

**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ**  
**НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ**  
**МОСКОВСКИЙ ЭНЕРГЕТИЧЕСКИЙ ИНСТИТУТ**  
**ИНСТИТУТ АВТОМАТИКИ И ВЫЧИСЛИТЕЛЬНОЙ ТЕХНИКИ (АВТИ)**

---

Направление подготовки: **010400 Прикладная математика и информатика**

Магистерская программа:

**Математическое и программное обеспечение вычислительных машин и компьютерных сетей**

Квалификация (степень) выпускника: **магистр**

Форма обучения: **очная**

**РАБОЧАЯ ПРОГРАММА УЧЕБНОЙ ДИСЦИПЛИНЫ**  
**"CASE-ТЕХНОЛОГИИ РАЗРАБОТКИ ПРОГРАММНЫХ СРЕДСТВ"**

<b>Цикл:</b>	<b>Профессиональный</b>	
<b>Часть цикла:</b>	<b>По выбору</b>	
<b>№ дисциплины по учебному плану:</b>	<b>М 3.1.1</b>	
<b>Часов (всего) по учебному плану:</b>	<b>252</b>	
<b>Трудоемкость в зачетных единицах:</b>	<b>7</b>	<b>2 семестр</b>
<b>Лекции</b>	<b>36 часов</b>	<b>2 семестр</b>
<b>Практические занятия</b>		
<b>Лабораторные работы</b>	<b>36 часов</b>	<b>2 семестр</b>
<b>Расчетные задания, рефераты</b>		
<b>Объем самостоятельной работы по учебному плану (всего)</b>	<b>180 часов</b>	<b>2 семестр</b>
<b>Экзамен</b>		<b>2 семестр</b>
<b>Курсовой проект</b>	<b>72 часа</b>	<b>2 семестр</b>

**Москва - 2012**

## **1. ЦЕЛИ И ЗАДАЧИ ОСВОЕНИЯ ДИСЦИПЛИНЫ**

**Целью дисциплины** является изучение технологии реализации крупных программных проектов.

По завершению освоения данной дисциплины студент способен и готов:

- способность самостоятельно приобретать с помощью информационных технологий и использовать в практической деятельности новые знания и умения в том числе в новых областях знаний, непосредственно не связанных со сферой деятельности (ОК-4);
- способность углубленного анализа проблем, постановки и анализа задач научной и проектно-технологической деятельности (ПК-3);
- способность управлять проектами, планировать научно-исследовательскую деятельность, анализировать риски, управлять командой проекта (ПК-5);
- способность реализации решений, направленных на поддержку социально-значимых проектов, на повышение электронной грамотности населения, обеспечения общедоступности информационных услуг (ПК-14);
- применять современные технологии разработки программных средств с применением CASE-средств, контролировать качество разрабатываемых программных продуктов;
- организовывать работу и руководить коллективами разработчиков программных средств.

**Задачами дисциплины являются:**

- познакомить обучающихся с современными методологиями разработки программного обеспечения;
- познакомить обучающихся с современными технологиями разработки программного обеспечения;
- познакомить обучающихся с современными инструментальными системами, используемыми при разработке программного обеспечения;
- познакомить обучающихся с проблемами, возникающими при реализации крупных программных проектов.

## **2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ООП ВПО**

Дисциплина относится к вариативной части профессионального цикла М.3 основной образовательной программы подготовки магистров по профилю "Математическое и программное обеспечение вычислительных машин и компьютерных сетей" направления 010400 Прикладная математика и информатика.

Дисциплина базируется на следующих дисциплинах: «Технология разработки программных средств», «Основы программирования».

Знания, полученные по освоению дисциплины, необходимы при работе над магистерской диссертацией.

## **3. РЕЗУЛЬТАТЫ ОСВОЕНИЯ ДИСЦИПЛИНЫ**

В результате освоения учебной дисциплины обучающиеся должны демонстрировать следующие результаты образования:

**Знать:**

- современные методологии разработки программного обеспечения;
- современные технологии разработки программного обеспечения;
- современные архитектурные концепции, применяемые при разработке программных систем;

**Уметь:**

- применять современные CASE-средства;
- применять типовые архитектурные решения;
- создавать и внедрять стандарты кодирования и проектирования;
- анализировать архитектуру программных систем, оценивать уместность применения той или иной разновидности архитектурного решения.

**Владеть:**

- навыками планирования работ в программном проекте;
- навыками оценки рисков;
- навыками работы с современными средствами автоматизации.

## 4. СТРУКТУРА И СОДЕРЖАНИЕ ДИСЦИПЛИНЫ

### 4.1 Структура дисциплины

Общая трудоемкость дисциплины составляет 3 зачетных единицы, 108 часов.

№ п/п	Раздел дисциплины. Форма промежуточной аттестации (по семестрам)	Всего часов на раздел	Семестр	Виды учебной работы, включая самостоятельную работу студентов и трудоемкость (в часах)				Формы текущего контроля успеваемости (по разделам)
				лк	пр	лаб	сам.	
1	2	3	4	5	6	7	8	9
1	CASE-технология. CASE-средства. CASE-системы. Классификация CASE-средств.	4	2	2	--	--	2	Тест на знание терминологии
2	Жизненный цикл программного обеспечения. Модели жизненного цикла программного обеспечения.	4	2	2	--	--	2	Тест на знание терминологии и знание различных моделей жизненного цикла ПО
3	Этап анализа в жизненном цикле программного обеспечения. Методологические аспекты анализа целей и требований к разрабатываемому программному обеспечению. Техническое задание.	10	2	2	--	4	4	Тест на знание современных методологий
4	Структурные методологии и подходы к анализу и проектированию.	18	2	2	--	6	10	Тест на знание терминологии
5	Объектно-ориентированный анализ и проектирование	18	2	2	--	6	10	Тест на знание терминологии
6	Унифицированный язык моделирования	34	2	6	--	6	22	Подготовка реферата
7	Принципы проектирования сложных систем	12	2	2	--	--	10	Тест на знание терминологии
8	Надежность программного обеспечения (ПО). CASE-средства и надежность ПО.	11	2	2	--	4	5	Тест на знание терминологии

	Контроль качества ПО.							
9	Методология Rational Unified Process (RUP)	9	2	2	--	2	5	Тест на знание терминологии
10	Методология разработки программных систем MSF (Microsoft Solutions Framework).	30	2	6	--	4	20	Подготовка реферата
11	Гибкие и адаптивные методики разработки (Agile)	25	4	5	--	--	20	Подготовка реферата
12	Дополнительные средства поддержки жизненного цикла программного обеспечения.	39	4	1	--	4	32	Подготовка реферата
	Зачет	2	2	--	--	--	2	
	Экзамен	36	2	-	-	-	36	
	<b>Итого:</b>	<b>252</b>	<b>2</b>	<b>36</b>	<b>-</b>	<b>36</b>	<b>180</b>	
	<b>Курсовой проект</b>	<b>108</b>	<b>-</b>	<b>-</b>	<b>-</b>	<b>-</b>	<b>108</b>	

## 4.2 Содержание лекционно-практических форм обучения

### 4.2.1. Лекции:

#### 2 семестр

Тема 1. CASE-технология. CASE-средства. CASE-системы. Классификация CASE-средств.

Области применения CASE-технологий. Классификация технологии разработки программных систем. Процесс разработки программного обеспечения с использованием CASE-средств. Информационная инженерия и обратное перепроектирование. Методология RAD.

Тема 2. Жизненный цикл программного обеспечения. Модели жизненного цикла программного обеспечения.

Понятие жизненного цикла ПО. Каскадная модель жизненного цикла. Спиральная модель. Итеративная модель жизненного цикла. Достоинства и недостатки различных моделей жизненного цикла ПО.

Тема 3. Этап анализа в жизненном цикле программного обеспечения. Методологические аспекты анализа целей и требований к разрабатываемому программному обеспечению.

Этап анализа в жизненном цикле. Методы сбора и анализа требований. Методологические аспекты анализа целей и требований к разрабатываемому программному обеспечению.

Тема 4. Структурные методологии и подходы к анализу и проектированию.

Функционально-ориентированное (структурное) проектирование программного обеспечения. Диаграммные методологии проектирования программного обеспечения. CASE-средства поддержки структурных методологий. Проектирование, ориентированное на данные. Методология DATARUN проектирования информационных систем.

Тема 5. Объектно-ориентированный анализ и проектирование

Методики объектно-ориентированного анализа и проектирования. Классификация, основные этапы и задачи объектно-ориентированных методов анализа и проектирования. Язык моделирования UML. Синтаксис. Семантика. Пакеты. Канонические диаграммы.

Тема 6. Унифицированный язык моделирования.

Язык моделирования UML. Синтаксис. Семантика. Пакеты. Канонические диаграммы. Диаграммы вариантов использования и сценарии. Диаграммы классов и их использование. Диаграммы кооперации и диаграммы последовательности. Диаграммы состояний и диаграммы деятельности. Диаграммы компонентов и диаграммы развертывания. Обзор UML 2.0

Тема 7. Принципы проектирования сложных систем

Принципы проектирования сложных систем. Декомпозиция. Абстракция. Повторное использование. Правила разбиения системы на подсистемы и модули.

Тема 8. Надежность программного обеспечения (ПО). CASE-средства и надежность ПО. Контроль качества ПО.

Причины возникновения ошибок при разработке программных средств. CASE-модель жизненного цикла программного обеспечения. Системы автоматизации тестирования программных средств.

Тема 9. Методология Rational Unified Process (RUP)

Методология объектно-ориентированной разработки RUP (Ration Unified Process). Унифицированный процесс Rational. Фазы и дисциплины. Жизненный цикл проекта в унифицированном процессе Rational.

#### Тема 10. Методология разработки программных систем MSF (Microsoft Solutions Framework).

Методология разработки программных систем MSF (Microsoft Solutions Framework). Модель процессов в MSF. Этап анализа в MSF. Этап планирования в MSF. Этап разработки в MSF. Этапы контроля качества и внедрения в MSF. Модель команды разработчиков в MSF. Преимущества и недостатки. Управление проектом в MSF. Общий обзор. Дисциплина управления проектом. Масштабируемость. Иерархическая структура работ (WBS). Оценка сроков разработки. Составление календарного графика работ в проекте. Диаграммы вариантов использования системы и сценариев использования системы. Управление компромиссами в MSF. Стратегия выпуска версий в MSF.

#### Тема 11. Гибкие и адаптивные методики разработки (Agile)

Манифест Agile. Семейство методологий Agile. Методология XP - «экстремальное программирование» : особенности, преимущества, недостатки. Методологии FDD, OpenUP, Scrum.

#### Тема 12. Дополнительные средства поддержки жизненного цикла разработки программного обеспечения.

Дополнительные средства поддержки жизненного цикла разработки программного обеспечения. Классификация инструментальных систем. Системы управления задачами и заявками. Основные понятия. Обзор. Системы отслеживания ошибок (Bug-tracking). Системы управления версиями. Основные понятия. Обзор. Модели версионирования. Системы управления версиями. RCS. CVS. Системы управления версиями. SVN. Основные возможности. Внутренняя архитектура, компоненты. Системы управления версиями. Контроль версий. Планирование. Средства управления требованиями, конфигурационного управления, документирования, тестирования, поддержки коллективной разработки. Системы автоматизации сборки программных средств

### **4.2.2. Практические занятия учебным планом не предусмотрены.**

## **4.3. Лабораторные работы**

### **2 семестр**

- № 1. Разработка технического задания на программный продукт
- № 2. Применение систем управления требованиями на примере IBM Rational RequisitePro
- № 3. Использование методологии функционального моделирования IDEF0 для выполнения этапа анализа и проектирования программного обеспечения
- № 4. Использование методологии DFD для выполнения этапа анализа и проектирования программного обеспечения
- № 5. Использование методологии IDEF3 для выполнения этапа анализа и проектирования программного обеспечения
- № 6. Использование методологии ARIS для выполнения этапа анализа и проектирования
- № 7. Использование методологии онтологического моделирования IDEF5
- № 8. Использование методологии проектирования реляционных баз данных IDEF1X
- № 9. Использование унифицированного языка моделирования (UML) для объектно-ориентированного анализа и проектирования
- № 10. Поддержка средств моделирования и реинжиниринга в Microsoft Visual Studio
- № 11. Практическое применение систем контроля версий

- № 12. Автоматизация сборки с применением MSBuild
- № 13. Практическое применение систем контроля качества кода на основе метрик на примере Microsoft Visual Studio
- № 14. Применение средства контроля качества кода на базе правил
- № 15. Создание автоматизированных тестов пользовательского интерфейса с применением Microsoft Visual Studio 11
- № 16. Практическое применение средств автоматической генерации документации по исходному коду

#### **4.4. Расчетные задания учебным планом не предусмотрены.**

#### **4.5. Курсовые проекты и курсовые работы**

Темы курсовых проектов:

1. Средства автоматизации тестирования от IBM
2. Технология XSLT
3. Архитектура, управляемая моделью (Model-driven architecture, MDA)
4. Архитектура, управляемая моделью. Генератор кода Acceleo
5. Обзор архитектурных фреймворков.
6. Шаблоны проектирования
7. Сервис-ориентированная архитектура. Критический обзор.
8. Средства автоматизации тестирования - Rational Robot и Rational Functional Tester.
9. Сервис-ориентированная архитектура. IBM WebSphere.
10. Сервис-ориентированная архитектура. Решения Microsoft.
11. Сервис-ориентированная архитектура. Решения Oracle.
12. Архитектура, основанная на событиях (EDA)
13. Системы автоматизации тестирования, обзор, сравнение возможностей (рассмотреть минимум четыре системы)
14. Системы автоматической сборки, обзор, сравнение возможностей (рассмотреть минимум четыре системы)
15. Системы планирования, обзор, сравнение возможностей (рассмотреть минимум три системы)
16. Планирование работ и распределение нагрузки с помощью Gantt Project
17. Интегрированные Case системы, современное состояние, обзор.
18. Автоматизация сборки в Microsoft Visual Studio Team System 2012
19. Автоматизация тестирования в Microsoft Visual Studio Team System 2012
20. Автоматизация контроля качества кода в Microsoft Visual Studio Team System 2012
21. Возможности Microsoft Team Foundation Server
22. Средства кодогенерации, современное состояние, обзор.
23. Инструментальные средства поддержки методологии ARIS.
24. Microsoft Dynamics – Обзор. Microsoft Dynamics AX. Microsoft Dynamics NAV. Microsoft Dynamics CRM.
25. Системы help-desk, обзор, сравнение возможностей (рассмотреть минимум три системы)
26. Системы автоматической сборки, обзор, сравнение возможностей (рассмотреть минимум четыре системы)
27. Системы планирования, обзор, сравнение возможностей
28. MSF основные положения.
29. MOF основные положения.
30. RUP и MSF сравнение, отличия, преимущества, недостатки.
31. Средства поддержки процесса тестирования в Microsoft Visual Studio 2012



32. Профилирование кода и анализ производительности кода в Microsoft Visual Studio 2012
33. Разработка архитектуры приложений в Microsoft Visual Studio 2012
34. Модульное тестирование и анализ области покрытия исходного кода тестами в Microsoft Visual Studio 2012
35. Обзор новых возможностей Microsoft Visual Studio 2012
36. Автоматизация тестирования с помощью IBM Rational Performance Tester
37. Автоматизация тестирования с помощью IBM Rational Functional Tester
38. Обзор IBM Rational Team Concert 4.0
39. Обзор возможностей среды разработки Xcode 4
40. Обзор Microsoft Robotics Developer Studio
41. JetBrains Resharper
42. JetBrains YouTrack
43. JetBrains TeamCity
44. JetBrains dotTrace, dotCover, dotPeek
45. Microfocus Caliber
46. Система автоматизации тестирования Microfocus Silk
47. JetBrains Meta Programming System
48. Система контроля версий и управления Microfocus StarTeam
49. Visual AssistX
50. DevExpress CodeRush
51. Atlassian FishEye
52. Atlassian Baboo
53. Atlassian Clower
54. Atlassian Crucible
55. Atlassian Bonfire
56. Обзор типов тестирования
57. Средства поддержки процесса тестирования
58. Методология Test Driven Development
59. Microfocus SilkTest
60. Средства автоматизации тестирования HP QuickTest Professional, HP LoadRunner, HP Quality Center
61. Средство автоматизации тестирования TestComplete
62. Технология модульного тестирования
63. Свободно распространяемые средства автоматизации тестирования
64. Codeplex White Framework
65. Современные методы генерации кода по шаблону
66. Шаблоны проектов и генерация каркасов типовых приложений в Eclipse
67. Шаблоны проектов и генерация каркасов типовых приложений в Microsoft Visual Studio
68. Text Template Transformation Toolkit
69. Применение макропроцессоров для генерации кода. Макропроцессор m4
70. Visual Studio vs Eclipse: Сравнение средств автоматизации и кодогенерации.
71. Современные информационные среды накопления и модификации знаний.
72. Системы корпоративной памяти.
73. Microsoft Visual Studio Team System 2010 как интегрированное CASE-средство
74. Microsoft FxCop
75. Средства кодогенерации (по модели), современное состояние, обзор.
76. Разработка приложений в Rational Software Architect.
77. Rational PurifyPlus. Поиск и устранение ошибок работы с памятью. Анализ области покрытия кода. Анализ производительности кода.

78. Система управления версиями Rational ClearCase. Установка и практическое использование.
79. Система управления требованиями Rational RequisitePro. Возможности, особенности практического применения.
80. Система управления изменениями Rational ClearQuest.
81. Rational Portfolio Manager. Возможности, особенности практического применения.
82. Системы автоматизации тестирования, обзор, сравнение возможностей (рассмотреть минимум четыре системы)
83. Системы автоматической сборки, обзор, сравнение возможностей (рассмотреть минимум четыре системы)
84. Системы планирования, обзор, сравнение возможностей (рассмотреть минимум три системы)
85. Интегрированные Case системы, современное состояние, обзор.
86. Автоматизация сборки в Microsoft Visual Studio Team System 2011
87. Автоматизация тестирования в Microsoft Visual Studio Team System 2011
88. Автоматизация контроля качества кода в Microsoft Visual Studio Team System 2011
89. Возможности Microsoft Team Foundation Server
90. Системы Bug-tracking-a, обзор, сравнение возможностей (рассмотреть минимум Bugzilla, JIRA, Mantis, Trac, TracStudio)

## **5. ОБРАЗОВАТЕЛЬНЫЕ ТЕХНОЛОГИИ**

**Лекционные занятия** проводятся в форме лекций с использованием презентаций и видео роликов. Презентации лекций содержат большое количество графических материалов и примеров. Выездная лекция-экскурсия в департаменте разработки ООО «ААМ Автоматик по теме «Система автоматизации сборки Component Builder» (2 часа).

**Лабораторные занятия** проводятся с применением компьютерной техники и современного программного обеспечения для анализа и моделирования бизнес-процессов.

**Самостоятельная работа** включает подготовку к тестам, оформление реферата и подготовку его презентации к защите, подготовку к зачету.

## **6. ОЦЕНОЧНЫЕ СРЕДСТВА ДЛЯ ТЕКУЩЕГО КОНТРОЛЯ УСПЕВАЕМОСТИ, ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ПО ИТОГАМ ОСВОЕНИЯ ДИСЦИПЛИНЫ**

Для текущего контроля успеваемости используются различные виды тестов, устный опрос, презентация реферата.

Аттестация по дисциплине – экзамен.

Оценка за освоение дисциплины, определяется на основе оценки, полученной студентом на экзамене.

В приложение к диплому вносится оценка за 8 семестр.

## **7. УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ**

### **7.1. Литература:**

#### **а) основная литература:**

1. Иванова Г.С. Технология программирования. – М.: КноРус, 2011 – 336 с.
2. Джек Гринфилд, Кит Шорт, Фабрики разработки программ. Поточковая сборка типовых приложений, моделирование, структуры и инструменты – М.:Издательский дом Вильямс, 2007. – 592 с.
3. Куриленко И.Е. Современные методологии разработки программных средств: учебное пособие - М.:Издательский дом МЭИ, 2011 - 112 с.

4. Чарнецки К., Айзенекер У. Порождающее программирование: методы, инструменты, применение. Для профессионалов. – СПб.: Питер, 2005. – 731с.: ил.
5. Microsoft Corporation Анализ требований и создание архитектуры решений на основе Microsoft .NET - Учебный курс MCSD (сертификационный экзамен 70-300) / Пер. с англ. - М.: Издательско-торговый дом "Русская редакция", 2004. - 416 стр.
6. Куриленко И.Е., Варшавский П.Р. Модульное тестирование : учебное пособие - М.:Издательский дом МЭИ, 2011 - 48 с.
7. М. Кон Scrum : гибкая разработка ПО. / Пер. с англ. - М.: ООО «Издательский дом Вильямс», 2011. - 576 стр.

**б) дополнительная литература:**

1. Дубейковский В. И. Эффективное моделирование с AllFusion Process Modeler - М.: ДИАЛОГ-МИФИ, 2007. - 384 с.
2. *David Whitehurst, Matt Raible The AppFuse Primer.* — 2008. — С. 23. — 214 с. — ISBN 9780974884349
3. *Stephen A. White - IBM and Derek Miers - BPM Focus, September 2008*
4. Методология функционального моделирования IDEF0: руководящий документ. - Официальное издание Госстандарта России
5. Методические рекомендации Федерального агентства по техническому регулированию и метрологии. Р50.1.028 – 2001. Методология функционального моделирования.
6. Уокер Ройс Управление проектами по созданию программного обеспечения, - М.: Лори, 2002. – 448 с.
7. *Крэг Ларман. Применение UML 2.0 и шаблонов проектирования = Applying UML and Patterns : An Introduction to Object-Oriented Analysis and Design and Iterative Development.* — 3-е изд. — М.: Вильямс, 2006. — 736 с.
8. Том Демарко, Тимоти Листер Человеческий фактор: успешные проекты и команды – М.: Символ-Плюс, 2005. – 250 с.
9. Одинцов И.О. Профессиональное программирование. Системный подход. – 2-е изд. перераб. и доп. – СПб.: БХВ-Петербург, 2004. – 624с.: ил.
10. Калянов Г.Н. CASE – технологии: консалтинг в автоматизации бизнес-процессов. М., "Лори", 2000.
11. Калянов Г.Н. CASE. Структурный системный анализ (автоматизация и применение). М., "Лори", 1996.
12. Розенберг Д., Скотт К. Применение объектного моделирования с использованием UML и анализ прецедентов: Пер. с англ. – М.: ДМК Пресс, 2002. – 160 с.,ил.
13. Кватрани Т. Rational Rose 2000 и UML. Визуальное моделирование: Пер. с англ. – М.: ДМК Пресс, 2001. – 176 с., ил.
14. Роб Томсетт Радикальное управление ИТ-проектами – М.: Лори, 2005. – 294 с.
15. Буч Г., Якобсон А., Рамбо Дж. UML. Классика CS. 2-е изд. / Пер. с англ.; Под общей редакцией проф. С. Орлова — СПб.: Питер, 2006. — 736 с.
16. Алан Шаллоуей, Джеймс Р. Тротт Шаблоны проектирования. Новый подход к объектно-ориентированному анализу и проектированию – М.: Издательский дом Вильямс, 2002. – 288 с.
17. Фаулер М., Скотт К. UML. Основы. – Пер. с англ. – СПб: Символ-Плюс, 2002. -192 с., ил.
18. Одинцов И.О. Профессиональное программирование. Системный подход. – 2-е изд. перераб. и доп. – СПб.: БХВ-Петербург, 2004. – 624с.: ил.
19. И. Грэхем, Объектно-ориентированные методы. Принципы и практика. 3-е издание. : Пер. с англ. – М.: Вильямс, 2004. – 880с.
20. Г.Буч, Объектно-ориентированный анализ и проектирование с примерами приложений на C++, 2-е изд. – М.: Бином, СПб: Невский диалект, 1998г – 560с.

21. Вендров А.М. Проектирование программного обеспечения экономических информационных систем. – М.: Финансы и статистика. 2005. — 544 с.
22. Якобсон А., Буч Г., Рамбо Дж. Унифицированный процесс разработки программного обеспечения. Пер. с англ. – СПб.: Питер, 2002. – 496 с.
23. Аллен Э. Типичные ошибки проектирования. Библиотека программиста – СПб.: Питер, 2003. – 224с.:ил.
24. Бек К. Экстремальное программирование. – СПб.: Питер, 2002. – 224 с.: ил.
25. Скотт К. Унифицированный процесс. Основные концепции. – М.: «Вильямс», 2002. – 160 с.: ил.
26. Вендров А.М. Один из подходов к выбору средств проектирования баз данных и приложений. "СУБД", 1995, №3.
27. Ed Yourdon, Just Enough Structured Analysis <http://www.yourdon.com/strucanalysis/>
28. Коналлен, Дж. Разработка Web-приложений с использованием UML. : Пер. с англ. – М.: Издательский дом «Вильямс», 2001. – 288с. : ил.
29. Гамма Э., Хелм Р., Джонсон Р., Влиссидес Дж. Приемы объектно-ориентированного проектирования. Паттерны проектирования. – СПб: Питер, 2006. - 366с.: ил
30. Шаллоуей А., Тротт, Дж. Р. Шаблоны проектирования. Новый подход к объектно-ориентированному анализу и проектированию: Пер. с англ. – М.: Издательский дом «Вильямс», 2002. – 288 с.: ил
31. Ларман К. Применение UML и шаблонов проектирования. : Пер. с англ. : Уч. Пос. – М.: Издательский дом «Вильямс». 2001. – 496с.: ил.
32. Влиссидес Дж. Применение шаблонов проектирования. Дополнительные штрихи.: Пер. с англ. – М.: Издательский дом «Вильямс», 2003. – 144с.: ил.
33. Фаулер М., Скотт к. UML в кратком изложении. Применение стандартного языка объектного моделирования: Пер. с англ. – М.: Мир, 1999. – 191 с., ил.
34. Астелс Д., Миллер Г., Новак М. Практическое руководство по экстремальному программированию. : Пер. с англ. – М.: «Вильямс», 2002. – 320с.: ил.
35. Боггс У., Боггс М. UML и Rational Rose 2002. Пер. с англ. – М.: Изд. «Лори». 2004. - 510 с.

## **7.2. Электронные образовательные ресурсы:**

### **а) лицензионное программное обеспечение и Интернет-ресурсы:**

IDS Sheer ARIS Express 2.2

Microsoft Office Word

Microsoft Office Visio

IBM Rational Rose

Microsoft Visual Studio

CodeGear RAD Studio

Component Builder

Microfocus StarTeam

Apache ANT

MSBuild

Atlassian Jira

Atlassian Confluence

Microsoft Project

[http://www.niios.ru/course\\_bpm/](http://www.niios.ru/course_bpm/)

[http://niios.ru/course\\_mpm/](http://niios.ru/course_mpm/)

[http://niios.ru/course\\_case/](http://niios.ru/course_case/)

<http://ru.wikipedia.org/wiki/BPMN>

<http://ru.wikipedia.org/wiki/IDEF3>

<http://ru.wikipedia.org/wiki/IDEF0>

<http://ru.wikipedia.org/wiki/DFD>

<http://ru.wikipedia.org/wiki/UML>

[http://en.wikipedia.org/wiki/Business\\_Process\\_Modeling\\_Notation](http://en.wikipedia.org/wiki/Business_Process_Modeling_Notation)

[http://ru.wikipedia.org/wiki/Atlassian\\_JIRA](http://ru.wikipedia.org/wiki/Atlassian_JIRA)

[http://ru.wikipedia.org/wiki/Atlassian\\_Confluence](http://ru.wikipedia.org/wiki/Atlassian_Confluence)

## **8. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ**

Для обеспечения освоения дисциплины необходимо наличие учебной аудитории, снабженной мультимедийными средствами для представления презентаций лекций и показа учебных фильмов, а также аудитории, оснащенной современной компьютерной техникой для выполнения лабораторных работ.

Программа составлена в соответствии с требованиями ФГОС ВПО и с учетом рекомендаций ПрООП ВПО по направлению подготовки 010400 «Прикладная математика и информатика»; профили:

Математическое и программное обеспечение вычислительных машин и компьютерных сетей.

### **ПРОГРАММУ СОСТАВИЛ:**

к.т.н., доцент

Куриленко И.Е.

### **"УТВЕРЖДАЮ":**

Зав. кафедрой Прикладной математики  
д.т.н., профессор

Еремеев А.П.