

«УТВЕРЖДАЮ»

Проректор по научной работе

Драгунов В.К.

« 16 » июня 2015 г.



Программа аспирантуры

Направление 09.06.01 Информатика и вычислительная техника

Направленность (специальность) 05.13.11 Математическое и программное обеспечение вычислительных машин, комплексов и компьютерных сетей

РАБОЧАЯ ПРОГРАММА

дисциплины по выбору

«Современные архитектуры крупных распределенных программных комплексов и систем»

Индекс дисциплины по учебному плану: Б1.В.ДВ.2.2

Всего: 108 часов

Семестр 3, в том числе

6 часов – контактная работа,
84 часа – самостоятельная работа,
18 часов – контроль

Программа составлена на основе федерального государственного образовательного стандарта высшего образования (уровень подготовки кадров высшей квалификации) по направлению подготовки 09.06.01 Информатика и вычислительная техника, утвержденного приказом Минобрнауки России от 30 июля 2014 г. № 875, и паспорта специальности, указанной в номенклатуре специальностей научных работников 05.13.11 Математическое и программное обеспечение вычислительных машин, комплексов и компьютерных сетей, утвержденной приказом Минобрнауки России от 25 февраля 2009 г. № 59.

ЦЕЛИ И ЗАДАЧИ ОСВОЕНИЯ ДИСЦИПЛИНЫ

Целью изучения дисциплины является освоение современных концепций, подходов и практик формирования архитектуры распределенных систем.

Задачами дисциплины являются:

- формирование представления о современных архитектурных шаблонах и концепциях;
- изучение современных языков описания архитектур, языков трансформации моделей и шаблонов проектирования;
- приобретение навыков использования полученных представлений при разработке промышленных распределенных систем.

В процессе освоения дисциплины **формируются следующие компетенции:**

- способность к критическому анализу и оценке современных научных достижений, генерированию новых идей при решении исследовательских и практических задач, в том числе в междисциплинарных областях (УК-1);
- готовность использовать современные методы и технологии научной коммуникации на государственном и иностранном языках (УК-2);

– владение культурой научного исследования, в том числе с использованием современных информационно-коммуникационных технологий (ОПК-2);

– владение методами проведения патентных исследований, лицензирования и защиты авторских прав при создании инновационных продуктов в области профессиональной деятельности (ОПК-7);

– владение методами создания программ и программных систем для параллельной и распределенной обработки данных, языками и инструментальными средствами параллельного программирования (ПК-3);

– способность выполнять анализ программ и программных систем, их эквивалентные преобразования, верификацию и тестирование (ПК-5);

– владение методами и средствами виртуализации и виртуального отображения реальности в современных распределенных системах (ПК-9).

ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ ОБРАЗОВАНИЯ, ФОРМИРУЕМЫЕ В РЕЗУЛЬТАТЕ ОСВОЕНИЯ ДИСЦИПЛИНЫ

В результате освоения дисциплины обучающийся должен демонстрировать **следующие результаты образования:**

знать:

– общую постановку задачи разработки архитектуры сложных распределенных систем (УК-1);

– виды и типы архитектур программных систем (ПК-9);

– языки описания архитектур программных систем (ОПК-2);

уметь:

– разрабатывать архитектуры крупных распределенных программных систем (ПК-3);

– анализировать, выбирать и применять типовые архитектуры распределенных систем и шаблоны проектирования (ПК-5);

владеть:

- терминологией и навыками ведения профессиональной дискуссии по соответствующей тематике (УК-2);
- навыками поиска информации о новых архитектурах и шаблонах проектирования (ОПК-2);
- навыками выбора эффективных решений на уровне применяемых при решении практических задач архитектурных шаблонов и шаблонов проектирования (ОПК-7).

КРАТКОЕ СОДЕРЖАНИЕ РАЗДЕЛОВ ДИСЦИПЛИНЫ

Современное состояние дел в области разработки программного обеспечения (4 часа самостоятельной работы)

Текущая ситуация на рынке разработки программного обеспечения. Факторы, тормозящие создание и внедрение новых IT-решений. Классические проблемы IT-инфраструктуры предприятий, которые развивали ее по историческим соображениям. Тенденции развития методов разработки программного обеспечения.

Архитектура. Типы архитектур. Клиент-серверная архитектура (5 часов самостоятельной работы)

1. Что такое программная архитектура.
2. Методики формирования программной архитектуры.
3. Использование архитектуры для достижения стратегических преимуществ.
4. Общее описание программной архитектуры.
5. Отличие архитектуры программного обеспечения от детального проектирования программного обеспечения.
6. Примеры архитектурных стилей и моделей.
7. Клиент-серверная архитектура.
8. Многоуровневая архитектура.
9. Прямой обмен данными.

10. Технологии COM и CORBA.
11. Проблемы поддержки и масштабирования клиент-серверных программ

Распределенные системы (5 часов самостоятельной работы)

1. Понятие системы. Определение системы.
2. Распределенная система.
3. Типы распределенных систем.
4. Примеры распределенных систем.
5. Требования к распределенным системам.
6. Архитектуры распределенных систем.
7. Распределенные базы данных.

Событийно-ориентированная архитектура (EDA)

(5 часов самостоятельной работы)

1. Концепция событийно-управляемой архитектуры. Предпосылки создания событийно-управляемой архитектуры.
2. Основные элементы программ, построенных согласно событийно-управляемой архитектуре.
3. Задачи системной шины.
4. Преимущества событийно-управляемой архитектуры.
5. Архитектура системы передачи сообщений.

Сервис-ориентированная архитектура (SOA)

(4 часа самостоятельной работы)

1. Обзор основных моментов сервис-ориентированной архитектуры. Сервис. Сервисная ориентация. Композитные приложения.
2. Жизненный цикл SOA.
3. Проектирование бизнес-логики в IBM WebSphere Business Modeler.

Введение в Windows Communication Foundation

(5 часов самостоятельной работы)

1. Концепция служебно-ориентированных приложений.
2. Web-сервис.
3. Язык описания web-сервисов.
4. Протокол SOAP.
5. Windows Communication Foundation.
6. WCF и SOA.
7. Канонические принципы WCF.
8. Практические принципы WCF.
9. Разработка сервис-ориентированных приложений с применением WCF.
10. Схема организации удаленного взаимодействия в WCF.
11. Структура WCF-сервиса.
12. Единая программная модель.

Архитектура, управляемая моделью (Model Driven Architecture, MDA)

(5 часов самостоятельной работы)

1. Концепция архитектуры, управляемой моделью.
2. Платформенно-независимая модель.
3. Платформенно-зависимая модель.
4. Процесс трансформации моделей.
5. Методология MDA.
6. Преимущества MDA.
7. SOA и MDA. Model Driven SOA.
8. Обзор Eclipse Modeling Framework (EMF).
9. Модели рабочих процессов. Windows Workflow Foundation.
10. Языки преобразования моделей. Графовые грамматики. Язык Query-View-Transformation. Язык преобразования моделей ATL.

Шаблоны проектирования (Design patterns)

(4 часа самостоятельной работы)

1. Понятие шаблона проектирования.
2. Классификация шаблонов проектирования.
3. Преимущества и недостатки применения шаблонов проектирования.

Основные шаблоны проектирования (5 часов самостоятельной работы)

1. Обзор основных шаблонов проектирования.
2. Шаблон делегирования (Delegation pattern).
3. Шаблон функционального дизайна (Functional design).
4. Неизменяемый объект (Immutable).
5. Интерфейс (Interface).
6. Маркер (Marker interface).
7. Контейнер свойств (Property Container).

Порождающие шаблоны проектирования

(5 часов самостоятельной работы)

1. Обзор порождающих шаблонов проектирования.
2. Абстрактная фабрика.
3. Строитель.
4. Фабричный метод.
5. Отложенная инициализация.
6. Объектный пул.
7. Прототип.
8. Получение ресурса есть инициализация.
9. Одиночка.
10. Мультитон.

Поведенческие шаблоны проектирования

(5 часов самостоятельной работы)

1. Обзор поведенческих шаблонов проектирования.
2. Цепочка ответственности (Chain of Responsibility).

3. Команда (Command, Action, Transaction).
4. Интерпретатор (Interpreter).
5. Итератор (Iterator, Cursor).
6. Посредник (Mediator).
7. Хранитель (Memento, Token).
8. Наблюдатель (Observer, Dependents, Publish-Subscribe, Listener).
9. Состояние (State).
10. Стратегия (Strategy).
11. Шаблонный метод (Template Method).
12. Пустой объект (Null Object).
13. Посетитель (Visitor).
14. Политика (Simple Policy).
15. Спецификация (Specification).
16. Подписчик событий (Event listener).

Шаблоны группы Model-view-controller (4 часа самостоятельной работы)

1. Обзор группы шаблонов Model-view-controller.
2. Model-view-controller.
3. Model-view-presenter.

Шаблон Repository (4 часа самостоятельной работы)

1. Обзор шаблона Repository.
2. Преимущества и недостатки шаблона.

Шаблоны параллельного программирования

(5 часов самостоятельной работы)

1. Обзор шаблонов параллельного программирования.
2. Активный объект (Active Object).
3. Задержка (Balking).
4. Взаимодействие посредством сообщений (Messaging).
5. Блокировка с двойной проверкой (Double checked locking).

6. Событийно-управляемая асинхронность (Event-based asynchronous).
7. Ведущий/ведомый (Leaders/followers).
8. Монитор (Monitor Object).
9. Реактор (Reactor).
10. Блокировка чтения/записи (Read write lock).
11. Планировщик (Scheduler).
12. Пул потоков (Thread pool).
13. Личная память потока (Thread-Specific Storage).
14. Однопоточное выполнение (Single Thread Execution).

Анти-паттерны (5 часов самостоятельной работы)

1. Анти-паттерны в программировании.
2. Анти-паттерны в управлении разработкой ПО.
3. Анти-паттерны в разработке ПО.
4. Анти-паттерны в объектно-ориентированном программировании.
5. Анти-паттерны в программировании.
6. Методологические анти-паттерны.

Архитектурные подходы искусственного интеллекта

(5 часов самостоятельной работы)

1. Обзор архитектурных подходов искусственного интеллекта.
2. Системы, основанные на правилах. Продукции. Схема вывода. Вывод в условиях наличия конфликтов – стратегии разрешения конфликтов. Преимущества продукционных систем. Недостатки продукционных систем.
3. Мультиагентные системы и их архитектуры. Определение агента. Свойства агента. Виды агентов. Архитектуры агентов. Кооперации агентов. Свойства мультиагентных систем. Отличие мультиагентной архитектуры от других архитектур.
4. Виртуальные организации.
5. Поиск-ориентированные архитектуры.
6. Системы, построенные вокруг планировщиков. Язык PDDL.

Языки описания архитектуры (4 часа самостоятельной работы)

1. Требования к языку описанию архитектуры.
2. Примеры языков описания архитектур: xAdl, Koala, Darwin, AADL.
3. Сравнение с UML.

Принципы проектирования крупных систем

(5 часов самостоятельной работы)

1. Принципы работы с сложными системами.
2. Абстракция.
3. Уточнение.
4. Модульная разработка. Принципы разбиения системы на модули. Выделение интерфейсов и сокрытие информации. Адекватность, полнота, минимальность и простота интерфейсов. Разделение ответственности. Слабая связность модулей и сильное родство функций в одном модуле.
5. Повторное использование.

ОЦЕНОЧНЫЕ СРЕДСТВА ДЛЯ КОНТРОЛЯ ОСВОЕНИЯ РЕЗУЛЬТАТОВ ОБРАЗОВАНИЯ ПО ДИСЦИПЛИНЕ

Промежуточная аттестация по итогам освоения дисциплины:

3 семестр – дифференцированный зачет.

Вопросы для самоконтроля и проведения зачета

1. Что такое программная архитектура?
2. Каковы методики формирования программной архитектуры?
3. Существует ли путь использования архитектуры для достижения стратегических преимуществ?
4. Раскройте отличие архитектуры программного обеспечения от детального проектирования программного обеспечения?
5. Дайте примеры архитектурных стилей и моделей.
6. Что такое клиент-серверная архитектура?

7. Что такое многоуровневая архитектура?
8. В чем отличие технологий COM и CORBA?
9. Раскройте проблемы поддержки и масштабирования клиент-серверных программ.
8. Что такое система?
9. Что такое распределенная система?
10. Перечислите типы распределенных систем.
11. Дайте примеры распределенных систем.
12. Укажите требования к распределенным системам.
13. Перечислите типовые архитектуры распределенных систем.
14. Что такое распределенные базы данных.
15. Перечислите основные элементы программ, построенных согласно событийно-управляемой архитектуре.
16. Каковы задачи системной шины?
17. В чем преимущества событийно-управляемой архитектуры?
18. Что такое сервисная ориентация?
19. Что такое композитные приложения?
20. Опишите жизненный цикл SOA.
21. Концепция служебно-ориентированных приложений.
22. Web-сервис.
23. Язык описания web-сервисов.
24. Протокол SOAP.
25. Windows Communication Foundation.
26. WCF и SOA.
27. Канонические принципы WCF.
28. Практические принципы WCF.
29. Разработка сервис-ориентированных приложений с применением WCF.
30. Схема организации удаленного взаимодействия в WCF.
31. Структура WCF-сервиса.
32. Единая программная модель.

33. Концепция архитектуры, управляемой моделью.
34. Платформенно-независимая модель.
35. Платформенно-зависимая модель.
36. Процесс трансформации моделей.
37. Методология MDA.
38. Преимущества MDA.
39. SOA и MDA. Model Driven SOA.
40. Обзор Eclipse Modeling Framework (EMF).
41. Модели рабочих процессов. Windows Workflow Foundation.
42. Языки преобразования моделей. Графовые грамматики. Язык Query-View-Transformation. Язык преобразования моделей ATL.
43. Понятие шаблона проектирования.
44. Классификация шаблонов проектирования.
45. Преимущества и недостатки применения шаблонов проектирования.
46. Шаблон делегирования (Delegation pattern).
47. Шаблон функционального дизайна (Functional design).
48. Неизменяемый объект (Immutable).
49. Интерфейс (Interface).
50. Маркер (Marker interface).
51. Контейнер свойств (Property Container).
52. Обзор порождающих шаблонов проектирования.
53. Абстрактная фабрика.
54. Строитель.
55. Фабричный метод.
56. Отложенная инициализация.
57. Объектный пул.
58. Прототип.
59. Получение ресурса.
60. Одиночка.
61. Мультитон.

62. Обзор поведенческих шаблонов проектирования.
63. Цепочка ответственности (Chain of Responsibility).
64. Команда (Command, Action, Transaction).
65. Интерпретатор (Interpreter).
66. Итератор (Iterator, Cursor).
67. Посредник (Mediator).
68. Хранитель (Memento, Token).
69. Наблюдатель (Observer, Dependents, Publish-Subscribe, Listener).
70. Состояние (State).
71. Стратегия (Strategy).
72. Шаблонный метод (Template Method).
73. Пустой объект (Null Object).
74. Посетитель (Visitor).
75. Политика (Simple Policy).
76. Спецификация (Specification).
77. Подписчик событий (Event listener).
78. Отличие Model-view-controller от Model-view-presenter.
79. Обзор шаблона Repository. Преимущества и недостатки шаблона.
80. Активный объект (Active Object).
81. Задержка (Balking).
82. Взаимодействие посредством сообщений (Messaging).
83. Блокировка с двойной проверкой (Double checked locking).
84. Событийно-управляемая асинхронность (Event-based asynchronous).
85. Ведущий/ведомый (Leaders/followers).
86. Монитор (Monitor Object).
87. Реактор (Reactor).
88. Блокировка чтения/записи (Read write lock).
89. Планировщик (Scheduler).
90. Пул потоков (Thread pool).
91. Личная память потока (Thread-Specific Storage).
92. Однопоточное выполнение (Single Thread Execution).

93. Анти-паттерны в управлении разработкой ПО.
94. Анти-паттерны в разработке ПО.
95. Анти-паттерны в объектно-ориентированном программировании.
96. Анти-паттерны в программировании.
97. Методологические анти-паттерны.
98. Дайте обзор архитектурных подходов искусственного интеллекта.
99. Каковы требования к языку описанию архитектуры.
100. Какие принципы проектирования крупных систем вы знаете?

Критерии оценки за освоение дисциплины определены в Инструктивном письме И-23 от 14 мая 2012 г.

РЕКОМЕНДУЕМАЯ ЛИТЕРАТУРА

Основная литература:

1. Куриленко И.Е. Современные методологии разработки программных средств: учебное пособие. – М.: Изд. дом МЭИ, 2011. – 112 с.
2. Д. Эспозито. Знакомство с Microsoft ASP.NET 2.0: пер. с англ. – М.: Русская редакция; СПб.: Питер, 2006. – 512 с.
3. Осипов Г.С. Методы искусственного интеллекта. – М.: Физматлит, 2011. – 296 с.

Дополнительная литература:

4. Куриленко И.Е. Разработка программного обеспечения с помощью CASE-технологий: методическое пособие. – М.: Изд-во МЭИ, 2013. – 72 с.
5. Рассел С., Норвиг П. Искусственный интеллект: современный подход, 2-е изд.: Пер. с англ. – М.: Издательский дом «Вильямс», 2006. – 1408 с.
6. Люгер Д.Ф. Искусственный интеллект: стратегии и методы решения сложных проблем, 4-е издание.: Пер. с англ. – М.: Издательский дом «Вильямс», 2005. – 864 с.
7. Варшавский П.Р. Программное обеспечение интеллектуальных систем: учебное пособие. – М.: Изд. дом МЭИ, 2011. – 64 с.

Лицензионное программное обеспечение: Среда разработки Microsoft Visual Studio.