

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ 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: вариативный

Рабочая программа дисциплины «Введение в UML-технологии» составлена в 2017 году в соответствии с требованиями ФГОС ВО по направлению подготовки 02.03.02- Фундаментальная информатика и информационные технологии (уровень бакалавриат) от 12 марта 2015г. № 224.

Разработчик: кафедра дискретной математики и информатики, канд. физ.-мат. наук, доцент Раджабова Н.Ш.

Рабочая программа дисциплины одобрена:  
на заседании кафедры от 6 марта 2017 г., протокол № 4.

Зав. кафедрой Магомедов А.М.  
(подпись)

на заседании Методического совета факультета математики и компьютерных наук от 10 марта 2017г. протокол № 4.

Председатель Меджидов З.Г.  
(подпись)

Рабочая программа дисциплины согласована с учебно-методическим управлением « 09 » 03 2017г. Магомедов А.М.  
(подпись)

## Аннотация рабочей программы дисциплины

Дисциплина “Введение в UML-технологии” входит в вариативную часть образовательной программы бакалавриата по направлению 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: общепрофессиональных - ОПК-2, профессиональных - ПК-9.

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

Рабочая программа дисциплины предусматривает проведение следующих видов контроля успеваемости: в форме приема лабораторных работ, контрольной работы, реферата и итогового экзамена в конце семестра.

Объем дисциплины составляет 3 зачетных единиц (108 часов), в том числе в академических часах по видам учебных занятий

Се- местр	Учебные занятия						СРС, в том числе экза- мен	Форма проме- жуточной атте- стации (зачет, дифференциро- ванный зачет, экзамен
	в том числе							
	Контактная работа обучающихся с преподавателем							
Все го	из них					КСР	консуль- тации	
	Лек- ции	Лабора- торные занятия	Практи- ческие занятия					
7	108	18	36	36			18	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 ООП бакалавриата

Дисциплина «Введение в UML-технологии» относится к вариативной части образовательной программы бакалавриата по направлению 02.03.02 – «Фундаментальная информатика и информационные технологии» и преподается на 4 курсе в 1 семестре (6 зачетных единиц). Изучение предмета завершается письменным экзаменом в конце семестра.

Дисциплина «Введение в UML-технологии» логически и содержательно взаимосвязана с такими дисциплинами, как «Анализ информационных технологий», «Современные парадигмы программирования».

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

Компетенции	Формулировка компетенции из ФГОС ВО	Планируемые результаты обучения (показатели достижения заданного уровня освоения компетенций)
<b>ОПК-2</b>	<b>Способность применять в профессиональной деятельности современные языки программирования и языки баз данных, методологии системной инженерии, системы автоматизации проектирования, электронные библиотеки и коллекции, сетевые технологии, библиотеки и пакеты программ, современные профессиональные стандарты информационных технологий.</b>	<b>Знать:</b> принципы объектно-ориентированного моделирования и разработки. <b>Уметь:</b> создавать различные типы диаграмм, используя CASE-инструменты среды Visual Studio и Rational Rose. <b>Владеть:</b> навыками создания в Rational Rose диаграмм вариантов использования и диаграмм классов.

<p><b>ПК-9</b></p>	<p><b>Способность разрабатывать, оценивать и реализовывать процессы жизненного цикла информационных систем, программного обеспечения, сервисов информационных технологий, а также реализовывать методы и механизмы оценки и анализа функционирования средств и информационных технологий, разрабатывать проектную и программную документацию, удовлетворяющую нормативным требованиям.</b></p>	<p>Знать: графическую нотацию языка UML и классы метамодели языка UML.</p> <p><b>Уметь:</b></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><b>Владеть:</b></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 3 зачетных единиц, 108 академических часов.

4.2. Структура дисциплины.

№ п/п	Разделы и темы дисциплины	Семестр	Неделя семестра	Виды учебной работы, включая самостоятельную работу студентов и трудоемкость (в часах)					Формы текущего контроля успеваемости (по неделям семестра) Форма промежуточной аттестации (по семестрам)
				Всего	Лекции	Практические занятия	Лабораторные занятия	Самостоятельная работа	
<b>Модуль 1. Объектно-ориентированный анализ и проектирование</b>									
1	Стандарты для объектно-ориентированного анализа и проектирования программного обеспечения	7	1-2	16	2	6	6	2	Прием лабораторных работ (ЛР) и реферата (Р)
2	Моделирование программных систем с помощью языка UML.	7	3-4	20	4	6	6	4	ЛР, Р
Итого				36	6	12	12	6	Модуль 1
<b>Модуль 2. Графическая нотация языка UML для описания статической структуры</b>									
3	Средства нотации языка UML для описания статической структуры модели системы.	7	5-6	16	2	6	6	2	ЛР, Р
4.	Средства нотации языка UML для описания сценариев использования моделируемой системы.	7	7-8	20	4	6	6	4	ЛР, Р
Итого				36	6	12	12	6	Модуль 2
<b>Модуль 3. Графическая нотация языка UML для описания поведения системы</b>									
5.	Средства нотации языка UML используемые для описания поведения моделируемой системы	7	9-10	36	6	12	12	6	ЛР, Р
Итого				36	6	12	12	6	Зачет
ИТОГО:				108	18	36	36	18	

### 4.3. Содержание дисциплины, структурированное по темам (разделам).

#### **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. Обзор основных понятий графической нотации и семантики языка UML. Unified Modeling Language (UML) - унифицированный язык моделирования.

#### **Модуль 2. Графическая нотация языка UML для описания статической структуры**

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.

#### **Модуль 3. Графическая нотация языка UML для описания поведения системы**

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.

#### **Темы практических занятий**

#### **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. Обзор основных понятий графической нотации и семантики языка UML. Unified Modeling Language (UML) - унифицированный язык моделирования.

#### **Модуль 2. Графическая нотация языка UML для описания статической структуры**

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.

### **Модуль 3. Графическая нотация языка UML для описания поведения системы**

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.

## **Темы лабораторных занятий**

### **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. Обзор основных понятий графической нотации и семантики языка UML. Unified Modeling Language (UML) - унифицированный язык моделирования.

### **Модуль 2. Графическая нотация языка UML для описания статической структуры**

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.

### **Модуль 3. Графическая нотация языка UML для описания поведения системы**

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.

## **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 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. Перечень компетенций с указанием этапов их формирования в процессе освоения образовательной программы.

Перечень компетенций с указанием этапов их формирования приведен в описании образовательной программы.

Компетенция	Знания, умения, навыки	Процедура освоения
ОПК-2	<b>Знать:</b> принципы объектно-ориентированного моделирования и разработки. <b>Уметь:</b> создавать различные типы диаграмм, используя CASE-инструменты среды Visual Studio и Rational Rose. <b>Владеть:</b> навыками создания в Rational Rose диаграмм вариантов использования и диаграмм классов.	Письменный опрос, выполнение лабораторных заданий.
ПК-9	<b>Знать:</b> графическую нотацию языка UML	Устный опрос, вы-

	<p>и классы метамодели языка UML.</p> <p><b>Уметь:</b> 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><b>Владеть:</b> навыками использования языка UML с помощью CASE-инструментов при проектировании программного обеспечения.</p>	<p>полнение лабораторных заданий, подготовка реферата.</p>
--	---	--

7.2. Описание показателей и критериев оценивания компетенций, описание шкал оценивания.

#### ОПК-2

Схема оценки уровня формирования компетенции «**способность применять в профессиональной деятельности современные языки программирования и языки баз данных, методологии системной инженерии, системы автоматизации проектирования, электронные библиотеки и коллекции, сетевые технологии, библиотеки и пакеты программ, современные профессиональные стандарты информационных технологий**».

Уровень	Показатели (что обучающийся должен продемонстрировать)	Оценочная шкала		
		Удовлетворительно	Хорошо	Отлично
Пороговый	Количество построенных диаграмм, связанных с тематикой занятий	1	2	3

#### ПК-9

Схема оценки уровня формирования компетенции «**способность разрабатывать, оценивать и реализовывать процессы жизненного цикла информационных систем, программного обеспечения, сервисов информационных технологий, а также реализовывать методы и механизмы оценки и анализа функционирования средств и информационных технологий, разрабатывать проектную и программную документацию, удовлетворяющую нормативным требованиям**».

Уровень	Показатели (что обучающийся должен продемонстрировать)	Оценочная шкала		
		Удовлетворительно	Хорошо	Отлично
Пороговый	Процент выполненных заданий.	50%	65%	80%

Если хотя бы одна из компетенций не сформирована, то положительной оценки по дисциплине быть не может.

## 7.3. Типовые контрольные задания

### 7.3.1 Вопросы для самостоятельной работы

1. Что такое жизненный цикл программного обеспечения?
2. Какие артефакты возникают в результате процессов ЖЦПО?
3. Что такое модель жизненного цикла?
4. Каковы основные этапы, процессы, роли участников ЖЦПО?
5. Как связан объём разработки с выбором модели ЖЦПО?
6. Какими стандартами и соглашениями регламентируются ЖЦПО?
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 ЖЦПО?
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?

### 7.3.2 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.

### 7.3.3 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.

### **7.4. 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) основная литература:

1. Брауде, Эрик Дж. Технология разработки программного обеспечения. [Текст] / Эрик Дж. Брауде. – СПб.: Питер, 2004. – 656 с.
2. Орлов, С. А. Технологии разработки программного обеспечения. Разработка сложных программных систем. [Текст] / С. А. Орлов. – СПб.: Питер, 2003. – 480 с.
3. Рамбо, Дж., Блаха, М. UML 2.0. Объектно-ориентированное моделирование и разработка. [Текст] / Дж. Рамбо, М. Блаха. – СПб.: Питер, 2006. – 544 с.

б) дополнительная литература:

4. Арлоу Дж., Нейштадт А. UML 2 и Унифицированный процесс. Практический объектно-ориентированный анализ и проектирование – СПб. Символ-Плюс, 2007.
5. Фаулер М. UML. Основы. 3-е издание. Краткое руководство по стандартному языку объектного моделирования.: Пер. с англ. – СПб: Символ-Плюс, 2005.

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

### Описание стандарта:

- 1) [www.uml.org](http://www.uml.org)
- 2) [www.uml-diagrams.org](http://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](http://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.
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 для описания статической структуры. 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 для описания статической структуры. 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 3. Графическая нотация языка UML для описания поведения системы Тема 1. 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. Типы диаграмм. Тема 1. Структурные диаграммы.	Проработка лекционного материала. Изучение рекомендованной литературы и материалов соответствующих форумов интернет для подготовки реферата. Подготовка к отчетам по лабораторным работам.
Модуль 4. Типы диаграмм. Тема 2. Поведенческие диаграммы.	Проработка лекционного материала. Изучение рекомендованной литературы и материалов соответствующих форумов интернет для подготовки отчетов по лабораторным работам.
Модуль 5. 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 5. 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 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.