

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ
Федеральное государственное бюджетное образовательное учреждение высшего образования
«ДАГЕСТАНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»
Факультет математики и компьютерных наук

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ
Введение в UML-технологии

Кафедра дискретной математики и информатики
факультета математики и компьютерных наук

Образовательная программа
02.03.02 Фундаментальная информатика и информационные технологии

Профиль подготовки
Информатика и компьютерные науки

Уровень высшего образования
бакалавриат

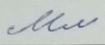
Форма обучения
очная

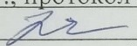
Статус дисциплины: входит в часть ОПОП,
формируемую участниками образовательных отношений

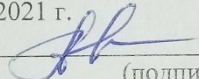
Махачкала, 2021

Рабочая программа дисциплины «Введение в UML-технологии» составлена в 2021 году в соответствии с требованиями ФГОС ВО по направлению подготовки 02.03.02 Фундаментальная информатика и информационные технологии (уровень бакалавриата) от «23» августа 2017 г. № 808.

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» мая 2021 г.,
протокол № 9;
зав. кафедрой  Магомедов А. М.
(подпись)

и
на заседании Методической комиссии факультета математики и компьютерных наук
от
«23» июня 2021 г., протокол № 6;
председатель  Бейбалаев В. Д.
(подпись)

Рабочая программа дисциплины согласована с учебно-методическим управлением
« 09 » 07 2021 г.
Начальник УМУ  Гасангаджиева А. Г.
(подпись)

Аннотация рабочей программы дисциплины

Дисциплина «Введение в UML-технологии» входит в часть, формируемую участниками образовательных отношений ОПОП бакалавриата по направлению подготовки 02.03.02 Фундаментальная информатика и информационные технологии.

Дисциплина реализуется на факультете математики и компьютерных наук кафедрой дискретной математики и информатики.

Содержание дисциплины охватывает круг вопросов, связанных с основами организации процесса анализа и проектирования программного обеспечения с использованием языка моделирования UML.

Дисциплина нацелена на формирование следующих компетенций выпускника: общепрофессиональной – ОПК-4, профессиональной – ПК-3.

Преподавание дисциплины предусматривает проведение следующих видов учебных занятий: лекции, практические занятия, лабораторные занятия.

Рабочая программа дисциплины предусматривает проведение следующих видов текущего контроля успеваемости в форме: приема лабораторных работ, подготовки и представления реферата, и промежуточного контроля – в форме зачета.

Объем дисциплины составляет 3 зачетные единицы – 108 часов, в том числе в академических часах по видам учебных занятий:

Очная форма обучения

Семестр	Учебные занятия						Форма промежуточной аттестации (зачет, дифференцированный зачет, экзамен)	
	в том числе:							
	Контактная работа обучающихся с преподавателем					СРС, в том числе экзамен		
	Всего	из них						
		Лекции	Лабораторные занятия	Практические занятия	КСР	консультации		
7	108	18	18	18			54	Зачет

1. Цели освоения дисциплины

Целями освоения дисциплины «Введение в UML-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 UML-технологии на практике.

2. Место дисциплины в структуре ОПОП бакалавриата

Дисциплина «Введение в UML-технологии» входит в часть, формируемую участниками образовательных отношений ОПОП бакалавриата по направлению подготовки 02.03.02 Фундаментальная информатика и информационные технологии.

Дисциплина «Введение в UML-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 «Распределенным объектным технологиям», «Современным веб-технологиям» 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
--	--	---------------------------------	--------------------

ОПК-4. Способен участвовать в разработке технической документации программных продуктов и комплексов с использованием стандартов, норм и правил, а также в управлении проектами создания информационных систем на стадиях жизненного цикла	ОПК-4.1. Знает принципы сбора и анализа информации, создания информационных систем на стадиях жизненного цикла.	Знает: принципы объектно-ориентированного моделирования и разработки на основе международного стандарта RUP. Умеет: создавать различные типы диаграмм, используя CASE-инструменты IDE и Visual Paradigm. Владеет: навыками создания диаграмм вариантов использования и диаграмм классов.	Конспектирование лекций, подготовка и защита реферата, участие в дискуссиях. Разработка и реализация проектов на лабораторных занятиях.
	ОПК-4.2. Умеет осуществлять управление проектами информационных систем.	Знает: принципы объектно-ориентированного моделирования и разработки на основе международного стандарта RUP. Умеет: создавать полную модель прецедентов. Владеет: навыками создания диаграмм вариантов использования и диаграмм классов.	

	ОПК-4.3. Имеет практический опыт анализа и интерпретации информационных систем.	Знает: графическую нотацию языка UML и классы метамодели языка UML. Умеет: 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. Владеет: навыками использования языка UML с помощью CASE-инструментов при проектировании программного обеспечения.	
--	---	---	--

ПК-3 Способность к установке, администрированию программных систем; к реализации технического сопровождения информационных систем; к интеграции информационных систем с используемыми аппаратно-программными комплексами.	ПК-3.1. Знает методику установки и администрирования программных систем.	Знает: инфраструктуру стандарта UML и классификацию CASE-инструментов. Умеет: готовить доклад-сообщение по вопросу курса на выбор, используя официальную страницу стандарта UML. Владеет: навыками использования CASE-инструментов, встроенных в IDE.	Конспектирование лекций, подготовка и защита реферата, участие в дискуссиях. Разработка и реализация проектов на лабораторных занятиях.
	ПК-3.2. Умеет реализовывать техническое сопровождение информационных систем.	Знает: международные и российские стандарты жизненного цикла ПО. Умеет: разрабатывать техническое задание для проекта по заданному шаблону. Владеет: навыками объектно-ориентированного проектирования задач в различных предметных областях.	

	ПК-3.3. Имеет практический опыт разработки интеграции информационных систем с использованием аппаратно-программных комплексов.	Знает: инфраструктуру стандарта UML и классификацию CASE-инструментов. Умеет: готовить доклад-сообщение по вопросу курса на выбор, используя официальную страницу стандарта UML. Владеет: навыками использования CASE-инструментов, встроенных в IDE.	
--	--	--	--

4. Объем, структура и содержание дисциплины.

4.1. Объем дисциплины составляет 3 зачетные единицы, 108 академических часов.

4.2. Структура дисциплины.

4.2.1. Структура дисциплины в очной форме

№ п/ п	Разделы и темы дисциплины	Семестр	Неделя семестра	Виды учебной работы, включающая самостоятельную работу студентов и трудоемкость (в часах)					Формы текущего контроля успеваемости (по неделям семестра) Форма промежуточной аттестации (по семестрам)
				Лекции	Практические занятия	Лабораторные занятия	Самостоятельная работа	Контроль сам. работы	
	Модуль 1. Объектно-ориентированный анализ и проектирование								
1	Стандарты для объектно-ориентированного анализа и проектирования программного обеспечения	7	1-3	4	4	4	9		Прием лабораторных работ (ЛР) и реферата (Р)
2	Моделирование программных систем с помощью языка UML.	7	4-6	2	2	2	9		ЛР, Р
	Итого по модулю 1:			6	6	6	18		Модуль1

	Модуль 2. Графическая нотация языка UML и программирование								
3	Средства нотации языка UML для описания статической структуры модели системы.	7	7-9	4	4	4	9		ЛР, Р
4.	Средства нотации языка UML используемые для описания поведения моделируемой системы	7	10-12	2	2	2	9		ЛР, Р
	Итого по модулю 2:			6	6	6	18		Модуль 2
	Модуль 3. Диаграммы UML								
5	Структурные диаграммы	7	13-15	4	4	4	9		ЛР, Р
6.	Поведенческие диаграммы	7	16-18	2	2	2	9		
	Итого по модулю 3:			6	6	6	18		Модуль 3
	ИТОГО:	144		18	18	18	54		Зачет

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.

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

Тема 2. Моделирование программных систем с помощью языка UML.

Графическая нотация UML. Мета модель языка UML. Структура метамодели языка UML. Обзор основных понятий графической нотации и семантики языка UML. Unified Modeling Language (UML) - унифицированный язык моделирования. Метамета модель языка UML.

Модуль 2. Графическая нотация языка UML и программирование

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

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

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

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

Диаграммы коммуникации объектов (Communication diagram). Диаграммы последовательности взаимодействия (Sequence diagram). Представление обозначений нотации языка UML на диаграммах коммуникации и на диаграммах последовательности взаимодействия с помощью конструкций языков программирования. Представление обозначений нотации языка UML на диаграммах взаимодействия. Представление обозначений нотации языка UML на диаграммах деятельности.

Модуль 3. Диаграммы UML

Тема 6. Структурные диаграммы

Диаграмма классов. Диаграмма объектов. Диаграмма компонентов. Диаграмма композитной структуры. Диаграмма размещения. Диаграмма профиля.

Тема 7. Поведенческие диаграммы.

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

4.3.2 Содержание лабораторно-практических занятий по дисциплине

Практические и 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

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

Лабораторная работа 1. Разработка стандартов для объектно-ориентированного анализа и проектирования программного обеспечения. Составление технического задания для ВКР.

Практическая работа 2. Обзор основных понятий графической нотации и семантики языка UML. Unified Modeling Language (UML) - унифицированный язык моделирования.

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

Графическая нотация UML. Мета модель языка UML.

Модуль 2. Графическая нотация языка UML и программирование

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

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

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

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

Практическая работа 5. Средства нотации языка UML, используемые для описания поведения моделируемой системы.

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

Модуль 3. Диаграммы UML

Практическая работа 6. Структурные диаграммы

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

Практическая работа 7. Поведенческие диаграммы.

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

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, мастер-классы экспертов и специалистов по проектированию программных систем.

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 материала.	Контрольный фронтальный опрос.	См. разделы 7.3, 8, 9 данного документа
2.	Изучение рекомендованной литературы и материалов соответствующих форумов интернет.	Контрольный фронтальный опрос, прием и представление рефератов.	См. разделы 7.3, 8, 9 данного документа
3.	Подготовка к отчетам по лабораторным работам.	Проверка выполнения работ, опрос по теме работы.	См. разделы 7.3, 8, 9 данного документа
4.	Подготовка к сдаче промежуточных форм контроля.	Контрольные работы по каждому модулю и прием рефератов.	См. разделы 7.3, 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. Какие артефакты возникают в результате процессов ЖЦ ПО?
3. Что такое модель жизненного цикла?
4. Каковы основные этапы, процессы, роли участников ЖЦ ПО?
5. Как связан объём разработки с выбором модели ЖЦ ПО?
6. Какими стандартами и соглашениями регламентируются ЖЦ ПО?
7. Где проходит граница между анализом и проектированием?
8. В чём особенности объектно-ориентированного подхода к анализу, проектированию и конструированию ПО?
9. Что такое быстрая разработка приложений?
10. Дайте общую характеристику процесса объектно-ориентированного проектирования.
11. Как выявляются необходимые сущности и связи между ними?
12. Что такое требование? Каковы его основные характеристики?
13. Дайте характеристику языку UML. Каковы его главные достоинства и недостатки?
14. Какие виды диаграмм специфицирует язык UML?
15. Какие виды отношений между объектами способен выразить язык UML?
16. Опишите назначение и правила составления UML-диаграмм.
17. Какие виды ассоциаций можно представить на языке UML?
18. С какого вида диаграмм нужно начинать описание модели?
19. Приведите пример тернарной ассоциации.
20. Какие программные средства используются для упрощения процессов объектно-ориентированного анализа, проектирования и конструирования?
21. Как объектно-ориентированный подход стыкуется с различными моделями данных при создании схем баз данных?
22. Что такое обратный инжиниринг?
23. Охарактеризуйте отличия концептуального, логического и физического проектирования.
24. Каковы основные характеристики интегрированных сред разработки ПО?
25. В чём особенности многопользовательской работы со средствами поддержки ЖЦ ПО?
26. В чём заключается сложность поддержки ЖЦ гетерогенных программных систем?
27. Что такое аутсорсинг и аутстаффинг?
28. Что такое платформа? Чем платформа разработки отличается от платформы развёртывания?
29. Когда требуется формально описанные требования к ПО?
30. Что специфицирует техническое задание?
31. Как техническое задание связано с проведением приёмо-сдаточных испытаний?
32. Какие методы применяются при сборе требований?
33. Чем отличаются друг от друга интервью, тестирование и анкетирование?
34. Как связана методология проектирования с поддерживающими её программными средствами?
35. Перечислите основные возможности систем управления исходным кодом (версионного контроля).
36. Как правильно сравнивать изменения в исходный код?
37. В чём заключаются основные проблемы слияния версий?
38. Перечислите основные возможности систем профилирования.
39. Дайте обзор уровней оптимизации.
40. Когда нужно начинать оптимизацию, а когда нельзя её заниматься?
41. Что такое рефакторинг исходного кода?
42. Как формализуется надёжность программной системы?
43. В чём главная проблема функционального тестирования?

44. Что такое интеграционное тестирование?
45. Что является целью процесса развёртывания?
46. Чем отличаются понятия поддержки и сопровождения?
47. Что такое эволюция программного продукта?
48. Перечислите основные возможности систем построения дистрибутивов.
49. Как связаны процессы сборки и развёртывания?
50. 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. Метамодел ь 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.

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 – 70 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>

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>

4. Брауде, Эрик Дж. Технология разработки программного обеспечения. [Текст] / Эрик Дж. Брауде. — Спб.: Питер, 2004. — 656 с.

5. Рамбо, Дж., Блаха, М. UML 2.0. Объектно-ориентированное моделирование и разработка. [Текст] / Дж. Рамбо, М. Блаха. — СПб.: Питер, 2006. — 544 с.

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>

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](http://www.intuit.ru)

Форумы по компьютерным наукам и программированию:

- 3) www.stackoverflow.com
- 4) <http://www.cyberforum.ru/>

10. Методические указания для обучающихся по освоению дисциплины.

При выполнении лабораторных заданий программистский подход непременно должен присутствовать (без него решение не будет полноценным), однако, он не должен заслонять другие 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. Составление технического задания для ВКР.
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 и программирование. 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 и программирование. Тема 2. Средства нотации языка UML для описания сценариев использования моделируемой системы.	Проработка лекционного материала. Изучение рекомендованной литературы и материалов соответствующих форумов интернет для подготовки реферата. Подготовка к отчетам по лабораторным работам
Модуль 2. Графическая нотация языка UML и программирование. 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 3. Диаграммы UML. Тема 1. Структурные диаграммы	Проработка лекционного материала. Изучение рекомендованной литературы и материалов соответствующих форумов ин-

	тернет для подготовки реферата. Подготовка к отчетам по лабораторным работам. Подготовка к сдаче промежуточных форм контроля.
Модуль 3. Диаграммы UML. Тема 2. Поведенческие диаграммы	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.

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.