

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ
Федеральное государственное бюджетное образовательное учреждение высшего
образования

«ДАГЕСТАНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

Факультет математики и компьютерных наук

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

Объектно-ориентированные CASE- технологии

Кафедра дискретной математики и информатики

факультета математики и компьютерных наук

Образовательная программа

02.04.02 Фундаментальная информатика и информационные технологии

Профиль подготовки

Информационные технологии

Уровень высшего образования

магистратура

Форма обучения

очно-заочная

Статус дисциплины: входит в обязательную часть ОПОП

Махачкала, 2021

Рабочая программа дисциплины «Объектно-ориентированные CASE-технологии» составлена в 2021 году в соответствии с требованиями ФГОС ВО по направлению подготовки 02.04.02 Фундаментальная информатика и информационные технологии (уровень магистратуры) от «27» августа 2017 г. № 811.

Разработчик(и): кафедра дискретной математики и информатики, Раджабова Наима Шамильевна, к.ф.-м.н., доцент.

Рабочая программа дисциплины одобрена:

на заседании кафедры дискретной математики и информатики от «30» 05. 2021 г., протокол № 9 ;

зав. кафедрой МБ Магомедов А. М.
(подпись)

и

на заседании Методической комиссии факультета математики и компьютерных наук от

«23» июня 2021 г., протокол № 6 ;
председатель В.Д. Бейбалаев
(подпись)

Рабочая программа дисциплины согласована с учебно-методическим управлением

«04» 07 2021 г.
Начальник УМУ Гасангаджиева А. Г.
(подпись)

1. Цели освоения дисциплины

Целями освоения дисциплины «Объектно-ориентированные CASE-технологии» являются:

- получение знаний по основам организации процесса анализа и проектирования программного обеспечения с использованием языка моделирования UML;
- получение практических навыков использования языка UML при проектировании и анализе программных систем, а также при реализации языка UML в системах программирования.

Задачи курса:

- ознакомить с методологическими основами современных объектно-ориентированных CASE-систем;
- дать систематические знания о языке UML;
- ознакомить с наиболее широко используемыми CASE-системами;
- научить применять объектно-ориентированные CASE-технологии на практике.

2. Место дисциплины в структуре ОПОП магистратуры

Дисциплина «Объектно-ориентированные CASE-технологии» относится к обязательной части образовательной программы магистратуры по направлению 02.04.02 Фундаментальная информатика и информационные технологии и преподается на 1 курсе в 1 семестре (5 зачетных единиц). Изучение предмета завершается письменным экзаменом в конце семестра.

Дисциплина «Объектно-ориентированные CASE-технологии» логически и содержательно взаимосвязана с дисциплинами, в которых рассматриваются вопросы проектирования сложных программных систем.

Для освоения данной дисциплины необходимо знание основ одного из объектно-ориентированных языков программирования, умение проектировать классы, приобретенные на занятиях по Языкам и методам программирования и Java-программированию.

Результаты освоения данной дисциплины будут востребованы на занятиях по «Программированию в .NET», «Распределенным объектным технологиям», «Анализу информационных систем» и при подготовке выпускной квалификационной работы..

3. Компетенции обучающегося, формируемые в результате освоения дисциплины (перечень планируемых результатов обучения).

Код и наименование компетенции из ОПОП	Код и наименование индикатора достижения компетенций (в соответствии с ОПОП)	Планируемые результаты обучения	Процедура освоения
<p>УК-2 Способен управлять проектом на всех этапах его жизненного цикла.</p>	<p>УК-2.1. Знает необходимые для осуществления профессиональной деятельности правовые нормы. УК-2.2. Умеет определять круг задач в рамках избранных видов профессиональной деятельности, планировать собственную деятельность исходя из имеющихся ресурсов; соотносить главное и второстепенное, решать поставленные задачи в рамках избранных видов профессиональной деятельности. УК-2.3. Имеет практический опыт применения нормативной базы и решения задач в области избранных видов профессиональной деятельности</p>	<p>Знает: международные и российские стандарты жизненного цикла ПО. Умеет: разрабатывать техническое задание по заданному шаблону для ВКР. Владеет: навыками объектно-ориентированного моделирования на языке UML для моделирования для решения задач в различных предметных областях.</p>	<p>Конспектирование лекций, участие в дискуссиях. Разработка и реализация проектов на лабораторных занятиях.</p>

<p>ОПК-5. Способен устанавливать и сопровождать программное обеспечение информационных систем, осуществлять эффективное управление разработкой программных средств и проектов.</p>	<p>ОПК-5.1. Знает методику установки и администрирования информационных систем и баз данных. Знаком с перечнем ПО, входящим в Единый реестр российских программ</p> <p>ОПК-5.2. Умеет реализовывать техническое сопровождение информационных систем и баз данных.</p> <p>ОПК-5.3. Имеет практические навыки установки и инсталляции программных комплексов.</p>	<p>Знает: международные и российские стандарты жизненного цикла ПО.</p> <p>Умеет:</p> <ul style="list-style-type: none"> – разрабатывать техническое задание (SRS) по заданному шаблону для ВКР. – реализовывать техническое сопровождение информационных систем и баз данных. <p>Владеет: навыками объектно-ориентированного моделирования на языке UML для моделирования для решения задач в различных предметных областях.</p>	<p>Конспектирование лекций, участие в дискуссиях. Разработка и реализация проектов на лабораторных занятиях.</p>
<p>ПК-3. Способность проектировать распределенные информационные системы, их компоненты и протоколы их взаимодействия, собирать, обрабатывать и интерпретировать экспериментальные данные, необходимые для проектной и</p>	<p>ПК-3.1. Знает основы проектирования и элементы архитектурных решений информационных систем.</p> <p>ПК-3.2. Умеет применять в практической деятельности профессиональные стандарты в</p>	<p>Знает:</p> <ul style="list-style-type: none"> – основы проектирования информационных систем на основе международного стандарта UML. – графическую нотацию языка UML и 	<p>Конспектирование лекций, участие в дискуссиях. Разработка, реализация и защита проектов.</p>

<p>производственно-технологической деятельности; способность к разработке новых алгоритмических, методических и технологических решений в конкретной сфере профессиональной деятельности.</p>	<p>области информационных технологий. ПК-3.3. Имеет практический опыт составления технического задания на разработку информационной системы.</p>	<p>классы метамодели языка UML.</p> <p>Умеет:</p> <p>отображать графическую нотацию языка UML в программы на языках программирования Delphi, C#, Java, C++ (по выбору студента), а также представлять программы на языках программирования Delphi, C#, Java, C++ (по выбору студента) с помощью графической нотации UML.</p> <p>Владеет:</p> <p>навыками использования языка UML с помощью CASE-инструментов при проектировании программного обеспечения.</p>	
---	--	---	--

4. Объем, структура и содержание дисциплины.

4.1. Объем дисциплины составляет 5 зачетных единиц, 180 академических часов.

4.2. Структура дисциплины.

4.2.1 Структура дисциплины в очно-заочной форме

№ п/п	Разделы и темы дисциплины	Семестр	Виды учебной работы, включая самостоятельную работу студентов и трудоемкость (в часах)					Формы текущего контроля успеваемости и промежуточной аттестации	
			Неделя семестра Лекции	Практические занятия	Лабораторные занятия	Самостоятельная работа	Контроль самостоятельной работы		
Модуль 1. Объектно-ориентированный анализ и проектирование									
1.	Стандарты для объектно-ориентированного анализа и проектирования программного обеспечения	1	1	2		2	14		Прием лабораторных работ (ЛР) и реферата (Р)
2	Моделирование программных систем с помощью языка UML.	1	2	2		2	14		ЛР, Р
	Итого по модулю 1:	36		4		4	28		Коллоквиум
Модуль 2. Структурные диаграммы									
3	Средства нотации языка UML для описания статической структуры	1	3	2		2	14		ЛР, Р

	модели системы.								
4.	Средства нотации языка UML для описания сценариев использования моделируемой системы.	1	4			2	16		ЛР, Р
	Итого по модулю 2:			2		4	30		Коллоквиум
Модуль 3. Диаграммы взаимодействия									
5.	Средства нотации языка UML используемые для описания поведения моделируемой системы	1	5	2		2	14		ЛР, Р
6.	Диаграммы взаимодействия программных систем			2		2	14		ЛР, Р
	Итого по модулю 3:			4		4	28		Коллоквиум
Модуль 4. Структура метамодели языка UML									
6.	Моделирование отношения "владелец-собственность".	1	6	1		1	16		ЛР, Р
7.	Моделирование пространств имен с помощью классов	1	7	1		1	16		ЛР, Р

	Namespace и NamedElement							
	Итого по модулю 4:			2		2	32	Коллоквиум
8.	Подготовка к экзамену	1	8				36	Модуль 5
	ИТОГО:			12		14	154	Экзамен

4.3. Содержание дисциплины, структурированное по темам (разделам)

4.3.1 Содержание лекционных занятий по дисциплине

Модуль 1. Объектно-ориентированный анализ и проектирование

Тема 1. Разработка стандартов для объектно-ориентированного анализа и проектирования программного обеспечения.

1. Жизненный цикл программного обеспечения.
2. История возникновения и развития языка UML.
3. Структура стандартов на язык UML.

Тема 2. Моделирование программных систем с помощью языка UML

1. Графическая нотация UML. Мета модель языка UML.
2. Обзор основных понятий графической нотации и семантики языка UML.
3. Unified Modeling Language (UML) - унифицированный язык моделирования.

Модуль 2. Структурные диаграммы

Тема 3. Средства нотации языка UML для описания статической структуры модели системы

1. Классификаторы на диаграмме статической структуры.
 1. Представление атрибутов и операций классификаторов.
 2. Свойства классификаторов, свойства атрибутов и операций.
 3. Типы данных языка UML.

4. Представление обозначений нотации языка UML для классификаторов с помощью конструкций языков C++, C#, Java, Delphi

Тема 4. Средства нотации языка UML для описания сценариев использования моделируемой системы

1. Диаграммы прецедентов (Use Case diagram)
2. Средства языка UML для детализации поведения системы, описанного на диаграммах сценариев использования.

Модуль 3. Диаграммы взаимодействия

Тема 5. Средства нотации языка UML используемые для описания поведения моделируемой системы.

Представление обозначений нотации языка UML на диаграммах коммуникации и на диаграммах последовательности взаимодействия с помощью конструкций языков программирования.

Тема 6. Диаграммы взаимодействия.

1. Диаграммы последовательности взаимодействия (Sequence diagram)
2. Диаграммы коммуникации объектов (Communication diagram)
3. Диаграмма активностей (Activity Diagram)
4. Диаграмма состояний (State Machine Diagram)

Модуль 4. Структура метамодели языка UML

Тема 6. Моделирование отношения "владелец-собственность" Мета модель языка UML 2.5.

1. Моделирование отношения "владелец-собственность" с помощью базового класса метамодели Element.
2. Моделирование направленных отношений с помощью класса метамодели DirectedRelationship.

Тема 7. Моделирование пространств имен с помощью классов Namespace и NamedElement

Моделирование пространств имен с помощью классов Namespace и NamedElement.

4.3.2 Содержание лабораторных занятий по дисциплине

Лабораторные занятия предусмотрены по всем темам модулей и их содержание совпадает с содержанием тем модулей. Целью каждой лабораторной работы является проектирование диаграмм UML в среде Rational Rose или Visual Studio Ultimate, указанных в соответствующей теме модуля.

Модуль 1. Объектно-ориентированный анализ и проектирование

Лабораторная работа 1. Разработка технического задания программного проекта

Лабораторная работа 2. Моделирование программных систем с помощью языка UML.

Графическая нотация UML. Мета модель языка UML.

Модуль 2. Структурные диаграммы

Лабораторная работа 3. Представление обозначений нотации языка UML для классификаторов с помощью конструкций языков C++, C#.

Лабораторная работа 4. Диаграммы прецедентов (Use Case diagram).

Модуль 3. Диаграммы взаимодействия

Лабораторная работа 5. Диаграммы коммуникации объектов (Communication diagram).
Диаграммы последовательности взаимодействия (Sequence diagram).

Лабораторная работа 6. Диаграмма классов. Диаграмма объектов. Диаграмма компонентов. Диаграмма композитной структуры. Диаграмма размещения. Диаграмма профиля.

Лабораторная работа 7. Диаграмма коммуникаций. Диаграмма последовательности взаимодействия. Диаграмма синхронизации. Диаграмма обзора взаимодействия. Диаграмма деятельности. Диаграмм состояний. Диаграмма вариантов использования.

Модуль 4. Структура метамодели языка UML

Сдача итогового проекта.

5. Образовательные технологии

Процесс изложения учебного материала сопровождается презентациями и демонстрацией решения задач в интерактивном режиме с использованием мультимедийного проектора. Предусмотрено регулярное общение с лектором и представителями российских и зарубежных компаний по электронной почте и по скайпу.

6. Учебно-методическое обеспечение самостоятельной работы студентов.

Самостоятельная работа студентов складывается из:

- проработки лекционного материала (настоятельно рекомендуется самостоятельное практическое решение всех разобранных на лекциях упражнений);
- изучения рекомендованной литературы и материалов соответствующих форумов интернет;
- подготовки к отчетам по лабораторным работам;
- подготовки к сдаче промежуточных форм контроля.

№	Вид самостоятельной работы	Вид контроля	Учебно-методическое обеспечение
1.	Проработка лекционного	Контрольный фронтальный опрос.	См. разделы 8, 9 данного документа

	материала.		
2.	Изучение рекомендованной литературы и материалов соответствующих форумов интернет.	Контрольный фронтальный опрос, прием и представление рефератов.	См. разделы 8, 9 данного документа
3.	Подготовка к отчетам по лабораторным работам.	Проверка выполнения работ, опрос по теме работы.	См. разделы 8, 9 данного документа
4.	Подготовка к сдаче промежуточных форм контроля.	Контрольные работы по каждому модулю и прием рефератов.	См. разделы 8, 9 данного документа

Пакет заданий для самостоятельной работы выдается в начале семестра, определяются предельные сроки их выполнения и сдачи.

7. Фонд оценочных средств для проведения текущего контроля успеваемости, промежуточной аттестации по итогам освоения дисциплины.

7.1 Типовые контрольные задания

7.1.1 Вопросы для самостоятельной работы

1. Что такое жизненный цикл программного обеспечения?
2. Что такое свободное ПО и в чём его отличия от бесплатного?
3. Какие артефакты возникают в результате процессов ЖЦПО?
4. Что такое модель жизненного цикла?
5. Каковы основные этапы, процессы, роли участников ЖЦПО?
6. Как связан объём разработки с выбором модели ЖЦПО?
7. Какими стандартами и соглашениями регламентируются ЖЦПО?
8. Где проходит граница между анализом и проектированием?
9. В чём особенности объектно-ориентированного подхода к анализу, проектированию и конструированию ПО?
10. Что такое быстрая разработка приложений?
11. Дайте общую характеристику процесса объектно-ориентированного проектирования.
12. Как выявляются необходимые сущности и связи между ними?

13. Что такое требование? Каковы его основные характеристики?
14. Дайте характеристику языку UML. Каковы его главные достоинства и недостатки?
15. Какие виды диаграмм специфицирует язык UML?
16. Какие виды отношений между объектами способен выразить язык UML?
17. Опишите назначение и правила составления UML-диаграмм.
18. Какие виды ассоциаций можно представить на языке UML?
19. С какого вида диаграмм нужно начинать описание модели?
20. Приведите пример тернарной ассоциации.
21. Какие программные средства используются для упрощения процессов объектно-ориентированного анализа, проектирования и конструирования?
22. Как объектно-ориентированный подход стыкуется с различными моделями данных при создании схем баз данных?
23. Что такое объектно-реляционное отображение?
24. Что такое обратный инжиниринг?
25. Охарактеризуйте отличия концептуального, логического и физического проектирования.
26. Каковы основные характеристики интегрированных сред разработки ПО?
27. В чём особенности многопользовательской работы со средствами поддержки ЖЦ ПО?
28. В чём заключается сложность поддержки ЖЦ гетерогенных программных систем?
29. Что такое аутсорсинг и аутстаффинг?
30. Что такое платформа? Чем платформа разработки отличается от платформы развёртывания?
31. Когда требуется формально описанные требования к ПО?
32. Что специфицирует техническое задание?
33. Как техническое задание связано с проведением приёмо-сдаточных испытаний?
34. Какие методы применяются при сборе требований?
35. Чем отличаются друг от друга интервью, тестирование и анкетирование?
36. Как связана методология проектирования с поддерживающими её программными средствами?
37. Перечислите основные возможности систем управления исходным кодом (версионного контроля).
38. Как правильно сравнивать изменения в исходный код?
39. В чём заключаются основные проблемы слияния версий?
40. Перечислите основные возможности систем профилирования.
41. Дайте обзор уровней оптимизации.
42. Когда нужно начинать оптимизацию, а когда нельзя её заниматься?

43. Что такое рефакторинг исходного кода?
44. Как формализуется надёжность программной системы?
45. В чём главная проблема функционального тестирования?
46. Что такое интеграционное тестирование?
47. Что является целью процесса развёртывания?
48. Чем отличаются понятия поддержки и сопровождения?
49. Что такое эволюция программного продукта?
50. Перечислите основные возможности систем построения дистрибутивов.
51. Как связаны процессы сборки и развёртывания?
52. Как повышается сложность процессов сборки с ростом числа версий продуктов?

7.1.2 Контрольные вопросы

1. Жизненный цикл программного обеспечения (для самостоятельного изучения).
2. История возникновения и развития языка UML (для самостоятельного изучения).
3. Стандарты, связанные с языком UML (MOF, XMI, OCL).
4. Обзор основных понятий графической нотации и семантики языка UML. Мета модель языка UML.
5. Моделирование программных систем с помощью языка UML (для самостоятельного изучения).
6. Классификация UML-диаграмм (14 типов диаграмм, UML 2.5.x).
7. Классификаторы на диаграмме статической структуры.
8. Представление атрибутов и операций классификаторов.
9. Типы данных языка UML.
10. Свойства классификаторов.
11. Свойства атрибутов и операций.
12. Интерфейсы.
13. Стереотипы классификаторов.
14. Ограничения на UML-модели.
15. Видимость классификаторов, их атрибутов и операций.
16. Область действия атрибутов и операций класса.
17. Отношения обобщения на диаграмме классов.
18. Отношения реализации на диаграмме классов.
19. Отношение ассоциации на диаграмме классов.
20. Отношение ассоциации. Свойство агрегации и композиции.
21. Отношение ассоциации. Стереотипы окончания ассоциации.

22. Отношение зависимости на диаграмме классов. Категория зависимости связывание.
23. Отношение зависимости. Категория зависимости абстракция.
24. Отношение зависимости Категория зависимости использование.
25. Отношение зависимости. Категория зависимости разрешение.
26. Диаграмма коммуникации (Communication diagram). Изображение объектов, отношения связи между объектами, рассылка сообщений объектами.
27. Диаграмма последовательности взаимодействия (Sequence diagram). Назначение диаграммы. Элементы и связи.
28. Типы сообщений на диаграммах последовательности взаимодействия.
29. Диаграммы прецедентов (Use Case diagram). Средства нотации языка UML для описания сценариев использования моделируемой системы
30. Диаграммы взаимодействия, их элементы и связи между элементами.
31. Виды диаграмм взаимодействия и область их применения.
32. Диаграммы состояний, их элементы и связи. Область их применения.
33. Диаграммы деятельности, их элементы и связи. Область их применения.
34. Диаграммы компонентов и диаграммы размещения, их элементы и связи между элементами. Область их применения.
35. Механизмы расширения UML (стереотипы, метасвойства, ограничения, комментарии).
36. Диаграммы профиля, их элементы и связи. Область их применения.
37. Моделирование отношения "владелец-собственность"
38. Моделирование отношения "владелец-собственность" с помощью базового класса метамодели Element.
39. Моделирование направленных отношений с помощью класса метамодели DirectedRelationship.
40. Моделирование пространств имен с помощью классов Namespace и NamedElement
41. Моделирование пространств имен с помощью классов Namespace и NamedElement.
42. Моделирование импорта элементов модели в пространство имен с помощью классов PackageableElement, ElementImport, PackageImport.

7.1.3 Темы для рефератов

1. История разработки стандартов для объектно-ориентированного анализа и проектирования программного обеспечения.
2. История развития популярных графических нотаций для объектно-ориентированного анализа и проектирования.
3. Предметы и процесс стандартизации в группе по анализу и проектированию в OMG.

4. Жизненный цикл программного обеспечения.
5. История возникновения и развития языка UML.
6. Стандарты, связанные с языком UML (MOF, XMI, OCL).
7. Обзор основных понятий графической нотации и семантики языка UML. Мета модель языка UML.
8. Моделирование программных систем с помощью языка UML (для самостоятельного изучения).
9. Классификация UML-диаграмм (14 типов диаграмм, UML 2.5.x).
10. Моделирование импорта элементов модели в пространство имен с помощью классов PackageableElement, ElementImport, PackageImport.

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

Общий результат выводится как интегральная оценка, складывающаяся из текущего контроля - 50% и промежуточного контроля - 50%. Текущий контроль по дисциплине включает:

- посещение занятий – 10 баллов,
- подготовка реферата – 20 баллов,
- выполнение лабораторных заданий – 20 баллов,
- выполнение домашних (аудиторных) контрольных работ – 50 баллов.

Промежуточный контроль по дисциплине включает:

- прием лабораторных работ – 40 баллов,
- письменная контрольная работа – 60 баллов,

8. Перечень основной и дополнительной учебной литературы, необходимой для освоения дисциплины.

а) адрес сайта курса

<https://explorecoursesit.blogspot.com/>

б) основная литература:

1. Гагарина, Лариса Геннадьевна. Технология разработки программного обеспечения: [учеб. пособие] / Гагарина, Лариса Геннадьевна, Е. В. Кокорева; под ред. Л.Г.Гагариной. – М. : ФОРУМ: ИНФРА-М, 2009, 2008. - 399 с. - (Высшее образование). - Допущено УМО. - ISBN 978-5-8199-0342-1 (ИД "ФОРУМ"): 246-84.
2. Леоненков А.В. Объектно-ориентированный анализ и проектирование с использованием UML и IBM Rational Rose. Курс лекций [Электронный ресурс] : учебное пособие для студентов вузов, обучающихся по специальностям в области информационных технологий / А.В. Леоненков. — Электрон. текстовые данные. — Москва, Саратов: Интернет-Университет Информационных Технологий (ИНТУИТ), Вузовское образование, 2017. —

318 с. — 978-5-4487-0081-1. — Режим доступа: <http://www.iprbookshop.ru/67388.html> (дата обращения 19.04.2018).

3. Бабич А.В. Введение в UML [Электронный ресурс] / А.В. Бабич. — Электрон. текстовые данные. — М.: Интернет-Университет Информационных Технологий (ИНТУИТ), 2016. — 198 с. — 978-5-94774-878-9. — Режим доступа: <http://www.iprbookshop.ru/62809.html> (дата обращения 19.04.2018).

в) дополнительная литература:

4. Бабич, Александр Викторович. UML: Первое знакомство: пособие для подготовки к сдаче теста UMO-100 (OMG Certified UML Professional Fundamental): учеб. пособие / Бабич, Александр Викторович. - М.: Изд-во Интернет-Ун-та Информ. Технологий: БИНОМ. Лаб. знаний, 2008. - 175 с. + 1 электрон. опт. диск (CD-ROM). - (Основы информационных технологий). - ISBN 978-5-94774-878-9 (БИНОМ.ЛЗ): 190-00.

5. Самуйлов С.В. Объектно-ориентированное моделирование на основе UML [Электронный ресурс] : учебное пособие / С.В. Самуйлов. — Электрон. текстовые данные. — Саратов: Вузовское образование, 2016. — 37 с. — 2227-8397. — Режим доступа: <http://www.iprbookshop.ru/47277.html> (дата обращения 19.04.2018).

9. Перечень ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины.

Описание стандарта:

- 1) www.uml.org
- 2) www.uml-diagrams.org

Видеокурсы лекций:

- 1) <https://www.coursera.org/>
- 2) <https://www.udacity.com/>
- 3) <http://www.intuit.ru>

Форумы по компьютерным наукам и программированию:

- 1) www.stackoverflow.com
- 2) <http://www.cyberforum.ru/>

10. Методические указания для обучающихся по освоению дисциплины.

При решении лабораторных заданий программистский подход непременно должен присутствовать (без него решение не будет полноценным), однако, он не должен заслонять сугубо математические (доказательство и др.) и алгоритмические (построение, оптимизация, верификация и др.) аспекты.

Все упражнения, приведенные на лекции с решениями, следует прорабатывать сразу после лекции. Самостоятельная работа студентов складывается из:

- проработки лекционного материала (настоятельно рекомендуется самостоятельное практическое решение всех разобранных на лекциях упражнений);
- изучения рекомендованной литературы и материалов соответствующих форумов интернет;
- подготовки к отчетам по лабораторным работам;
- подготовки к сдаче промежуточных форм контроля (контрольных работ и сдаче реферата).

Пакет лабораторных заданий рассчитан на семестр. Рекомендуется выполнять и сдавать задания своевременно с прохождением соответствующего материала.

Модули и темы для самостоятельного изучения	Виды и содержание самостоятельной работы
Модуль 1. Объектно-ориентированный анализ и проектирование. Тема 1. Разработка стандартов для объектно-ориентированного анализа и проектирования программного обеспечения.	Проработка лекционного материала. Изучение рекомендованной литературы и материалов соответствующих форумов интернет для подготовки реферата.
Модуль 1. Объектно-ориентированный анализ и проектирование. Тема 2. Моделирование программных систем с помощью языка UML.	Проработка лекционного материала. Изучение рекомендованной литературы и материалов соответствующих форумов интернет для подготовки реферата.
Модуль 2. Графическая нотация языка UML. Тема 1. Средства нотации языка UML для описания статической структуры модели системы.	Проработка лекционного материала. Подготовка к отчетам по лабораторным работам. Подготовка к сдаче промежуточных форм контроля.
Модуль 2. Графическая нотация языка UML. Тема 2. Средства нотации языка UML для описания сценариев использования моделируемой	Проработка лекционного материала. Изучение рекомендованной литературы и материалов соответствующих форумов интернет для подготовки реферата. Подготовка

системы.	к отчетам по лабораторным работам
Модуль 3. Графическая нотация языка UML. Тема 3. Средства нотации языка UML, используемые для описания поведения моделируемой системы.	Проработка лекционного материала. Изучение рекомендованной литературы и материалов соответствующих форумов интернет для подготовки реферата. Подготовка к отчетам по лабораторным работам. Подготовка к сдаче промежуточных форм контроля.
Модуль 4. Структура метамодели языка UML. Тема 1. Моделирование отношения "владелец-собственность"	Проработка лекционного материала. Изучение рекомендованной литературы и материалов соответствующих форумов интернет для подготовки реферата. Подготовка к отчетам по лабораторным работам.
Модуль 4. Структура метамодели языка UML. Тема 2. Моделирование пространств имен с помощью классов Namespace и NamedElement	Проработка лекционного материала. Изучение рекомендованной литературы и материалов соответствующих форумов интернет для подготовки отчетов по лабораторным работам.

11. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине, включая перечень программного обеспечения и информационных справочных систем.

При осуществлении образовательного процесса студентами и профессорско-преподавательским составом используются следующее программное обеспечение: Microsoft Windows, Ubuntu Linux, Microsoft Visual Studio Ultimate, Rational Rose, Skype. Также студентам предоставляется доступ к российским и международным электронным библиотекам через компьютеры университета.

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

Имеется необходимая литература в библиотеке, медиапроектор и компьютер для проведения лекций-презентаций.

Лабораторные занятия проводятся в компьютерных классах с необходимым программным обеспечением.

Вся основная литература предоставляется студенту в электронном формате.

