

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ 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
очная

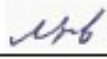
Статус дисциплины: базовый

Махачкала, 2017

Рабочая программа дисциплины «Объектно-ориентированные CASE- технологии» составлена в 2017 году в соответствии с требованиями ФГОС ВО по направлению подготовки 02.04.02 - Фундаментальная информатика и информационные технологии (уровень магистратуры) от 17.08.2015 г. № 830.

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


Рабочая программа дисциплины одобрена: на заседании кафедры дискретной математики и информатики от «13» 01 2017 г., протокол № 5.

Зав. кафедрой  Магомедов А.М.
(подпись)

на заседании Методической комиссии _____ факультета от
« 17 » 01 2017 г., протокол № 5

Председатель  З.Г. Меджидов
(подпись)

Рабочая программа дисциплины согласована с учебно-методическим

управлением « _____ » _____ 2017 г. 
(подпись)

Аннотация рабочей программы дисциплины

Дисциплина “Объектно-ориентированные CASE-технологии” входит в базовую часть образовательной программы магистратуры по направлению 02.04.02 - 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: общекультурных - ОК-1, общепрофессиональных - ОПК-3, профессиональных: ПК-2, ПК-4, ПК-6.

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

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

Объем дисциплины составляет 4 зачетные единицы (144 часа), в том числе в академических часах по видам учебных занятий

Се- местр	Учебные занятия						СРС, в том числе экза- мен	Форма промежу- точной аттеста- ции (зачет, диф- ференцированный зачет, экзамен
	в том числе							
	Контактная работа обучающихся с преподавателем					СРС, в том числе экза- мен		
Все- го	из них	Лек- ции	Лаборатор- ные заня- тия	Практи- ческие занятия	КСР		консультации	экзамен
1	144	12	26					

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 ООП магистратуры

Дисциплина «Объектно-ориентированные CASE- технологии» относится к базовой части образовательной программы магистратуры по направлению 02.04.02 - Фундаментальная информатика и информационные технологии и преподается на 1 курсе в 1 семестре (4 зачетные единицы). Изучение предмета завершается письменным экзаменом в конце семестра.

Дисциплина «Объектно-ориентированные CASE- 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)
ОК-1	Способность к абстрактному мышлению, анализу, синтезу.	Знать:принципы самостоятельного поиска достоверных источников информации. Уметь:обрабатывать, анализировать и синтезировать информацию для выбора метода решения проблемы в стандартных условиях. Владеть:навыками решения проблемы с использованием выбранного метода.
ОПК-3	Способность использовать и применять	Знать: принципы объектно-

	углубленные теоретические и практические знания в области фундаментальной информатики и информационных технологий.	ориентированного моделирования и разработки. Уметь: создавать различные типы диаграмм, используя CASE-инструменты среды VisualStudio или RationalRose. Владеть: навыками создания в RationalRose диаграмм вариантов использования и диаграмм классов.
ПК-2	Способность использовать углубленные теоретические и практические знания в области информационных технологий и прикладной математики, фундаментальных концепций и системных методологий, международных и профессиональных стандартов в области информационных технологий.	Знать: графическую нотацию языка UML и классы метамодели языка UML. Уметь: 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. Владеть: навыками использования языка UML с помощью CASE-инструментов при проектировании программного обеспечения.
ПК-4	Способность к применению методов тематического и алгоритмического моделирования при решении теоретических и прикладных задач	Знать: графическую нотацию языка UML и классы метамодели языка UML. Уметь: 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. Владеть: навыками использования языка UML с помощью

		CASE-инструментов при проектировании программного обеспечения.
ПК-6	Способность к собственному видению прикладного аспекта встрогих математических формулировках.	<p>Знать: графическую нотацию языка UML и классы метамодели языка UML.</p> <p>Уметь: строить 4 основных типа диаграмм для проектирования программных систем.</p> <p>Владеть: навыками использования языка UML с помощью CASE-инструментов и чтения основных типов диаграмм.</p>

4. Объем, структура и содержание дисциплины.

4.1. Объем дисциплины составляет 4 зачетные единицы, 144 академических часа.

4.2. Структура дисциплины.

№ п/п	Разделы и темы дисциплины	Семестр	Неделя семестра	Виды учебной работы, включая самостоятельную работу студентов и трудоемкость (в часах)				Самостоятельная работа	Формы текущего контроля успеваемости (по неделям семестра) Форма промежуточной аттестации (по семестрам)
				Всего	Лекции	Практические занятия	Лабораторные занятия		
Модуль 1. Объектно-ориентированный анализ и проектирование									
1	Стандарты для объектно-ориентированного анализа и проектирования программного обеспечения	1	1	18	2		4	12	Прием лабораторных работ (ЛР) и реферата (Р)
2	Моделирование программных систем с помощью языка UML.	1	2	18	2		4	12	ЛР, Р
	Итого			36	4		8	24	Модуль 1
Модуль 2. Графическая нотация языка UML									

3	Средства нотации языка UML для описания статической структуры модели системы.	1	3	12	2		4	6	ЛР, Р
4.	Средства нотации языка UML для описания сценариев использования моделируемой системы.	1	4	12	2		4	6	ЛР, Р
5.	Средства нотации языка UML используемые для описания поведения моделируемой системы	1	5	12	2		4	6	ЛР, Р
	Итого			36	6		12	18	Модуль 2
Модуль 3. Структура метамодели языка UML									
6.	Моделирование отношения "владелец-собственность".	1	6	20	2		4	14	ЛР, Р
7.	Моделирование пространств имен с помощью классов Namespace и NamedElement	1	7	16			2	14	ЛР, Р
	Итого			36	2		6	28	Модуль 3
8.	Подготовка к экзамену	1	8	36				36	
	ИТОГО:			108	12		26	70	Экзамен

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.

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. Графическая нотация языка UML

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.

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

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

Модуль 3. Структура метамодели языка UML

Тема 6. Моделирование отношения "владелец-собственность"

1. Метамодель языка UML 2.0.
2. Моделирование отношения "владелец-собственность" с помощью базового класса метамодели Element.
3. Моделирование направленных отношений с помощью класса метамодели DirectedRelationship.

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

1. Моделирование пространств имен с помощью классов Namespace и NamedElement.
2. Моделирование импорта элементов модели в пространство имен с помощью классов PackageableElement, ElementImport, PackageImport.

Темы лабораторных занятий совпадают с темами модулей. Задания лабораторных работ состоят в построении соответствующих типов диаграмм UML в программной среде Rational Rose.

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, определяют 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
ОК-1	Знать: принципы самостоятельного поиска достоверных источников информации. Уметь: обрабатывать, анализировать и синтезировать информацию для выбора метода решения проблемы в стандартных условиях. Владеть: навыками решения проблемы с использованием выбранного метода.	Устный опрос, письменный опрос, подготовка реферата.
ОПК-3	Знать: принципы объектно-ориентированного моделирования и разработки. Уметь: создавать различные типы диаграмм, используя CASE-инструменты среды VisualStudio и RationalRose. Владеть: навыками создания в RationalRose диаграмм вариантов использования и диаграмм классов.	Письменный опрос, выполнение лабораторных заданий.
ПК-2	Знать: графическую нотацию языка UML и классы метамодели языка UML. Уметь: отображать графическую нотацию языка UML в программы на языках программиро-	Устный опрос, выполнение лабораторных заданий, подготовка реферата.

	<p>вания 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>Владеть: навыками использования языка UML с помощью CASE-инструментов при проектировании программного обеспечения.</p>	
ПК-4	<p>Знать: графическую нотацию языка UML и классы метамодели языка UML.</p> <p>Уметь: 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>Владеть: навыками использования языка UML с помощью CASE-инструментов при проектировании программного обеспечения.</p>	Устный опрос, выполнение лабораторных заданий, подготовка реферата.
ПК-6	<p>Знать: графическую нотацию языка UML и классы метамодели языка UML.</p> <p>Уметь: строить 4 основных типа диаграмм для проектирования программных систем.</p> <p>Владеть: навыками использования языка UML с помощью CASE-инструментов и чтения основных типов диаграмм.</p>	Устный опрос, выполнение лабораторных заданий, подготовка реферата.

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

ОК-1

Схема оценки уровня формирования компетенции «Способность к абстрактному мышлению, анализу, синтезу».

Уровень	Показатели (что обучающийся должен продемонстрировать)	Оценочная шкала		
		Удовлетворительно	Хорошо	Отлично
Базовый Способен самостоятельно выявить надежные источники информации по указанной проблеме, собрать информацию, необходимую и достаточную	<p>Способен выявить достоверные источники информации</p> <p>Способен обработать, проанализировать и синтезировать информацию.</p> <p>Способен выбрать метод решения проблемы в стандартных условиях</p>	<p>Допускает ошибки в определении достоверности источников информации, способен правильно решать только типичные, наиболее часто встречающиеся проблемы в конкретной области (обрабатывать ин-</p>	<p>В большинстве случаев способен выявить достоверные источники информации, обработать, анализировать и синтезировать предложенную информа-</p>	<p>Свободно и уверенно находит достоверные источники информации, оперирует предоставленной информацией, отлично владеет навыками анализа и синтеза информации,</p>

для решения проблемы, обработать ее, избрать метод решения проблемы в стандартных условиях и решить ее.	ях. Способен решить проблему, используя выбранный метод.	формацию, выбрать метод решения проблемы и решать ее).	цию, выбрать метод решения проблемы и решить ее. Допускает единичные серьезные ошибки в решении проблем, испытывает сложности в редко встречающихся или сложных случаях решения проблем.	знает все основные методы решения проблем, предусмотренные учебной программой, знает типичные ошибки и возможные сложности при решении той или иной проблемы и способен выбрать и эффективно применить адекватный метод решения конкретной проблемы.
---	---	--	--	--

ОПК-3

Схема оценки уровня формирования компетенции «Способностью использовать и применять углубленные теоретические и практические знания в области фундаментальной информатики и информационных технологий»

Уровень	Показатели (что обучающийся должен продемонстрировать)	Оценочная шкала		
		Удовлетворительно	Хорошо	Отлично
Базовый Способен создавать различные типы диаграмм, используя CASE-инструменты среды VisualStudio и RationalRose, знает принципы объектно-ориентированного моделирования и разработки сложных программных систем.	Способен использовать принципы объектно-ориентированного моделирования и разработки. Способен создавать различные типы диаграмм, используя CASE-инструменты среды VisualStudio и RationalRose. Способен моделировать и разрабатывать сложные программные системы для решения задач реального мира на основе CASE-технологий.	Допускает ошибки при моделировании и разработке программных систем, способен создавать диаграммы простой структуры с использованием CASE-инструментов среды VisualStudio и RationalRose.	В большинстве случаев способен разработать сложную программную систему с использованием CASE-инструментов среды Visual Studio и Rational Rose. Испытывает сложности в редко встречающихся или сложных случаях решения проблем.	Свободно и уверенно разрабатывает объектно-ориентированную модель сложной программной системы для решения задачи реального мира с использованием CASE-инструментов среды Visual Studio и Rational Rose.

ПК-2

Схема оценки уровня формирования компетенции «Способность использовать углубленные теоретические и практические знания в области информационных технологий и прикладной математики, фундаментальных концепций и системных методов»

логий, международных и профессиональных стандартов в области информационных технологий»

Уровень	Показатели (что обучающийся должен продемонстрировать)	Оценочная шкала		
		Удовлетворительно	Хорошо	Отлично
Базовый	Процент выполненных заданий, связанных с изучением международных и профессиональных стандартов в области информационных технологий	50%	65%	80%

ПК-4

Схема оценки уровня формирования компетенции

«Способность к применению методов математического и алгоритмического моделирования при решении теоретических и прикладных задач».

Уровень	Показатели (что обучающийся должен продемонстрировать)	Оценочная шкала		
		Удовлетворительно	Хорошо	Отлично
Базовый	Процент выполненных заданий, связанных с применением методов математического и алгоритмического моделирования	50%	65%	80%

ПК-6

Схема оценки уровня формирования компетенции

«Способность к собственному видению прикладного аспекта в строгих математических формулировках».

Уровень	Показатели (что обучающийся должен продемонстрировать)	Оценочная шкала		
		Удовлетворительно	Хорошо	Отлично
Базовый	Процент выполненных заданий, связанных с проектированием распределенных программных систем	50%	65%	80%

Если хотя бы одна из компетенций не сформирована, то положительная оценка 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. 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?

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 – 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) 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
- 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 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 про-	Проработка лекционного материала. Изучение рекомендованной литературы и материалов соответствующих форумов интернет для подготовки реферата.

ектирования программного обеспечения.	
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 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. 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. 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 3. 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: MicrosoftWindows, UbuntuLinux, MicrosoftVisualStudioUltimate, RationalRose, 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.