связи с индивидуальными особенностями обучающихся. Эти средства могут быть предоставлены филиалом, или могут использоваться собственные технические средства.

Проведение процедуры оценивания результатов обучения допускается с использованием дистанционных образовательных технологий.

Обеспечивается **доступ к информационным и библиографическим ресурсам в сети Интернет** для каждого обучающегося в формах, адаптированных к ограничениям их здоровья и восприятия информации:

для слепых и слабовидящих:

- в печатной форме увеличенным шрифтом;
- в форме электронного документа;

- в форме аудиофайла.

для глухих и слабослышащих:
- в печатной форме;
- в форме электронного документа.

для обучающихся с нарушениями опорно-двигательного аппарата:
- в печатной форме;
- в форме электронного документа;
- в форме аудиофайла.

9. УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

Основная литература.

1. Барков, И. А. Объектно-ориентированное программирование: учебник / И.А. Барков. – Санкт-Петербург: Лань, 2019. – 700 с. – ISBN 978-5-8114-3586-9. – // Лань: электронно-библиотечная система. — URL: https://e.lanbook.com/book/119661.

2. Кирсяев А.Н. Теория и технология программирования. Программное обеспечение вычислительной математики [Электронный ресурс] : учебное пособие / А.Н. Кирсяев. – Электрон. дан. – Санкт-Петербург: СПбГПУ, 2017. – 104 с. URL: https://e.lanbook.com/book/105484.

3. Мишова В.В. Технологии программирования: практикум / В.В. Мишова ; Министерство культуры Российской Федерации, Кемеровский государственный институт культуры, Институт информационных и библиотечных технологий. – Кемерово : Кемеровский государственный институт культуры (КемГИК), 2016. – 87 с. : ил. – Режим доступа: по подписке. – URL: https://biblioclub.ru/index.php?page=book&id=472686. – Библиогр.: с. 84. – ISBN 978-5-8154-0360-4. – Текст: электронный.

4. Петрухин, В. А. Методы и средства инженерии программного обеспечения: учебное пособие / В. А. Петрухин, Е. М. Лаврищева. – 2-е изд. – Москва: ИНТУИТ, 2016. – 467 с. URL: https://e.lanbook.com/book/100645.

Дополнительная литература.

1. Кручинин В.В. Технологии программирования [Электронный ресурс] : учебное пособие / В.В. Кручинин. — Электрон. дан. — Москва : ТУСУР, 2013. — 271 с. — Режим доступа: https://e.lanbook.com/book/110371.

2. Коузен, К. Современный Java: рецепты программирования / К. Коузен. – Москва : ДМК Пресс, 2018. – 275 с. – ISBN 978-5-97060-134-1. – URL: https://e.lanbook.com/book/116121.

Список авторских методических разработок.

Малашенкова И.В., Панкратова Е.А. Паттерны проектирования в Java : учеб. пособие по курсу «Объектно-ориентированные технологии» [по напр. подготовки бакалавров 09.03.01 «Информатика и вычислительная техника»] / И.В. Малашенкова, Е.А. Панкратова ; Филиал ФГБОУ ВО «НИУ МЭИ» в г. Смоленске .– Смоленск : Универсум, 2018 .– 123, [1] с. : ил. – Библиогр.: с. 121 .– ISBN 978-5-91412-372-4 : 104.55.

Панкратова Е.А. Тестирование программного обеспечения [Текст]: методические рекомендации / Е.А. Панкратова, О.В. Семенова - Смоленск: РИО филиала МЭИ в г. Смоленске, 2011. – 24 с.

Федулов Я.А. Комплект мультимедийных презентаций к лекциям по дисциплине «Технологии объектного программирования» (расположен в ЭИОС филиала и передается обучающимся на 1-й лекции для подготовки к лекциям и самостоятельного изучения дисциплины).

ЛИСТ РЕГИСТРАЦИИ ИЗМЕНЕНИЙ

Номер изменения	Номера страниц				Всего страниц в документе	Наименование и № документа, вводящего изменения	Подпись, Ф.И.О. внесшего изменения в данный экземпляр	Дата внесения изменения в данный экземпляр	Дата введения изменения
	измененных	замененных	новых	аннулированных					
1	2	3	4	5	6	7	8	9	10