

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

УТВЕРЖДАЮ

Зам. директора
филиала ФГБОУ ВО «НИУ «МЭИ»
в г. Смоленске
по учебно-методической работе
В.В. Рожков
« 28 » // 2018 г.



**РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ
ТЕХНОЛОГИЯ ОБЪЕКТНОГО ПРОГРАММИРОВАНИЯ**

(Название дисциплины)

Направление подготовки: 09.03.01 Информатика и вычислительная техника

Профиль подготовки: Вычислительные машины, комплексы, системы и сети

Уровень высшего образования: бакалавриат

Нормативный срок обучения: 4 года

Форма обучения: очная

Смоленск – 2018 г.

1 Цель и задачи освоения дисциплины (модуля)

Целью освоения дисциплины «Технология объектного программирования» является подготовка обучающихся по направлению подготовки 09.03.01 "Информатика и вычислительная техника" посредством обеспечения этапов формирования компетенций, предусмотренных ФГОС, в части представленных ниже знаний, умений и навыков.

Задачами дисциплины являются:

- обучение студентов систематизированному представлению о принципах построения и проектирования объектно-ориентированных программных систем;
- изучение методов объектно-ориентированного тестирования;
- изучение методов оценки качества объектно-ориентированных приложений;
- изучение типовых приемов проектирования (паттернов проектирования)

Дисциплина направлена на формирование следующих общепрофессиональных и профессиональных компетенций:

- ОПК-1. Способностью устанавливать программное и аппаратное обеспечение для информационных и автоматизированных систем
- ОПК-2. Способностью осваивать методики использования программных средств для решения практических задач
- ОПК-4. Способностью участвовать в настройке и наладке программно-аппаратных комплексов
- ПК-1. Способностью разрабатывать модели компонентов информационных систем, включая модели баз данных и модели интерфейсов "человек-электронно-вычислительная машина"
- ПК-2. способностью разрабатывать компоненты аппаратно-программных комплексов и баз данных, используя современные инструментальные средства и технологии программирования

В результате изучения курса студент должен знать:

- методологию объектно-ориентированного проектирования программного обеспечения с использованием унифицированного языка моделирования UML (ОПК-1, ОПК-2);
- принципы тестирования объектно-ориентированных приложений (ОПК-4);
- методы оценки качества объектно-ориентированных приложений (ПК-1);
- типовые приемы проектирования (паттерны проектирования) (ПК-2).

В результате изучения курса студент должен уметь:

- использовать методы декомпозиции и абстракции при проектировании программного обеспечения (ПК-2);
- применять средства разработки программного обеспечения: инструментальные среды разработки, средства поддержки проекта, отладчики (ОПК-2);
- документировать и оценивать качество программных продуктов (ПК-1);
- выполнять тестирование объектно-ориентированных приложений (ОПК-4);
- использовать готовые решения (ОПК-1).

В результате изучения курса студент должен владеть:

- методами проектирования программного обеспечения при объектно-ориентированном подходе (ОПК-2);
- методами объектно-ориентированного тестирования (ОПК-4, ПК-2)
- методами оценки качества объектно-ориентированных приложений (ПК-2);
- методикой использования в разрабатываемых приложениях готовых решений (ОПК-1, ПК-1)

2 Место дисциплины в структуре образовательной программы

Дисциплина «Технология объектного программирования» относится дисциплинам по выбору вариативной части профессионального цикла Б1.В.ДВ.8.1 основной образовательной программы подготовки бакалавров по направлению 09.03.01 «Информатика и вычислительная техника».

В соответствии с учебным планом по направлению "Информатика и вычислительная техника" дисциплина «Технология объектного программирования» базируется на следующих дисциплинах:

- Б1.Б.13 Информатика
- Б1.Б.9.1 Алгебра и геометрия
- Б1.Б.9.2 Математический анализ
- Б1.Б.12 Инженерная и компьютерная графика
- Б1.В.ОД.1 Математическая логика и теория алгоритмов
- Б1.Б.7 Физика
- Б1.Б.11 Дискретная математика
- Б2.У.1 Практика по получению первичных профессиональных умений и навыков, в том числе первичных умений и навыков научно-исследовательской деятельности
- Б2.У.2 Исполнительская практика
- Б1.Б.15.1 ЭВМ
- Б1.В.ОД.2 Программирование
- Б1.В.ОД.3 Операционные системы
- Б1.В.ОД.4 Компьютерная графика
- Б1.В.ОД.5 Технология программирования
- Б1.В.ДВ.2.1 Введение в оптимизацию
- Б1.В.ДВ.2.2 Программные средства для математических расчетов
- Б1.Б.8 Вычислительная математика
- Б1.Б.10 Теория вероятностей и математическая статистика
- Б2.П.1 Практика по получению профессиональных умений и опыта профессиональной деятельности
- Б1.В.ОД.9 Базы данных
- Б1.В.ОД.6 Прикладная статистика
- Б1.В.ОД.7 Электронные цепи ЭВМ
- Б1.В.ОД.8 Основы теории управления
- Б1.В.ОД.10 Теория передачи информации
- Б1.В.ОД.11 Метрология, стандартизация и сертификация
- Б1.В.ДВ.3.1 Теория принятия решений
- Б1.В.ДВ.3.2 Исследование операций
- Б1.В.ДВ.4.1 Основы логического программирования
- Б1.В.ДВ.4.2 Кластерные вычислительные системы
- Б1.В.ДВ.5.1 Инженерное проектирование и САПР
- Б1.В.ДВ.5.2 Лингвистическое и программное обеспечение САПР
- Б1.В.ДВ.6.1 Теория автоматов
- Б1.В.ДВ.6.2 Аппаратные и программные средства
- Б1.В.ОД.7 Электронные цепи ЭВМ
- Б2.П.3 Технологическая практика

Знания, умения и навыки, полученные студентами в процессе изучения дисциплины «Технологии объектного программирования» необходимы для формирования компетенций в следующих дисциплинах:

- Б1.Б.15.2 Периферийные устройства
- Б1.В.ОД.15 Сети и телекоммуникации
- Б1.В.ОД.16 Защита информации
- Б1.В.ДВ.7.1 Структурный анализ и проектирование информационных систем
- Б1.В.ДВ.7.2 Информационные технологии
- Б1.В.ДВ.8.2 Вычислительные системы
- Б1.В.ОД.13 Микропроцессорные системы
- Б1.В.ОД.14 Моделирование
- Б1.В.ОД.17 Конструкторско-технологическое обеспечение производства ЭВМ
- Б1.В.ОД.12 Системное программное обеспечение
- Б2.П.4 Преддипломная практика
- Б3 Государственная итоговая аттестация

3 Объем дисциплины в зачетных единицах с указанием количества академических часов, выделенных на контактную работу обучающихся с преподавателем (по видам учебных занятий) и на самостоятельную работу обучающихся)

Общая трудоемкость дисциплины составляет 8 зачетных единицы, 288 часа.

Объем занятий, проводимых в интерактивной форме, 16 часов

Аудиторная работа

Цикл:	Б1	Семестр
Часть цикла:	Вариативная Дисциплина по выбору	
Индекс дисциплины по учебному плану	Б1.В.ДВ.8.1	
Часов всего по учебному плану	288	7 семестр
Трудоемкость в зачетных единицах (ЗЕТ)	8	7 семестр
Лекции (ЗЕТ/ часов)	1/36	7 семестр
Лабораторные работы (ЗЕТ/ часов)	1/36	7 семестр
Объем самостоятельной работы по учебному плану (ЗЕТ/ часов всего)	5/180	7 семестр
Экзамен	1/36	7 семестр

Самостоятельная работа студента

Вид работ	Трудоёмкость	
	ЗЕТ	час
Подготовка к лекции	0,94	34
Изучение дополнительного теоретического материала	0,89	32
Подготовка к сеансу тестирования	0,56	20
Подготовка к контрольной работе	0,22	8
Подготовка к выполнению и защите лабораторных работ (лаб)	1,89	68
Расчетно-графическая работа	0,5	18
Всего:	5	180

4 Содержание дисциплины, структурированное по темам с указанием отведенного на них количества академических часов и видов учебных занятий

№ п/п	Темы дисциплины	Общая трудоемкость, всего	Виды учебной работы, включая самостоятельную работу студентов и трудоемкость (в часах)							
			Аудиторные занятия			Экзамен	Самостоятельная работа			Занятия в интерактивной форме
			Всего	Лекции	Лабораторные работы		Всего	Реферат, РГР	Другая	
1	Введение в объектно-ориентированный анализ и проектирование	30	8	4	4		22	2	20	2
2	Унифицированный язык моделирования UML	36	10	6	4		26	2	24	2
3	Основы объектно-ориентированного тестирования программных систем	30	8	4	4		22	2	20	2
4	Оценка характеристик программ на основе объектно-ориентированных метрик	34	12	6	6		22	2	20	2
5	Введение в паттерны проектирования. Базовые паттерны проектирования	30	8	4	4		22	2	20	2
6	Порождающие паттерны	26	8	4	4		18	2	16	2
7	Структурные паттерны	29	8	4	4		21	3	18	2
8	Поведенческие паттерны	37	10	4	6		27	3	24	2
Экзамен		36				36				
Всего		288	72	36	36	36	180	18	162	16

Тема 1. Введение в объектно-ориентированный анализ и проектирование

Лекция 1 (2 часа)

Предмет объектно-ориентированного анализа и проектирования. Понятие повторного использования проектных решений. Основные концепции ООП с точки зрения повторного использования. Влияние повторного использования на механизмы современных языков программирования.

Лекция 2 (2 часа)

Инкапсуляция, наследование, полиморфизм. Достоинства и недостатки. Элементарные приемы объектного проектирования: композиция и делегирование. Разница между объектным и структурным подходами.

Лабораторная работа 1 (2 часа)

Знакомство со средой Java Eclipse.

Лабораторная работа 2 (2 часа)

Разработка многопоточного приложения. Цель работы – ознакомиться с библиотеками Java Eclipse для создания многопоточных приложений и синхронизации работы потоков.

Самостоятельная работа

Тема учебной дисциплины	Вид самостоятельной работы студента (реферат, расчетно-графическая работа, др.)	Всего часов
Тема 1. Введение в объектно-ориентированный анализ и проектирование	Подготовка к лекции (2 лекции)	4
	Оформление и подготовка к защите лабораторной работы (2 лабораторные работы)	8
	Подготовка к сеансу тестирования	4
	Изучение дополнительного теоретического материала	4
	Расчетное задание	2
ИТОГО:		22

Текущий контроль – устные опросы по лекциям 1 и 2, тестирование по теме 1.

Тестирование

Цель тестирования - проверка знаний у студентов методов объектно-ориентированного программирования (ООП) и основных принципов ООП.

Коды формируемых компетенций: ОПК-1, ПК-2

Результаты освоения:

ОПК-1: Иметь навыки работы в среде программирования Java. Владеть типовыми приемами проектирования, инструментарием для документирования проектных решений.

ПК-2: знать особенности разработки программного обеспечения, ориентированного на повторное использование программного кода.

Тема 2. Унифицированный язык моделирования UML

Лекция 3 (2 часа)

Основные концепции Унифицированного языка моделирования: визуализация, прямое и обратное проектирование, специфицирование, документирование. Виды диаграмм UML.

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).

Лекция 4 (2 часа)

Диаграмма деятельности. Состояние действия. Переходы. Дорожки. Объекты.

Диаграмма последовательностей. Линия жизни объекта. Фокус управления. Сообщения. Ветвление потока управления. Стереотипы сообщений. Временные ограничения на диаграммах последовательностей.

Диаграмма кооперации. Объекты. Составные объекты. Связи. Сообщения.

Диаграмма пакетов.

Лекция 5 (2 часа)

Диаграммы состояний объекта. Состояние – имя состояния, список внутренних действий, начальное состояние, конечное состояние. Переход. Событие. Сторожевое условие. Выражение действия. Составное состояние и подсостояние. Последовательные подсостояния. Параллельные подсостояния. Историческое состояние. Сложные переходы.

Диаграмма компонентов. Имя компонента. Виды компонентов. Интерфейсы. Зависимости. Диаграмма размещения. Узел. Соединения.

Обзор CASE-средств для построения диаграмм UML

Лабораторная работа 3 (2 часа)

Разработка объектно-ориентированного приложения с оконным интерфейсом в среде Java Eclipse.

Лабораторная работа 4 (2 часа)

Первое знакомство с паттернами (Паттерн наблюдатель) Создание UML- диаграмм.

Самостоятельная работа

Тема учебной дисциплины	Вид самостоятельной работы студента (реферат, расчетно-графическая работа, др.)	Всего часов
Тема 2. Унифицированный язык моделирования UML	Подготовка к лекции (3 лекции)	6
	Оформление и подготовка к защите лабораторной работы (2 лабораторные работы)	8
	Подготовка к сеансу тестирования	4
	Подготовка к контрольной работе	2
	Изучение дополнительного теоретического материала	4
	Расчетное задание	2
ИТОГО:		26

Текущий контроль – устные опросы по лекциям 3, 4 и 5 , тестирование по теме 2, контрольная работа по теме 2.

Тестирование

Цель тестирования – проверка знаний основных принципов языка моделирования UML .

Контрольная работа

Цель контрольной работы – закрепление практических навыков моделирования объектно-ориентированных приложений с помощью языка UML.

Коды формируемых компетенций: ОПК-2, ПК-1

Результаты освоения:

ОПК-2: Имеет навыки работы CASE-средствами для разработки UML-диаграмм.

ПК-1: способен разрабатывать программные комплексы с использованием унифицированного языка UML, современных CASE-средств. Способен разрабатывать приложения в среде Java Eclipse

Тема 3. Основы объектно-ориентированного тестирования программных систем

Лекция 6 (2 часа)

Тестирование объектно-ориентированной интеграции. Объектно-ориентированное тестирование правильности. Проектирование объектно-ориентированных тестовых вариантов. Способы тестирования содержания класса.

Лекция 7 (2 часа)

Способы тестирования взаимодействия классов. Предваряющее тестирование при экстремальной разработке.

Лабораторная работа 5 (2 часа)

Разработать объектно-ориентированное приложение в среде Java Eclipse.

Лабораторная работа 6 (2 часа)

Паттерн «Iterator». Выполнить объектно-ориентированное тестирование разработанного приложения.

Самостоятельная работа

Тема учебной дисциплины	Вид самостоятельной работы студента (реферат, расчетно-графическая работа, др.)	Всего часов
Тема 3. Основы объектно-ориентированного тестирования программных систем	Оформление и подготовка к защите лабораторных работ (2 лабораторные работы)	8
	Подготовка к лекции (2 лекции)	4
	Подготовка к сеансу тестирования	4
	Изучение дополнительного теоретического материала	4
	Расчетное задание	2
ИТОГО:		22

Текущий контроль – устные опросы по лекциям 6 и 7, тестирование по теме 3.

Тестирование

Цель тестирования – проверка знаний методов объектно-ориентированного тестирования приложений (классов, операций класса, взаимодействия классов).

Коды формируемых компетенций:

Результаты освоения: ОПК-4

ОПК-4: владеет методами тестирования объектно-ориентированных программных систем.

Тема 4. Оценка характеристик программ на основе объектно-ориентированных метрик

Лекция 8 (2 часа)

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.

Лекция 9 (2 часа)

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.

Лекция 10 (2 часа)

Метрики Фернандо Абреу – фактор закрытости метода; фактор закрытости свойства; фактор наследования метода; фактор наследования свойства; фактор полиморфизма; фактор сцепления.

Лабораторная работа 7 (2 часа)

Разработать приложение в среде Java Eclipse.

Лабораторная работа 8 (2 часа)

Выполнить оценку характеристик разработанного в лабораторной работе 8 приложения с помощью метрик Мартина, Чидамбера-Кемерера, Лоренца-Кидда, Абреу.

Самостоятельная работа

Тема учебной дисциплины	Вид самостоятельной работы студента (реферат, расчетно-графическая работа, др.)	Всего часов
Тема 4. Оценка характеристик программ на основе объектно-ориентированных метрик	Подготовка к лекции (3 лекции)	6
	Оформление и подготовка к защите лабораторных работ (2 лабораторные работы)	8
	Изучение дополнительного теоретического материала	4
	Подготовка к контрольной работе	2
	Расчетное задание	2
ИТОГО:		22

Текущий контроль – устные опросы по лекциям 8, 9 и 10, контрольная работа по теме 4.

Контрольная работа

Цель контрольной работы – повторение методов оценки характеристик объектно-ориентированных программ, применение методов оценки для практических задач.

Коды формируемых компетенций: ПК-1

Результаты освоения:

ПК-1: способен оценивать качество разработанных приложений с использованием метрик объектно-ориентированных программных систем.

Тема 5. Введение в паттерны проектирования. Базовые паттерны проектирования

Лекция 11 (2 часа)

Требования к программному обеспечению. Архитектура и дизайн программного обеспечения, влияние архитектуры на его свойства. Жизненный цикл. Особенности разработки сложных программных систем. Модели программного обеспечения. Алгоритмическая и объектная декомпозиция. Рефакторинг. Повторное использование кода.

Лекция 12 (2 часа)

Понятие паттернов проектирования, принципы классификации. Паттерн: имя, задача, решение, результаты. Характерные ошибки проектирования.

Базовые паттерны проектирования.

Лабораторная работа 9 (2 часа)

Разработать объектно-ориентированное приложение для работы с файлами.

Лабораторная работа 10 (2 часа)

Использование базовых паттернов проектирования.

Самостоятельная работа

Тема учебной дисциплины	Вид самостоятельной работы студента (реферат, расчетно-графическая работа, др.)	Всего часов
Тема 5. Введение в паттерны проектирования	Подготовка к лекции (2 лекции)	4
	Изучение дополнительного теоретического материала	4
	Оформление и подготовка к защите лабораторных работ (2 лабораторные работы)	8
	Подготовка к сеансу тестирования	4
	Расчетное задание	2
ИТОГО:		22

Текущий контроль – устные опросы по лекциям 11 и 12, тестирование по теме 5.

Тестирование

Цель тестирования – закрепление знаний по механизмам повторного использования кода, правилам описания паттернов проектирования.

Коды формируемых компетенций: ОПК-2

Результаты освоения:

ОПК-2: знает основы работы с паттернами проектирования, классификацию паттернов проектирования. Умеет применять паттерны проектирования в среде Java для решения практических задач.

Тема 6. Порождающие паттерны

Лекция 13 (2 часа)

Фабричный метод (Factory Method) или виртуальный конструктор (Virtual Constructor), Одиночка (Singleton).

Лекция 14 (2 часа)

Абстрактная фабрика (Abstract Factory), Строитель (Builder), Прототип, Пул объектов.

Лабораторная работа 11 (4 часа)

Использование порождающих паттернов при работе с базами данных. Разработка баз данных. Доступ к базам данных с использованием интерфейса JDBC.

Лабораторная работа 12 (2 часа)

Формирование SQL-запросов. Создание интерфейса пользователя при работе с базами данных

Самостоятельная работа

Тема учебной дисциплины	Вид самостоятельной работы студента (реферат, расчетно-графическая работа, др.)	Всего часов
Тема 6. Порождающие паттерны	Подготовка к лекции	2
	Изучение дополнительного теоретического материала	4
	Оформление и подготовка к защите лабораторных работ (2 лабораторные работы)	8
	Подготовка к контрольной работе	2
	Расчетное задание	2
ИТОГО:		18

Текущий контроль – устные опросы по лекциям 13 и 14, контрольная работа по теме 6.

Контрольная работа

Цель контрольной работы – закрепление знаний по практическому применению порождающих паттернов проектирования.

Коды формируемых компетенций: ОПК-2, ОПК-4

Результаты освоения:

ОПК-2: умеет применять паттерны проектирования при решении практических задач.

ОПК-4: умеет использовать паттерны проектирования при разработке моделей компонентов информационных систем.

Тема 7. Структурные паттерны

Лекция 15 (2 часа)

Адаптер (Adapter), Итератор (Iterator), Мост (Bridge), Фасад (Facade), Приспособленец (Flyweight), Динамическая компоновка (Composite).

Лекция 16 (2 часа)

Виртуальный заместитель, Декоратор (Decorator) или Оболочка (Wrapper)

Лабораторная работа 13 (2 часа)

Разработка игры. Использование структурных паттернов при разработке игры.

Лабораторная работа 14 (2 часа)

Разработка интерфейса пользователя для игры из лабораторной работы 13..

Самостоятельная работа

Тема учебной дисциплины	Вид самостоятельной работы студента (реферат, расчетно-графическая работа, др.)	Всего часов
Тема 7. Структурные паттерны	Подготовка к лекции (2 лекции)	4
	Изучение дополнительного теоретического материала	4
	Оформление и подготовка к защите лабораторных работ (2 лабораторные работы)	8
	Подготовка к контрольной работе	2
	Расчетное задание	3
ИТОГО:		21

Текущий контроль – устные опросы по лекциям 15 и 16 , контрольная работа по теме 7.

Контрольная работа

Цель контрольной работы – закрепление знаний по практическому применению структурных паттернов проектирования.

Коды формируемых компетенций: ОПК-2, ОПК-4

Результаты освоения:

ОПК-2: умеет применять паттерны проектирования при решении практических задач.

ОПК-4: умеет использовать паттерны проектирования при разработке моделей компонентов информационных систем.

Тема 8. Поведенческие паттерны

Лекция 17 (2 часа)

Цепочка обязанностей (Chain of Responsibility), Команда (Command), Посредник (Mediator), Наблюдатель (Observer), Состояние (State).

Лекция 18 (2 часа)

Стратегия (Strategy), Посетитель (Visitor), Шаблонный метод (Template Method), Хранитель (Memento).

Лабораторная работа 15 (2 часа)

Использование поведенческих паттернов при сетевом программировании: программирование сокетов.

Лабораторная работа 16 (2 часа)

WEB-программирование в Java.

Лабораторная работа 17 (2 часа)

Использование поведенческих паттернов при сетевом программировании: создание сервлетов. Разработка интерфейса пользователя. Подключение к сервлету базы данных.

Самостоятельная работа

Тема учебной дисциплины	Вид самостоятельной работы студента (реферат, расчетно-графическая работа, др.)	Всего часов
Тема 8. Поведенческие паттерны	Подготовка к лекции (2 лекции)	4
	Изучение дополнительного теоретического материала	4
	Оформление и подготовка к защите лабораторных работ (2 лабораторные работы)	12
	Подготовка к сеансу тестирования	4
	Расчетное задание	3
ИТОГО:		27

Текущий контроль – устные опросы по лекциям 17 и 18 , тестирование.

Тестирование

Тестирование является итоговым по паттернам проектирования. Цель тестирования – закрепление знаний, полученных на лекциях по паттернам проектирования и при выполнении лабораторных работ.

Коды формируемых компетенций: ОПК-4, ПК-1, ПК-2

Результаты освоения:

ОПК-1: имеет навыки применения паттернов проектирования для программирования сетевых приложений и при WEB-программировании.

ПК-1: умеет применять паттерны проектирования при решении практических задач.

ПК-2: умеет использовать паттерны проектирования при разработке моделей компонентов информационных систем.

Лабораторные работы (16 часов) проводятся в интерактивной форме. Каждому студенту выдается индивидуальное задание. Затем организуется активный диалог студентов с преподавателем и между собой для подведения итогов решения задания и определения его практической значимости.

Промежуточная аттестация по дисциплине: экзамен.

Изучение дисциплины заканчивается экзаменом. Экзамен проводится в соответствии с Положением о порядке организации и проведения промежуточной аттестации обучающихся, расположенном на официальном сайте филиала:

http://sbmpei.ru/files/uplfiles/06_Polojenie_o_poryadke_organizatsii_i_provedeniya_promejutochnoy_attestatsii_obuchayuschihся_2017.pdf

5 Самостоятельная работа студента

Для обеспечения самостоятельной работы разработаны:

1. Конспект лекций по дисциплине (см. приложение 3.РПД Б1.В.ДВ.8.1 (лк));
2. Методические указания к выполнению лабораторных работ (см. приложение 3.РПД Б1.В.ДВ.8.1 (лб));
3. Методические указания к самостоятельной работе студентов (см. приложение 3.РПД Б1.В.ДВ.8.1 (срс)).
4. Методические рекомендации к расчетно-графической работе (см. приложение 3.РПД Б1.В.ДВ.8.1 (ргр))

6 Фонд оценочных средств для проведения промежуточной аттестации обучающихся по дисциплине

6.1 Перечень компетенций с указанием этапов их формирования

При освоении дисциплины формируются следующие компетенции:

- общепрофессиональные ОПК-1, ОПК-2, ОПК-3;
- профессиональные ПК-1, ПК-2.

Указанные компетенции формируются в соответствии со следующими этапами:

1. Формирование и развитие теоретических знаний, предусмотренных указанными компетенциями (лекционные занятия, самостоятельная работа студентов).
2. Приобретение и развитие практических умений, предусмотренных компетенциями (практические занятия, лабораторные работы, выполнение расчетно-графической работы, самостоятельная работа студентов).

3. Закрепление теоретических знаний, умений и практических навыков, предусмотренных компетенциями, в ходе защит лабораторных работ, выполнения расчетно-графической работы, а также решения конкретных технических задач на практических занятиях, успешной сдачи экзамена.

6.2 Описание показателей и критериев оценивания компетенций на различных этапах их формирования, описания шкал оценивания

Формы текущего контроля по темам дисциплины

№№пп	Наименование темы дисциплины	Формы текущего контроля
1.	Тема 1	Тест
2.	Тема 2	Тест, Контрольная работа
3.	Тема 3	Тест
4.	Тема 4	Контрольная работа
5.	Тема 5	Тест
6.	Тема 6	Контрольная работа
7.	Тема 7	Контрольная работа
8.	Тема 8	Тест

Виды контроля самостоятельной работы студентов и оценочные средства

№ п/п	№ семестра	Тема учебной дисциплины	Виды контроля	Оценочные средства
1	7	Тема 1, 2, 3, 5, 8	Тест	«50%»- Пороговый уровень освоения компетенции «70%»- Продвинутый уровень освоения компетенции «90%»- Высокий уровень освоения компетенции
2	7	Тема 2,4,6,7	Контрольная работа	«3»- Пороговый уровень освоения компетенции «4»- Продвинутый уровень освоения компетенции «5»- Высокий уровень освоения компетенции

Образовательные технологии, обеспечивающие результаты освоения дисциплины в форме компетенций

Код компетенции	Компонентный состав компетенции (дескрипторы)	Технологии формирования	Средства оценки
ОПК-1	Знать: типовые приемы проектирования, инструментарий для документирования проектных решений, основы работы в среде Java Eclipse.	лекции, лабораторные занятия	Опрос, контрольные работы
	Уметь: разрабатывать приложения в среде Java Eclipse, создавать проектную документацию	лабораторные занятия	Опрос, контрольные работы
	Владеть: навыками работы в среде программирования Java, типовыми приемами проек-	лабораторные занятия	Опрос, тест

	тирования, инструментарием для документирования проектных решений и навыками работы с CASE-средствами для разработки UML-диаграмм.		
ОПК-2	Знать: современные технологии разработки программных комплексов	лекции, лабораторные занятия	Опрос, контрольные работы
	Уметь: создавать сетевые приложения на сокетах в среде Java Eclipse. Уметь создавать Web-приложения	лабораторные занятия	Опрос, тест
	Владеть: имеет навыки применения программных технологий для создания сетевых приложений и WEB-приложений	лабораторные занятия	Опрос, тест
ОПК-4	Знать: основы работы с паттернами проектирования, классификацию паттернов проектирования	лекции, лабораторные занятия	Опрос, контрольные работы
	Уметь: применять паттерны проектирования в среде Java Eclipse для решения практических задач	лабораторные занятия	Опрос, тест
	Владеть: навыками использования паттернов проектирования в среде Java Eclipse для решения практических задач	лабораторные занятия	Опрос, тест
ПК-1	Знать: знать особенности разработки программного обеспечения, ориентированного на повторное использование программного кода	лекции, лабораторные занятия	Опрос, тест
	Уметь: использовать паттерны проектирования при разработке моделей компонентов информационных систем	лабораторные занятия	Опрос, контрольные работы
	Владеть: навыками использования паттернов проектирования при разработке моделей компонентов информационных систем	лабораторные занятия	Опрос, тест
ПК-2	Знать: методы объектно-ориентированного тестирования, методы оценки качества разработанных приложений с использованием метрик объектно-ориентированных программных систем	лекции, лабораторные занятия	Опрос, контрольные работы
	Уметь: выполнять тестирование и оценку качества объектно-ориентированных приложений	лабораторные занятия	Опрос, контрольные работы
	Владеть: методами тестирования объектно-ориентированных программных систем. Способен оценивать качество разработанных приложений с использованием метрик объектно-ориентированных программных систем	лабораторные занятия	Опрос, контрольные работы

**Оценка уровней сформированности компетенций в результате освоения
учебной дисциплины**

Коды компетенций	Уровни сформированности компетенции	Основные признаки уровня
Общепрофессиональные компетенции - ОПК		
ОПК-1	Пороговый уровень освоения компетенции	Знает: типовые приемы проектирования, инструментов для документирования проектных решений, основы работы в среде Java Eclipse.
	Продвинутый уровень освоения компетенции	Дополнительно умеет: разрабатывать приложения в среде Java Eclipse, создавать проектную документацию
	Высокий уровень освоения компетенции	Дополнительно владеет: навыками работы в среде программирования Java, типовыми приемами проектирования, инструментарием для документирования проектных решений и навыками работы с CASE-средствами для разработки UML-диаграмм.
ОПК-12	Пороговый уровень освоения компетенции	Знает: современные технологии разработки программных комплексов
	Продвинутый уровень освоения компетенции	Дополнительно умеет: создавать сетевые приложения на сокетах в среде Java Eclipse. Уметь создавать Web-приложения
	Высокий уровень освоения компетенции	Дополнительно владеет: имеет навыки применения программных технологий для создания сетевых приложений и WEB-приложений
ОПК-4	Пороговый уровень освоения компетенции	Знает: основы работы с паттернами проектирования, классификацию паттернов проектирования
	Продвинутый уровень освоения компетенции	Дополнительно умеет: применять паттерны проектирования в среде Java Eclipse для решения практических задач
	Высокий уровень освоения компетенции	Дополнительно владеет: навыками использования паттернов проектирования в среде Java Eclipse для решения практических задач
Профессиональные компетенции - ПК		
ПК-1	Пороговый уровень освоения компетенции	Знает: особенности разработки программного обеспечения, ориентированного на повторное использование программного кода
	Продвинутый уровень освоения компетенции	Дополнительно умеет: использовать паттерны проектирования при разработке моделей компонентов информационных систем
	Высокий уровень освоения компетенции	Дополнительно владеет: навыками использования паттернов проектирования при разработке моделей компонентов информационных систем
ПК-2	Пороговый уровень освоения компетенции	Знает: методы объектно-ориентированного тестирования, методы оценки качества разработанных приложений с использованием метрик объектно-ориентированных программных систем

	Продвинутый уровень освоения компетенции	Дополнительно умеет: выполнять тестирование и оценку качества объектно-ориентированных приложений
	Высокий уровень освоения компетенции	Дополнительно владеет: методами тестирования объектно-ориентированных программных систем. Способен оценивать качество разработанных приложений с использованием метрик объектно-ориентированных программных систем

6.3 Типовые контрольные задания или иные материалы, необходимые для оценки знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций в процессе освоения образовательной программы

Вопросы к экзамену

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. Способы тестирования взаимодействия классов. Стохастическое тестирование
31. Способы тестирования взаимодействия классов. Тестирование разбиений и тестирование на основе состояний.
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. Базовые паттерны проектирования . Паттерн делегирования (Delegation)
42. Базовые паттерны проектирования . Паттерн интерфейс (Interface)
43. Базовые паттерны проектирования . Паттерн Заместитель (Proxy)
44. Порождающие паттерны проектирования. Паттерн Абстрактная фабрика (Abstract Factory)
45. Порождающие паттерны проектирования. Паттерн Метод фабрики (Factory Method)
46. Порождающие паттерны проектирования. Паттерн Строитель (Builder)
47. Порождающие паттерны проектирования. Паттерн Прототип (Prototype)
48. Порождающие паттерны проектирования. Паттерн Одиночка (Singleton)
49. Структурные паттерны проектирования. Паттерн Итератор (Iterator)
50. Структурные паттерны проектирования. Паттерн Адаптер (Adapter).
51. Структурные паттерны проектирования. Паттерн Мост (Bridge)
52. Структурные паттерны проектирования. Паттерн Декоратор (Decorator)
53. Структурные паттерны проектирования. Паттерн Управление кэшем (Cache Management)
54. Поведенческие паттерны проектирования. Паттерн Observer (Наблюдатель)
55. Поведенческие паттерны проектирования. Паттерн Цепочка обязанностей (Chain of Responsibility)
56. Поведенческие паттерны проектирования. Паттерн Состояние (State)
57. Поведенческие паттерны проектирования. Паттерн Стратегия (Strategy)
58. Поведенческие паттерны проектирования. Паттерн Посетитель (Visitor)

Примеры тестов для промежуточной аттестации

Тест по теме 1. Введение в объектно-ориентированный анализ и проектирование

- 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

Тест по теме 2. Унифицированный язык моделирования UML

- 1) Объектно-ориентированные технологии имеют следующие преимущества:
- a) Уменьшенную связность между модулями
 - b) Повышенное качество кода
 - c) Высокий уровень абстракции
 - d) Все из вышеперечисленного
- 2) В Объектно-ориентированной Технологии слово “UML” означает
- a) Unified Module Language
 - b) Unified Modeling Language
 - c) Universal Module Leveling
 - d) Universal Module Language
- 3) Объект имеет
- a) Поведение
 - b) Состояние
 - c) Атрибуты
 - d) Все из вышеперечисленного
- 4) 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
- 5) 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
- 6) 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
- 7) Какое из следующих утверждений правильное:
- a) 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 В зависит от пакета А, то любые изменения в А вызывают изменения в пакете В.
 - d) Диаграммы объектов и диаграммы классов взаимозаменяемы
- 8) 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
- 9) 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
- 10) 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
- 11) Машина имеет 4 колеса. Какой тип отношений имеют класс “Машина” и класс “Колесо”?
- a) Ассоциация
 - b) Агрегация
 - c) Наследование
 - d) Реализация
- 12) 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
- 13) 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
- 14) 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.
- 15) 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.
- 16) 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
- 17) 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

- 18) Какое из следующих утверждений является неверным?
- В какой-либо момент времени объект может быть более чем в одном состоянии
 - Автомат может иметь множество конечных состояний
 - Автомат может запоминать, какое подсостояние было последним состоянием
 - Переход может срабатывать условно
- 19) Какое из следующих утверждений о диаграмме состояний является правильным:
- Все действия на диаграмме состояний связаны с переходами
 - Событие может явиться причиной того, что объект останется в том же состоянии, предшествующем событию
 - Если объект выходит из состояния, он не может вернуться в это состояние.
 - Два различных перехода из одного состояния могут перекрывать друг друга (они могут быть вызваны одним и тем же событием)
- 20) Названия прецедентов должны начинаться:
- с глагола или существительного
 - с глагола
 - с прилагательного
 - с глагола или прилагательного

Тест по теме 3. Основы объектно-ориентированного тестирования программных систем

- 1) Что такое CRC-карта?
- Список всех UML-диаграмм, применяемых при проектировании объектно-ориентированных приложений
 - Карточка, которая помогает установить задачи класса и выявить его окружение
 - Карточка, которая содержит список всех методов класса и способ его тестирования
- 2) Особенности тестирования объектно-ориентированных модулей.
- Расширение области тестирования
 - Изменение методики тестирования
 - Учет особенностей ООП при проектировании тестовых вариантов
 - Все вышеперечисленное
- 3) В чем состоит суть методики тестирования интеграции объектно-ориентированных систем, основанной на потоках?
- Объектом интеграции является набор классов, обслуживающих единичный ввод данных в систему
 - Вначале тестируются независимые классы, затем работают с первым слоем зависимых классов и т.д.
 - Определяется кластер сотрудничающих классов, строятся тестовые варианты, цель которых обнаружение ошибок сотрудничества
 - Разработчик выдвигает гипотезу о предполагаемых ошибках, для проверки которой разрабатываются тестовые варианты
 - Исходные данные для тестовых вариантов генерируются случайным образом
- 4) Поясните содержание методики тестирования интеграции объектно-ориентированных систем, основанной на использовании.
- Объектом интеграции является набор классов, обслуживающих единичный ввод данных в систему
 - Вначале тестируются независимые классы, затем работают с первым слоем зависимых классов и т.д.
 - Определяется кластер сотрудничающих классов, строятся тестовые варианты, цель которых обнаружение ошибок сотрудничества
 - Разработчик выдвигает гипотезу о предполагаемых ошибках, для проверки которой разрабатываются тестовые варианты

- e) Исходные данные для тестовых вариантов генерируются случайным образом
- 5) В чем заключается тестирование, основанное на ошибках?
 - a) Объектом интеграции является набор классов, обслуживающих единичный ввод данных в систему
 - b) Вначале тестируются независимые классы, затем работают с первым слоем зависимых классов и т.д.
 - c) Определяется кластер сотрудничающих классов, строятся тестовые варианты, цель которых обнаружение ошибок сотрудничества
 - d) Разработчик выдвигает гипотезу о предполагаемых ошибках, для проверки которой разрабатываются тестовые варианты
 - e) Исходные данные для тестовых вариантов генерируются случайным образом
- 6) В чем состоит стохастическое тестирование класса?
 - a) Объектом интеграции является набор классов, обслуживающих единичный ввод данных в систему
 - b) Вначале тестируются независимые классы, затем работают с первым слоем зависимых классов и т.д.
 - c) Определяется кластер сотрудничающих классов, строятся тестовые варианты, цель которых обнаружение ошибок сотрудничества
 - d) Разработчик выдвигает гипотезу о предполагаемых ошибках, для проверки которой разрабатываются тестовые варианты
 - e) Исходные данные для тестовых вариантов генерируются случайным образом
- 7) Методы тестирования разбиений на уровне классов.
 - a) Разбиение на категории по состояниям
 - b) Разбиение на категории по свойствам
 - c) Разбиение на категории по функциональности
 - d) Все вышеперечисленные методы
- 8) Перечислите методы тестирования взаимодействия классов.
 - a) Стохастическое тестирование
 - b) Тестирование разбиений
 - c) Тестирование на основе сценариев
 - d) Тестирование на основе состояний
 - e) Все вышеперечисленные методы

Тест по теме 5. Введение в паттерны проектирования. Базовые паттерны проектирования

- 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)

Тест по теме 8. Поведенческие паттерны

- 1) 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, логировать, а также поддерживать отмену операций
 - c) Не нарушая инкапсуляцию, определяет и сохраняет внутреннее состояние объекта и позволяет позже восстановить объект в этом состоянии
- 2) 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
- 3) Назначение паттерна Команда (Command)
- a) Инкапсулирует запрос в виде объекта, позволяя передавать его клиентам в качестве параметров, ставить в очередь, логировать, 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
- 4) 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
- 5) 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
- 6) 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
- 7) 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
- 8) 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
- 9) 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, логировать, а также поддерживать отмену операций

Примеры контрольных работ

Контрольная работа по теме 2. Унифицированный язык моделирования UML

1. Виды диаграмм UML
2. Диаграммы вариантов использования. Описание вариантов использования. Виды отношений между вариантами использования
3. Диаграмма классов. Построение концептуальной модели предметной области.
4. Диаграмма классов. Имя класса, атрибуты класса, операции.
5. Диаграмма классов.
6. Отношения между классами
7. Объекты. Параметризованные классы
8. Диаграмма деятельности. Состояние действия. Переходы. Дорожки.
9. Диаграмма последовательностей. Линия жизни объекта. Фокус управления.
10. Диаграмма последовательностей.
11. Сообщения. Ветвление потока управления. Стереотипы сообщений. Временные ограничения на диаграммах последовательностей.
12. Диаграмма кооперации. Объекты. Составные объекты. Связи. Сообщения.
13. Диаграмма пакетов. Системные пакеты.
14. Диаграммы состояний объекта. Имя состояния, список внутренних действий, начальное состояние, конечное состояние.
15. Диаграммы состояний объекта. Переход. Событие. Сторожевое условие. Выражение действия. Составное состояние и подсостояние.
16. Диаграммы состояний объекта.
17. Диаграммы состояний объекта. Последовательные подсостояния. Параллельные подсостояния. Историческое состояние. Сложные переходы.
18. Диаграмма компонентов. Имя компонента. Виды компонентов. Интерфейсы. Зависимости.
19. Диаграмма размещения. Узел. Соединения.

Контрольная работа по теме 4. Оценка характеристик программ на основе объектно-ориентированных метрик

1. Метрики Мартина. Рекомендуемые значения метрик
2. Метрики Чидамбера-Кемерера. Рекомендуемые значения метрик
3. Метрики Лоренца-Кидда. Рекомендуемые значения метрик
4. Метрики Абреу. Рекомендуемые значения метрик

Контрольная работа по теме 6. Порождающие паттерны

1. Из каких частей состоит описание паттерна
2. В каких случаях применяется паттерн Абстрактная фабрика (Abstract Factory)
3. В каких случаях применяется паттерн Метод фабрики (Factory Method)
4. В каких случаях применяется паттерн Строитель (Builder)
5. В каких случаях применяется паттерн Прототип (Prototype)
6. В каких случаях применяется паттерн Паттерн Одиночка (Singleton)
7. В каких случаях применяется паттерн Паттерн Пул объектов (Object Pool)

Контрольная работа по теме 7. Структурные паттерны

1. В каких случаях применяется паттерн Итератор (Iterator)
2. В каких случаях применяется паттерн Адаптер (Adapter).
3. В каких случаях применяется паттерн Мост (Bridge)
4. В каких случаях применяется паттерн Приспособленец (Ayweight)
5. В каких случаях применяется паттерн Динамическая компоновка (Dynamic Linkage)
6. В каких случаях применяется паттерн Виртуальный заместитель (Virtual Proxy)
7. В каких случаях применяется паттерн Декоратор (Decorator)

8. В каких случаях применяется паттерн Управление кэшем (Cache Management)

6.4 Методические материалы, определяющие процедуры оценивания знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций

Процедуры оценивания знаний, умений, навыков, характеризующих этапы формирования компетенций, изложены в:

1. Конспект лекций по дисциплине (см. приложение 3.РПД Б1.В.ДВ.8.1 (лк));
2. Методические указания к выполнению лабораторных работ (см. приложение 3.РПД Б1.В.ДВ.8.1 (лб));
3. Методические указания к самостоятельной работе студентов (см. приложение 3.РПД Б1.В.ДВ.8.1 (срс)).
4. Методические рекомендации к расчетно-графической работе (см. приложение 3.РПД Б1.В.ДВ.8.1(ргр))

7 Перечень основной и дополнительной учебной литературы, необходимой для освоения дисциплины

Основная учебная литература

1. Кузнецов, А.А. Технология объектно-ориентированного программирования [Электронный ресурс] : учебное пособие / А.А. Кузнецов, М.Ю. Михеев. — Электрон. дан. — Пенза : ПензГТУ, 2011. — 70 с. — Режим доступа: <https://e.lanbook.com/book/62736>. — Загл. с экрана.
2. Малашенкова И.В.. Паттерны проектирования в Java/, И.В. Малашенкова, Е.А. Панкратова. – Смоленск: филиал ГБОУ ВО «Национальный исследовательский университет «МЭИ (ТУ)» в г.Смоленске, 2017. – 120 с.
3. Панкратова Е.А. Проектирование программного обеспечения.[Текст]: методические рекомендации/ Е.А. Панкратова, О.В. Семенова, В.В. Малахов - Смоленск: РИО филиала ГОУВПО «МЭИ(ТУ)» в г. Смоленске, 2010 . – 36 с.
4. 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.

Дополнительная учебная литература

1. Орлов С.А. Технологии разработки программного обеспечения: Учебник /С.Орлов –СПб. Питер, 2012. -464 с.: ил.
2. Черников Б.В. Оценка качества программного обеспечения: Практикум; учебное пособие / Б.В. Черников, Б.Е. Поклонов /Под ред. Б.В. Черникова –М: ИД «ФОРУМ»: ИНФРА-М, 2012, -400 с.: ил.(Высшее образование)

8 Перечень ресурсов информационно-телекоммуникационной сети «Интернет» необходимых для освоения дисциплины

1. <http://www.interface.ru/home.asp?artId=1602>
2. <http://study-java.ru>
3. <http://altcode.ru/java/>

9 Методические указания для обучающихся по освоению дисциплины

Дисциплина предусматривает лекции раз в неделю, и лабораторные работы раз в две недели. Изучение курса завершается экзаменом.

Успешное изучение курса требует посещения лекций, активной работы на лабораторных работах, выполнения всех учебных заданий преподавателя, ознакомления с основной и дополнительной литературой.

Во время **лекции** студент должен вести краткий конспект.

Работа с конспектом лекций предполагает просмотр конспекта в тот же день после занятий. При этом необходимо пометить материалы конспекта, которые вызывают затруднения для понимания. При этом обучающийся должен стараться найти ответы на затруднительные вопросы, используя рекомендуемую литературу. Если ему самостоятельно не удалось разобраться в материале, необходимо сформулировать вопросы и обратиться за помощью к преподавателю на консультации или на ближайшей лекции.

Обучающемуся необходимо регулярно отводить время для повторения пройденного материала, проверяя свои знания, умения и навыки по контрольным вопросам.

Лабораторные работы составляют важную часть профессиональной подготовки студентов. Они направлены на экспериментальное подтверждение теоретических положений и формирование учебных и профессиональных практических умений.

Выполнение студентами лабораторных работ направлено на:

обобщение, систематизацию, углубление, закрепление полученных теоретических знаний по конкретным темам дисциплин;

формирование необходимых профессиональных умений и навыков;

Содержание лабораторных работ фиксируется в РПД в разделе 4 настоящей программы.

При планировании лабораторных работ следует учитывать, что наряду с ведущей целью - подтверждением теоретических положений - в ходе выполнения заданий у студентов формируются практические умения и навыки обращения с лабораторным оборудованием, аппаратурой и пр., которые могут составлять часть профессиональной практической подготовки, а также исследовательские умения (наблюдать, сравнивать, анализировать, устанавливать зависимости, делать выводы и обобщения, самостоятельно вести исследование, оформлять результаты).

Состав заданий для лабораторной работы +спланирован с таким расчетом, чтобы за отведенное время они могли быть качественно выполнены большинством студентов.

Выполнению лабораторных работ предшествует проверка знаний студентов – их теоретической готовности к выполнению задания.

Помимо собственно выполнения работы для каждой лабораторной работы предусмотрена процедура защиты, в ходе которой преподаватель проводит устный или письменный опрос студентов для контроля понимания выполненных ими измерений, правильной интерпретации полученных результатов и усвоения ими основных теоретических и практических знаний по теме занятия.

При подготовке к **экзамену** в дополнение к изучению конспектов лекций и учебных пособий, необходимо пользоваться учебной литературой, рекомендованной к настоящей программе. При подготовке к экзамену нужно изучить теорию: определения всех понятий и подходы к оцениванию до состояния понимания материала и самостоятельно решить по несколько типовых задач из каждой темы. При решении задач всегда необходимо уметь качественно интерпретировать итог решения.

Самостоятельная работа студентов (СРС) по дисциплине играет важную роль в ходе всего учебного процесса. Методические материалы и рекомендации для обеспечения СРС готовятся преподавателем и являются неотъемлемой частью программы.

10 Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине, включая перечень программного обеспечения и информационных справочных систем (при необходимости)

При проведении **лабораторных работ** предусматривается использование персональных компьютеров, оснащенных необходимым комплектом свободно распространяемого программного обеспечения. – Java Eclipse 8.

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

Лекционные занятия:

Аудитория.

Лабораторные работы по данной дисциплине проводятся в компьютерных классах, оснащенных необходимым комплектом программного обеспечения.

Автор
канд. техн. наук, доцент

Е.А. Панкратова

Зав. кафедрой ВТ
д-р техн. наук, профессор

А.С. Федулов

Программа одобрена на заседании кафедры ВТ 21 ноября 2018 года, протокол № 03.