

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ  
Федеральное государственное автономное учреждение  
высшего профессионального образования  
"Казанский (Приволжский) федеральный университет"  
Институт математики и механики им. Н.И. Лобачевского



**УТВЕРЖДАЮ**

Проректор  
по образовательной деятельности КФУ  
Проф. Минзарипов Р.Г.

\_\_\_\_\_ 20\_\_ г.

**Программа дисциплины**

Информационные системы на основе CASE-технологий М2.ДВ.3

Направление подготовки: 010100.68 - Математика

Профиль подготовки: Геометрия и топология

Квалификация выпускника: магистр

Форма обучения: очное

Язык обучения: русский

**Автор(ы):**

Липачев Е.К.

**Рецензент(ы):**

-

**СОГЛАСОВАНО:**

Заведующий(ая) кафедрой: Авхадиев Ф. Г.

Протокол заседания кафедры No \_\_\_\_ от " \_\_\_\_ " \_\_\_\_\_ 201\_\_ г

Учебно-методическая комиссия Института математики и механики им. Н.И. Лобачевского :

Протокол заседания УМК No \_\_\_\_ от " \_\_\_\_ " \_\_\_\_\_ 201\_\_ г

Регистрационный No

Казань  
2014

## Содержание

1. Цели освоения дисциплины
2. Место дисциплины в структуре основной образовательной программы
3. Компетенции обучающегося, формируемые в результате освоения дисциплины /модуля
4. Структура и содержание дисциплины/ модуля
5. Образовательные технологии, включая интерактивные формы обучения
6. Оценочные средства для текущего контроля успеваемости, промежуточной аттестации по итогам освоения дисциплины и учебно-методическое обеспечение самостоятельной работы студентов
7. Литература
8. Интернет-ресурсы
9. Материально-техническое обеспечение дисциплины/модуля согласно утвержденному учебному плану

Программу дисциплины разработал(а)(и) доцент, к.н. (доцент) Липачев Е.К. Кафедра теории функций и приближений отделение математики , Evgeny.Lipachev@kpfu.ru

### 1. Цели освоения дисциплины

Целями освоения дисциплины (модуля) "Информационные системы на основе CASE-технологий" являются развитие у студентов профессиональных компетенций в области современных информационных технологий.

### 2. Место дисциплины в структуре основной образовательной программы высшего профессионального образования

Данная учебная дисциплина включена в раздел " М2.ДВ.3 Профессиональный" основной образовательной программы 010100.68 Математика и относится к дисциплинам по выбору. Осваивается на 1 курсе, 1 семестр.

Дисциплина "Информационные системы на основе CASE-технологий" входит в цикл профессиональных дисциплин по выбору.

Для прохождения курса необходимы знания компьютерных наук в объеме стандартного университетского курса. Освоение дисциплины "Информационные системы на основе CASE-технологий" позволит обучающимся получить представление о современных методах создания объектно-ориентированных систем.

### 3. Компетенции обучающегося, формируемые в результате освоения дисциплины /модуля

В результате освоения дисциплины формируются следующие компетенции:

Шифр компетенции	Расшифровка приобретаемой компетенции
ОК-10 (общекультурные компетенции)	Общекультурные: - умением находить, анализировать и контекстно обрабатывать информацию, в том числе относящуюся к новым областям знаний, непосредственно не связанным со сферой профессиональной деятельности (ОК-10).
ОК-5 (общекультурные компетенции)	Общекультурные: - способность порождать новые идеи (ОК-5);
ОК-9 (общекультурные компетенции)	Общекультурные: - способностью к организации и планированию (ОК-9);
ПК-1 (профессиональные компетенции)	Профессиональные: - владение методами математического моделирования при анализе глобальных проблем на основе глубоких знаний фундаментальных математических дисциплин и компьютерных наук (ПК-1);
ПК-3 (профессиональные компетенции)	Профессиональные: - способность к интенсивной научно-исследовательской и научно-изыскательской деятельности (ПК-3);
ПК-10 (профессиональные компетенции)	Профессиональные: - определение общих форм, закономерностей, инструментальных средств для групп дисциплин (ПК-10);
ПК-5 (профессиональные компетенции)	Профессиональные: - умение публично представить собственные новые научные результаты (ПК-5);

Шифр компетенции	Расшифровка приобретаемой компетенции
ПК-6 (профессиональные компетенции)	Профессиональные: - самостоятельное построение целостной картины дисциплины (ПК-6);
ПК-7 (профессиональные компетенции)	Профессиональные: - умение ориентироваться в современных алгоритмах компьютерной математики, совершенствовать, углублять и развивать математическую теорию, лежащую в их основе (ПК-7);

В результате освоения дисциплины студент:

1. должен знать:

методы объекто-ориентированной разработки

2. должен уметь:

создавать модели информационных систем

3. должен владеть:

приемами визуального проектирования в IBM Rational Rose

создавать модели информационных систем

#### 4. Структура и содержание дисциплины/ модуля

Общая трудоемкость дисциплины составляет зачетных(ые) единиц(ы) 108 часа(ов).

Форма промежуточного контроля дисциплины экзамен в 1 семестре.

Суммарно по дисциплине можно получить 100 баллов, из них текущая работа оценивается в 50 баллов, итоговая форма контроля - в 50 баллов. Минимальное количество для допуска к зачету 28 баллов.

86 баллов и более - "отлично" (отл.);

71-85 баллов - "хорошо" (хор.);

55-70 баллов - "удовлетворительно" (удов.);

54 балла и менее - "неудовлетворительно" (неуд.).

#### 4.1 Структура и содержание аудиторной работы по дисциплине/ модулю

##### Тематический план дисциплины/модуля

N	Раздел Дисциплины/ Модуля	Семестр	Неделя семестра	Виды и часы аудиторной работы, их трудоемкость (в часах)			Текущие формы контроля
				Лекции	Практические занятия	Лабораторные работы	
1.	Тема 1. Тема. Методологии объектно-ориентированного анализа и проектирования сложных систем. CASE Computer Aided Software Engineering	1	1	2	1	0	

N	Раздел Дисциплины/ Модуля	Семестр	Неделя семестра	Виды и часы аудиторной работы, их трудоемкость (в часах)			Текущие формы контроля
				Лекции	Практические занятия	Лабораторные работы	
2.	Тема 2. Тема. Жизненный Цикл Программного Обеспечения. Модели Жизненного Цикла.	1	2	1	1	0	
3.	Тема 3. Тема. Методология быстрой разработки приложений RAD (Rapid Application Development).	1	3	1	1	0	
4.	Тема 4. Тема. Структурный подход к проектированию информационных систем. Основные модели.	1	4	1	2	0	
5.	Тема 5. Тема. Объектно-ориентированные CASE-средства.	1	5	1	1	0	
6.	Тема 6. Тема. Universal Modeling Language (UML) обзор. Концептуальные области UML. Назначение моделей.	1	6	1	2	0	
7.	Тема 7. Тема. Представления модели и диаграммы в UML.	1	7-8	1	2	0	
8.	Тема 8. Тема. Статическое представление модели. Классификаторы и их отношения.	1	9	1	1	0	
9.	Тема 9. Тема. Классы в UML. Диаграммы классов. Абстрактные, корневые, листовые, полиморфные элементы. Кратность, атрибуты, операции.	1	10	1	2	0	

N	Раздел Дисциплины/ Модуля	Семестр	Неделя семестра	Виды и часы аудиторной работы, их трудоемкость (в часах)			Текущие формы контроля
				Лекции	Практические занятия	Лабораторные работы	
10.	Тема 10. Тема. Моделирование поведения. Взаимодействия. Прецеденты. Диаграммы прецедентов. Диаграммы взаимодействия (диаграммы последовательностей, диаграммы кооперации). Диаграмма деятельности.	1	11-12	1	2	0	
11.	Тема 11. Тема. Представление в виде конечного автомата. Конечный автомат. Событие. Состояние Переход.	1	13	1	1	0	
12.	Тема 12. Тема. Архитектурное моделирование. Диаграммы компонентов. Диаграммы развертывания.	1	14	0	1	0	
13.	Тема 13. Тема. Визуальное моделирование с помощью IBM Rational Rose.	1	15	1	1	0	
14.	Тема 14. Тема. Генерация программного кода (C++, Java, XML) в IBM Rational Rose.	1	16	1	1	0	
15.	Тема 15. Тема. Процесс прямого и обратного проектирования в IBM Rational Rose.	1	17-18	2	1	0	
.	Тема . Итоговая форма контроля	1		0	0	0	экзамен
	Итого			16	20	0	

#### 4.2 Содержание дисциплины

**Тема 1. Тема. Методологии объектно-ориентированного анализа и проектирования сложных систем. CASE Computer Aided Software Engineering**

**лекционное занятие (2 часа(ов)):**

CASE (Computer-Aided Software Engineering) ? набор инструментов и методов программной инженерии для проектирования программного обеспечения, который помогает обеспечить высокое качество программ, отсутствие ошибок и простоту в обслуживании программных продуктов. Также под CASE понимают совокупность методов и средств проектирования информационных систем с использованием CASE-инструментов

**практическое занятие (1 часа(ов)):**

Основной целью CASE-технологии является разграничение процесса проектирования программных продуктов от процесса кодирования и последующих этапов разработки, максимально автоматизировать процесс разработки. Для выполнения поставленной цели CASE-технологии используют два принципиально разных подхода к проектированию: структурный и объектно-ориентированный. Структурный подход предполагает декомпозицию (разделение) поставленной задачи на функции, которые необходимо автоматизировать. В свою очередь, функции также разбиваются на подфункции, задачи, процедуры. В результате получается упорядоченная иерархия функций и передаваемой информацией между функциями.

**Тема 2. Тема. Жизненный Цикл Программного Обеспечения. Модели Жизненного Цикла.**

**лекционное занятие (1 часа(ов)):**

Жизненный цикл программного обеспечения (ПО) ? период времени, который начинается с момента принятия решения о необходимости создания программного продукта и заканчивается в момент его полного изъятия из эксплуатации. Этот цикл ? процесс построения и развития ПО.

**практическое занятие (1 часа(ов)):**

Стандарт группирует различные виды деятельности, которые могут выполняться в течение жизненного цикла программных систем, в семь групп процессов. Каждый из процессов жизненного цикла в пределах этих групп описывается в терминах цели и желаемых выходов, списков действий и задач, которые необходимо выполнять для достижения этих результатов. процессы соглашения ? два процесса; процессы организационного обеспечения проекта ? пять процессов; процессы проекта ? семь процессов; технические процессы ? одиннадцать процессов; процессы реализации программных средств ? семь процессов; процессы поддержки программных средств ? восемь процессов; процессы повторного применения программных средств ? три процесса.

**Тема 3. Тема. Методология быстрой разработки приложений RAD (Rapid Application Development).**

**лекционное занятие (1 часа(ов)):**

RAD (от англ. rapid application development ? быстрая разработка приложений) ? концепция создания средств разработки программных продуктов, уделяющая особое внимание скорости и удобству программирования, созданию технологического процесса, позволяющего программисту максимально быстро создавать компьютерные программы. Практическое определение: RAD ? это жизненный цикл процесса проектирования, созданный для достижения более высокой скорости разработки и качества ПО, чем это возможно при традиционном подходе к проектированию.

**практическое занятие (1 часа(ов)):**



Фазы разработки. 1. Планирование ? совокупность требований, полученных при системном планировании и анализе процедуры разработки жизненного цикла (SDLC). На этом этапе пользователи, менеджеры и IT-специалисты обсуждают задачи проекта, его объём, системные требования, а также сложности, которые могут возникнуть при разработке. Фаза завершается согласованием ключевых моментов с RAD-группой и получением от руководителей проекта разрешения на продолжение. 2. Пользовательское проектирование ? на протяжении данного этапа пользователи, взаимодействуя с системными аналитиками, разрабатывают модели и прототипы, которые включают в себя все необходимые системные функции. Для перевода пользовательских прототипов в рабочие модели RAD-группа обычно использует технику объединенной разработки приложений (JAD) и CASE-инструменты. Пользовательское проектирование оказывается длительным интерактивным процессом, который позволяет пользователям понять, изменить и в конечном счете выбрать рабочую модель, отвечающую их требованиям. Конструирование ? этап, в котором основная задача заключается в разработке программ и приложений. Аналогична стадии ?реализация? в SDLC. В RAD, однако, пользователи продолжают принимать участие и по-прежнему могут предлагать изменения или улучшения в виде разработанных ими докладов. В их задачи входит программирование и разработка приложений, написание кода, интеграция модулей и системное тестирование. Переключение ? включает в себя операции по конверсии данных, тестирование, переход на новую систему и тренировку пользователей. По своим задачам напоминает финальную стадию SDLC. Сравнивая с традиционными методами разработки ПО, весь процесс оказывается сжатым по времени. Как результат, новая система оказывается быстрее построенной, доставленной до заказчика и установленной на рабочих местах.

#### **Тема 4. Тема. Структурный подход к проектированию информационных систем. Основные модели.**

##### ***лекционное занятие (1 часа(ов)):***

Сущность структурного подхода к разработке ИС заключается в ее декомпозиции (разбиении) на автоматизируемые функции: система разбивается на функциональные подсистемы, которые в свою очередь делятся на подфункции, подразделяемые на задачи и так далее. Процесс разбиения продолжается вплоть до конкретных процедур. При этом автоматизируемая система сохраняет целостное представление, в котором все составляющие компоненты взаимосвязаны. При разработке системы "снизу-вверх" от отдельных задач ко всей системе целостность теряется, возникают проблемы при информационной стыковке отдельных компонентов.

##### ***практическое занятие (2 часа(ов)):***

В структурном анализе используются в основном две группы средств, иллюстрирующих функции, выполняемые системой и отношения между данными. Каждой группе средств соответствуют определенные виды моделей (диаграмм), наиболее распространенными среди которых являются следующие: SADT (Structured Analysis and Design Technique) модели и соответствующие функциональные диаграммы; DFD (Data Flow Diagrams) диаграммы потоков данных); ERD (Entity-Relationship Diagrams) диаграммы "сущность-связь".

#### **Тема 5. Тема. Объектно-ориентированные CASE-средства.**

##### ***лекционное занятие (1 часа(ов)):***

Объектно - ориентированное проектирование предполагает объектную декомпозицию системы. Объект - это реально существующая сущность, имеющая важное функциональное назначение в данной предметной области. Объект характеризуется структурой, состоянием, четко определяемым поведением. Состояние объекта определяется перечнем всех возможных (обычно статических) свойств и текущими значениями (обычно динамическими) каждого из этих свойств. Свойства объекта характеризуются значениями его параметров.

##### ***практическое занятие (1 часа(ов)):***



Концепции объектно-ориентированного подхода и распределенных вычислений стали базой для создания консорциума Object Management Group (OMG), членами которой являются более 500 ведущих компьютерных компаний (Sun, DEC, IBM, HP, Motorola и др.). Основным направлением деятельности консорциума является разработка спецификаций и стандартов для создания распределенных объектных систем в разнородных средах. Базисом стали спецификации под названием Object Management Architecture (OMA). Идеальное объектно-ориентированное CASE-средство должно содержать четыре основных блока: анализ, проектирование, разработка и инфраструктура.

## **Тема 6. Тема. Universal Modeling Language (UML) обзор. Концептуальные области UML. Назначение моделей.**

### ***лекционное занятие (1 часа(ов)):***

UML (Unified Modeling Language ? унифицированный язык моделирования) ? язык графического описания для объектного моделирования в области разработки программного обеспечения. UML является языком широкого профиля, это ? открытый стандарт, использующий графические обозначения для создания абстрактной модели системы, называемой UML-моделью. UML был создан для определения, визуализации, проектирования и документирования, в основном, программных систем. UML не является языком программирования, но на основании UML-моделей возможна генерация кода. Использование UML не ограничивается моделированием программного обеспечения. Его также используют для моделирования бизнес-процессов, системного проектирования и отображения организационных структур. UML позволяет также разработчикам программного обеспечения достигнуть соглашения в графических обозначениях для представления общих понятий (таких как класс, компонент, обобщение (англ. generalization), агрегация (англ. aggregation) и поведение) и больше сконцентрироваться на проектировании и архитектуре.

### ***практическое занятие (2 часа(ов)):***

Формальная спецификация последней версии UML 2.0 опубликована в августе 2005 года. Семантика языка была значительно уточнена и расширена для поддержки методологии Model Driven Development ? MDD. Последняя версия UML 2.4.1 опубликована в августе 2011 года. UML 2.4.1 принят в качестве международного стандарта ISO/IEC 19505-1, 19505-2

## **Тема 7. Тема. Представления модели и диаграммы в UML.**

### ***лекционное занятие (1 часа(ов)):***

В UML используются диаграммы: Structure Diagrams: Class diagram Component diagram Composite structure diagram Collaboration (UML2.0) Deployment diagram Object diagram Package diagram Profile diagram (UML2.2) Структурные диаграммы: Диаграмма классов Диаграмма компонентов Диаграмма композитной/составной структуры Диаграмма кооперации (UML2.0) Диаграмма развёртывания Диаграмма объектов Диаграмма пакетов Диаграмма профилей (UML2.2)

### ***практическое занятие (2 часа(ов)):***

В UML используются диаграммы (продолжение): Behavior Diagrams: Activity diagram State Machine diagram Use case diagram Interaction Diagrams: Communication diagram (UML2.0) / Collaboration (UML1.x) Interaction overview diagram (UML2.0) Sequence diagram Timing diagram (UML2.0) Диаграммы поведения: Диаграмма деятельности Диаграмма состояний Диаграмма вариантов использования Диаграммы взаимодействия: Диаграмма коммуникации (UML2.0) / Диаграмма кооперации (UML1.x) Диаграмма обзора взаимодействия (UML2.0) Диаграмма последовательности Диаграмма синхронизации (UML2.0)

## **Тема 8. Тема. Статическое представление модели. Классификаторы и их отношения.**

### ***лекционное занятие (1 часа(ов)):***

Статическое представление (Static view) отражает статическую структуру системы и не описывает динамику в любом ее проявлении. Чаще всего, статическое представление отражает логические концепции приложения, основой которого служат классы и их отношения.

### ***практическое занятие (1 часа(ов)):***

К высокоуровневым составляющим статического представления относятся классификаторы (класс, интерфейс, тип данных), отношения (ассоциация, обобщение, зависимость, реализация), ограничения и комментарии. Кроме этого, в качестве организационных блоков в этом представлении находятся пакеты и подсистемы. Все прочие составляющие второстепенны по отношению к этим элементам и располагаются внутри них.

### **Тема 9. Тема. Классы в UML. Диаграммы классов. Абстрактные, корневые, листовые, полиморфные элементы. Кратность, атрибуты, операции.**

#### **лекционное занятие (1 часа(ов)):**

Диаграмма классов показывает набор классов, интерфейсов и коопераций, а также их связи. Чаще других диаграмм используется при моделировании объектно-ориентированных систем. Иллюстрирует статическое представление дизайна системы. Диаграмма классов, которая включает активные классы, определяет статический взгляд на процессы системы.

#### **практическое занятие (2 часа(ов)):**

Диаграмма классов - это графическое представление статической модели. Обычно для того, чтобы отобразить всю статическую модель целиком, требуется несколько диаграмм классов. Наличие отдельных диаграмм классов не указывает на то, что сама модель делится при этом на части. При создании диаграмм, как правило, пользуются логическими разделителями модели, то есть пакетами, которые представляют собой естественные внутренние границы статической модели.

### **Тема 10. Тема. Моделирование поведения. Взаимодействия. Прецеденты. Диаграммы прецедентов. Диаграммы взаимодействия (диаграммы последовательностей, диаграммы кооперации). Диаграмма деятельности.**

#### **лекционное занятие (1 часа(ов)):**

Диаграмма использования представляет собой граф актантов, множество вариантов использования, заключенное в границы-системы (прямоугольник), ассоциации между актантами и вариантами использования, отношения между вариантами использования и обобщения между актантами. На диаграмме использования изображены элементы модели вариантов использования - сами варианты и актанты.

#### **практическое занятие (2 часа(ов)):**

Выражение, служащее для описания деятельности, является алгоритмом, выраженным на языке программирования или другом формальном языке. Впрочем, выражение может быть описано и естественным языком. Правда, в этом случае его нельзя будет проверять на наличие ошибок, и обрабатывать программными средствами, Впрочем, на начальном этапе разработки этого может быть вполне достаточно. Выражение также может описывать длительные процессы, происходящие в реальном мире.

### **Тема 11. Тема. Представление в виде конечного автомата. Конечный автомат. Событие. Состояние Переход.**

#### **лекционное занятие (1 часа(ов)):**

state machine (конечный автомат) Спецификация последовательности состояний, через которые проходит в течение своей жизни объект, или взаимодействие в ответ на происходящие события (а также ответные действия объекта на эти события). Конечный автомат прикреплен к исходному элементу (классу, кооперации или методу) и служит для определения поведения его экземпляров.

#### **практическое занятие (1 часа(ов)):**

В конечном автомате событие может запустить переход в новое состояние. У такого события есть список (возможно, пустой) параметров которые несут информацию от источника события к его получателю. У каждого события есть неявный параметр - время, когда оно происходит. Все прочие параметры представляют собой часть определения события. У экземпляра события есть аргументы (действительные значения), соответствующие параметрам события. Значение каждого параметра доступно действию, прикрепленному к переходу, который запускается этим событием.

### **Тема 12. Тема. Архитектурное моделирование. Диаграммы компонентов. Диаграммы развертывания.**

#### **практическое занятие (1 часа(ов)):**

deployment diagram (диаграмма развертывания) Диаграмма, на которой изображается конфигурация для работающих узлов и экземпляров компонента, а также объектов, которые на них существуют. Компоненты представляют собой выполнимые блоки программного кода. Компоненты, которые не существуют как единицы времени выполнения (в том случае, если они не были задействованы при компиляции), на этих диаграммах не указываются. Их место - на диаграммах компонентов. На диаграмме развертывания изображаются экземпляры компонентов, в то время как на диаграмме компонентов показаны определения их типов. В представлении развертывания содержатся экземпляры узлов, соединенные между собой коммуникационными связями. Узлы могут содержать экземпляры времени выполнения, такие как экземпляры компонентов и объекты. Экземпляры компонентов и объекты могут, в свою очередь, содержать в себе прочие объекты. Модель может демонстрировать зависимости, существующие между экземплярами и их интерфейсами. Кроме этого, она может отображать миграцию сущностей между узлами и другими контейнерами. У представления развертывания есть две формы - форма дескриптора и форма экземпляра.

### **Тема 13. Тема. Визуальное моделирование с помощью IBM Rational Rose.**

#### ***лекционное занятие (1 часа(ов)):***

IBM Rational Rose Developer for Visual Studio предоставляет набор функций, управляемых моделью, для разработки приложений на основе Microsoft Visual Studio. Это решение позволяет ускорить разработку приложений благодаря созданию кода из визуальных моделей с помощью UML (Unified Modeling Language).

#### ***практическое занятие (1 часа(ов)):***

В основе данного инструмента лежит унифицированный язык визуального моделирования Unified Modeling Language (UML). Введение подобного стандарта способствует большей эффективности организации проектной деятельности в команде и достижению понимания всеми участниками проекта по аспектам реализации конечного продукта. Например, аналитики могут использовать модели для согласования с заказчиками основных функций программного продукта и построения иерархии требований, архитекторы ? для создания гибкой и надежной архитектуры, разработчики ? для понимания того, что и как требуется реализовать в виде программного кода, а тестировщики ? для построения сценариев тестирования. Проектировщики баз данных могут с помощью IBM Rational Rose визуально проектировать и генерировать базы данных. При этом поддерживаются стандарты ANSI SQL, Microsoft SQL Server, Oracle, IBM DB2 и др.

### **Тема 14. Тема. Генерация программного кода (C++, Java, XML) в IBM Rational Rose.**

#### ***лекционное занятие (1 часа(ов)):***

Практика генерации программного кода (C++, Java, XML) в IBM Rational Rose.

#### ***практическое занятие (1 часа(ов)):***

Генерация программного кода (C++, Java, XML).

### **Тема 15. Тема. Процесс прямого и обратного проектирования в IBM Rational Rose.**

#### ***лекционное занятие (2 часа(ов)):***

Прямое проектирование диаграммы классов производится следующим образом: Идентифицируйте правила отображения модели на один или несколько языков программирования по вашему выбору. Это допустимо делать как при работе над одним конкретным проектом, так и для вашей организации в целом. В зависимости от семантики выбранных языков, вероятно, придется отказаться от использования некоторых возможностей UML. Например, язык UML допускает множественное наследование, а язык программирования Smalltalk - только одиночное. В связи с этим можно или запретить авторам моделей пользоваться множественным наследованием (что сделает создаваемые модели зависимыми от языка), или разработать идиомы для трансляции таких расширенных возможностей в конструкции, поддерживаемые данным языком программирования (что усложнит отображение). Применяйте помеченные значения для спецификации языка программирования. Это можно сделать как на уровне индивидуальных классов (если нужна тонкая настройка), так и на более высоком уровне, например для пакетов или коопераций. Пользуйтесь инструментальными средствами для прямого проектирования моделей.

#### ***практическое занятие (1 часа(ов)):***

Обратное проектирование диаграммы классов осуществляется так: Идентифицируйте правила для преобразования из выбранного вами языка реализации. Это можно сделать на уровне проекта или организации в целом. С помощью инструментального средства укажите код, который вы хотите подвергнуть обратному проектированию. Воспользуйтесь этим средством для создания новой модели или для модификации ранее созданной. Пользуясь инструментальными средствами, создайте диаграмму классов путем опроса полученной модели. Можно начать, например, с одного или нескольких классов, а затем расширить диаграмму, следуя вдоль некоторых отношений или добавив соседние классы.

#### 4.3 Структура и содержание самостоятельной работы дисциплины (модуля)

N	Раздел Дисциплины	Семестр	Неделя семестра	Виды самостоятельной работы студентов	Трудоемкость (в часах)	Формы контроля самостоятельной работы
1.	Тема 1. Тема. Методологии объектно-ориентированного анализа и проектирования сложных систем. CASE Computer Aided Software Engineering	1	1	Изучение материала	2	Демонстрация
2.	Тема 2. Тема. Жизненный Цикл Программного Обеспечения. Модели Жизненного Цикла.	1	2	Водопадная (каскадная, последовательная) модель. Итерационная модель. Спиральная модель Сравнение	4	Демонстрация
3.	Тема 3. Тема. Методология быстрой разработки приложений RAD (Rapid Application Development).	1	3	RAD-продукты	2	Демонстрация
4.	Тема 4. Тема. Структурный подход к проектированию информационных систем. Основные модели.	1	4	Основные модели - изучение	2	Контрольный опрос
5.	Тема 5. Тема. Объектно-ориентированные CASE-средства.		5	Основные Объектно-ориентированные CASE-средства. Знакомство на сайте IBM	2	Демонстрация
6.	Тема 6. Тема. Universal Modeling Language (UML) обзор. Концептуальные области UML. Назначение моделей.	1	6	Знакомство с моделированием с помощью UML	4	Опрос

N	Раздел Дисциплины	Семестр	Неделя семестра	Виды самостоятельной работы студентов	Трудоемкость (в часах)	Формы контроля самостоятельной работы
7.	Тема 7. Тема. Представления модели и диаграммы в UML.	1	7-8	Изучение моделей	4	Демонстрация
8.	Тема 8. Тема. Статическое представление модели. Классификаторы и их отношения.	1	9	знакомство со статическим представлением	2	Опрос
9.	Тема 9. Тема. Классы в UML. Диаграммы классов. Абстрактные, корневые, листовые, полиморфные элементы. Кратность, атрибуты, операции.	1	10	Запись классов в UML	2	Демонстрация
10.	Тема 10. Тема. Моделирование поведения. Взаимодействия. Прецеденты. Диаграммы прецедентов. Диаграммы взаимодействия (диаграммы последовательностей, диаграммы кооперации). Диаграмма деятельности.	1	11-12	Знакомство с диаграммами раздела	2	Демонстрация
11.	Тема 11. Тема. Представление в виде конечного автомата. Конечный автомат. Событие. Состояние Переход.	1	13	Знакомство с динамическим представлением	2	Демонстрация
12.	Тема 12. Тема. Архитектурное моделирование. Диаграммы компонентов. Диаграммы развертывания.	1	14	Знакомство с диаграммами раздела	2	Контрольный опрос
13.	Тема 13. Тема. Визуальное моделирование с помощью IBM Rational Rose.	1	15	Практическая работа в среде	2	Демонстрация диаграмм



N	Раздел Дисциплины	Семестр	Неделя семестра	Виды самостоятельной работы студентов	Трудоемкость (в часах)	Формы контроля самостоятельной работы
14.	Тема 14. Тема. Генерация программного кода (C++, Java, XML) в IBM Rational Rose.	1	16	Практическая работа в среде	2	Демонстрация
15.	Тема 15. Тема. Процесс прямого и обратного проектирования в IBM Rational Rose.	1	17-18	Практическая работа в среде	2	Демонстрация
	Итого				36	

## 5. Образовательные технологии, включая интерактивные формы обучения

активные и интерактивные формы.

## 6. Оценочные средства для текущего контроля успеваемости, промежуточной аттестации по итогам освоения дисциплины и учебно-методическое обеспечение самостоятельной работы студентов

### Тема 1. Тема. Методологии объектно-ориентированного анализа и проектирования сложных систем. CASE Computer Aided Software Engineering

Демонстрация, примерные вопросы:

Доклад с презентацией

### Тема 2. Тема. Жизненный Цикл Программного Обеспечения. Модели Жизненного Цикла.

Демонстрация, примерные вопросы:

Доклад с презентацией

### Тема 3. Тема. Методология быстрой разработки приложений RAD (Rapid Application Development).

Демонстрация, примерные вопросы:

Доклад с презентацией

### Тема 4. Тема. Структурный подход к проектированию информационных систем. Основные модели.

Контрольный опрос, примерные вопросы:

Опрос на знание диаграмм

### Тема 5. Тема. Объектно-ориентированные CASE-средства.

Демонстрация, примерные вопросы:

Доклад с презентацией

### Тема 6. Тема. Universal Modeling Language (UML) обзор. Концептуальные области UML. Назначение моделей.

Опрос, примерные вопросы:

Опрос на знание языка

### Тема 7. Тема. Представления модели и диаграммы в UML.

Демонстрация, примерные вопросы:

Доклад с презентацией

### Тема 8. Тема. Статическое представление модели. Классификаторы и их отношения.

Опрос, примерные вопросы:

Контрольные задания

**Тема 9. Тема. Классы в UML. Диаграммы классов. Абстрактные, корневые, листовые, полиморфные элементы. Кратность, атрибуты, операции.**

Демонстрация, примерные вопросы:

Доклад с презентацией

**Тема 10. Тема. Моделирование поведения. Взаимодействия. Прецеденты. Диаграммы прецедентов. Диаграммы взаимодействия (диаграммы последовательностей, диаграммы кооперации). Диаграмма деятельности.**

Демонстрация, примерные вопросы:

Доклад с презентацией

**Тема 11. Тема. Представление в виде конечного автомата. Конечный автомат. Событие. Состояние Переход.**

Демонстрация, примерные вопросы:

Доклад с презентацией

**Тема 12. Тема. Архитектурное моделирование. Диаграммы компонентов. Диаграммы развертывания.**

Контрольный опрос, примерные вопросы:

Контрольные задания

**Тема 13. Тема. Визуальное моделирование с помощью IBM Rational Rose.**

Демонстрация диаграмм, примерные вопросы:

Демонстрация диаграмм, построенных в среде в режиме реального времени

**Тема 14. Тема. Генерация программного кода (C++, Java, XML) в IBM Rational Rose.**

Демонстрация, примерные вопросы:

Демонстрация в среде в режиме реального времени

**Тема 15. Тема. Процесс прямого и обратного проектирования в IBM Rational Rose.**

Демонстрация, примерные вопросы:

Демонстрация в среде в режиме реального времени

**Тема . Итоговая форма контроля**

Примерные вопросы к экзамену:

В течение семестра студенты делают доклады на семинарах по темам, перечисленным в приведенной программе.

**7.1. Основная литература:**

Технология программирования. Базовые конструкции C/C++, Липачёв, Евгений Константинович, 2012г.

Буч Г. Объектно-ориентированный анализ и проектирование с примерами приложений на C++. - М.: Бином, 1998. - 560 с.

2. Буч Г., Рамбо Д., Джекобсон А. Язык UML. Руководство пользователя. - М.: ДМК Пресс, 2001. - 432 с.

3. Рамбо Дж., Якобсон А., Буч Г. UML: специальный справочник. - СПб.: Питер, 2002. - 656 с.

4. Вендров А.М.. CASE-технологии. Современные методы и средства проектирования информационных систем.

5. Трофимов С.А. CASE-технологии: практическая работа в Rational Rose. - М.: Бином-Пресс, 2002. - 288 с.

**7.2. Дополнительная литература:**

Веб-технологии для математика: основы MathML, Елизаров, Александр Михайлович; Липачев, Евгений Константинович; Малахальцев, Михаил Арменович, 2010г.

1. Фаулер М, Скотт К. UML. Основы. - СПб.: Символ-Плюс, 2002. - 192 с.



2. Кватрани Т. Rational Rose 2000 и UML. Визуальное моделирование. - М.: ДМК Пресс, 2001. - 176 с.
3. Кватрани Т., Палистрант Дж. Визуальное проектирование с помощью IBM Rational Software Architect и UML - М.: КУДИС-ПРЕСС, 2007. - 192 с.
4. Boggs W., Boggs M. Mastering UML with Rational Rose 2002. - SYBEX Inc., 2002. - 714 p.

### **7.3. Интернет-ресурсы:**

citforum - <http://citforum.ru/programming/case.shtml>

IBM - <http://www.ibm.org>

Unified Modeling Language - <http://www.uml.org/>

Интернет-университет - <http://www.intuit.ru/>

Современные технологии управления бизнесом -

<http://www.piter-soft.ru/automation/more/glossary/process/CASE-tehnologii/>

### **8. Материально-техническое обеспечение дисциплины(модуля)**

Освоение дисциплины "Информационные системы на основе CASE-технологий" предполагает использование следующего материально-технического обеспечения:

Для проведения лекций необходимы: проектор, экран, ноутбук. Практические занятия проводятся с использованием персональных компьютеров, имеющих выход в Интернет. Лекции выставляются в Интернет - на сайт факультета.

Программа составлена в соответствии с требованиями ФГОС ВПО и учебным планом по направлению 010100.68 "Математика" и магистерской программе Геометрия и топология .

Автор(ы):

Липачев Е.К. \_\_\_\_\_

"\_\_" \_\_\_\_\_ 201\_\_ г.

Рецензент(ы):

"\_\_" \_\_\_\_\_ 201\_\_ г.