

**Правительство Российской Федерации**

**Федеральное государственное автономное образовательное учреждение  
высшего профессионального образования  
"Национальный исследовательский университет  
"Высшая школа экономики"**

Факультет бизнес-информатики

**Программа дисциплины**

**Распределенные вычисления**

для направления 231000.62 Программная инженерия  
подготовки бакалавра

Автор программы:

Замятина Е.Б., к.ф.-м.н., доцент, e\_zamyatina@mail.ru

Одобрена на заседании кафедры информационных технологий в бизнесе  
«23» мая 2014 г.

И.о. зав. кафедрой О.Л. Викентьева \_\_\_\_\_

Утверждена Учебно-методическим Советом НИУ ВШЭ – Пермь  
«29» мая 2014 г.

Председатель Г.Е. Володина \_\_\_\_\_

Пермь, 2014

*Настоящая программа не может быть использована другими подразделениями  
университета и другими вузами без разрешения кафедры-разработчика программы*



## 1. Область применения и нормативные ссылки

Настоящая программа учебной дисциплины устанавливает минимальные требования к знаниям и умениям студента и определяет содержание и виды учебных занятий и отчетности.

Программа предназначена для преподавателей, ведущих данную дисциплину, учебных ассистентов и студентов направления подготовки 231000.62 Программная инженерия, изучающих дисциплину «Распределенные вычисления».

Программа разработана в соответствии с:

- Образовательным стандартом федерального государственного автономного образовательного учреждения высшего профессионального образования Национального исследовательского университета «Высшая школа экономики» по направлению подготовки 231000.62 Программная инженерия (уровень подготовки бакалавр). Утверждён 02.07.2010 (протокол № 15), редакция 2011 г.
- Учебным планом университета по направлению подготовки 231000.62 Программная инженерия, утвержденным в 2012 г.

## 2. Цели освоения дисциплины

Целями освоения дисциплины «Распределенные вычисления» являются:

*В области обучения* – получение высшего профессионально профилированного (на уровне бакалавра) образования, позволяющего выпускнику успешно работать в избранной сфере деятельности, обладать универсальными и предметно-специализированными компетенциями, способствующими его социальной мобильности и устойчивости на рынке труда, а именно: изучение теоретических основ распределенных вычислений, получение практических навыков применения технологий распределенных вычислений при разработке распределенных программных систем различного назначения.

*В области воспитания* – развитие у студентов социально-личностных качеств: целеустремленности, организованности, трудолюбия, ответственности, самостоятельности, гражданственности, эмоционального интеллекта, приверженности этическим ценностям, коммуникативности, толерантности, повышение их общей культуры и мышления.

Для достижения целей при изучении дисциплины решаются следующие задачи:

- рассмотреть основные понятия теории распределенных вычислений;
- познакомиться с историей развития распределенных вычислений;
- познакомиться с различными архитектурами программных систем, которые поддерживают распределенные вычисления;
- познакомиться с новыми технологиями (компонентные, агентные, основанные на веб-сервисах, GRID, облачные);
- научиться работать со специальными программными средствами, реализующими распределенные вычисления.

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

В результате освоения дисциплины студент должен:

- **Иметь представление:**
  - о значении и областях применения распределенных вычислений;
  - о современных тенденциях развития распределенных вычислений.
- **Знать:**
  - особенности организации программных систем той или иной распределенной архитектуры;



- характерные особенности различных технологий, поддерживающих распределенные вычисления;
  - фундаментальные распределенные алгоритмы поиска, выбора лидера и др. для разработки распределенных приложений;
  - технологии распределенных приложений (клиент-серверные, сервис-ориентированные, агентные, облачные, GRID-технологии и др.);
  - особенности обработки данных в распределенных приложениях;
  - особенности поддержания отказоустойчивости распределенных систем.
- **Уметь:**
- разрабатывать распределенные приложения, используя знания о распределенных алгоритмах;
  - применять различные механизмы коммуникации при разработке распределенных приложений;
  - разрабатывать распределенные приложения, используя различные технологии (клиент-сервер, GRID, Cloud).

В результате освоения дисциплины студент осваивает следующие компетенции:

<b>Компетенция</b>	<b>Код по стандарту</b>	<b>Дескрипторы – основные признаки освоения (показатели достижения результата)</b>	<b>Формы и методы обучения, способствующие формированию и развитию компетенции</b>
Владение культурой мышления, способность к обобщению, анализу, восприятию информации, постановке цели и выбору путей её достижения (ОК–1);	ОК-1	Демонстрирует умение обосновывать предлагаемые решения, доказывать правильность используемых методов, анализировать и оценивать эффективность решений.	На аудиторных занятиях студентам предлагается проанализировать методы и алгоритмы распознавания, выбрать наилучший метод, обосновать свое решение
Стремление к саморазвитию, повышению своей квалификации и мастерства.	ОК-6	Демонстрирует способность самостоятельно определять формирующиеся дефициты знаний, умений и навыков в ходе обучения.	Самостоятельное изучение отдельных тем. Выполнение индивидуальных заданий (с получением консультаций преподавателя).
		Показывает умение сформулировать проблемы, связанные с недостатком знаний и навыков, и выбрать подходы к их решению.	
		Владеет знаниями, достаточными для самостоятельного изучения и понимания методов и алгоритмов.	
Умение применять основы информатики и программирования к проектированию, конструированию и тестированию программных продуктов;	ПК-10	Демонстрирует способность к применению навыков программирования и изученных технологий к разработке программных продуктов, реализующих распределенные алгоритмы и взаимодействие распределенных программных компонентов	Практические занятия. Выполнение индивидуальных заданий, требующих знаний в области распределенных вычислений



Компетенция	Код по стандарту	Дескрипторы – основные признаки освоения (показатели достижения результата)	Формы и методы обучения, способствующие формированию и развитию компетенции
Навыки использования различных технологий разработки программного обеспечения	ПК-16	Демонстрирует способность реализовать выбранные методы и алгоритмы, применяя новые технологии (в том числе, облачные) инструментальные средства (WCF, AZURE)	Выполнение индивидуальных заданий, связанных с реализацией распределенных приложений.

#### 4. Место дисциплины в структуре образовательной программы

Настоящая дисциплина относится к базовой части профессионального цикла.

Изучение данной дисциплины базируется на следующих дисциплинах:

- Информатика, математическая логика и теория алгоритмов.
- Конструирование программного обеспечения.
- Компонентно-ориентированное программирование.
- Операционные системы.
- Базы данных.
- Архитектура вычислительных систем.
- Обеспечение качества и тестирование.
- Проектирование и архитектура программных систем.
- Программирование.

Для освоения учебной дисциплины студенты должны владеть следующими знаниями:

- знание основ архитектуры ЭВМ;
- знание основ процедурного и объектно-ориентированного программирования;
- знание основ проектирования и архитектуры программных систем.

Основные положения дисциплины используются при написании выпускных квалификационных работ бакалавра и прохождении практик.

#### 5. Тематический план учебной дисциплины

№	Название раздела	Всего часов	Аудиторные часы			Самостоятельная работа
			Лекции	Семинары	Практические занятия	
	<b>Раздел 1. Основные принципы организации распределенных систем обработки информации</b>	<b>91</b>	<b>9</b>	<b>0</b>	<b>18</b>	<b>64</b>
1.	Тема 1. Теоретические предпосылки	9	1	0	2	6



	создания распределенных систем обработки информации					
2.	Тема 2. Задачи и свойства распределенных систем	13	1	0	2	10
3.	Тема 3. Достоинства и недостатки распределенных систем различных классов	9	1	0	2	6
4.	Тема 4. Особенности разработки программного обеспечения для распределенных систем	9	1	0	2	6
5.	Тема 5. Особенности реализации распределенных алгоритмов	11	1	0	2	8
6.	Тема 6. Клиент-серверная архитектура, основные понятия	9	1	0	2	6
7.	Тема 7. Компонентное программирование и сервис-ориентированная архитектура, основные понятия	7	1	0	2	4
8.	Тема 8. Распределенные интеллектуальные системы на основе агентов	11	1	0	2	8
9.	Тема 9. Распределенная обработка данных	13	1	0	2	10
	<b>Раздел 2. Фундаментальные распределенные алгоритмы</b>	<b>64</b>	<b>6</b>	<b>0</b>	<b>10</b>	<b>48</b>
10.	Тема 10. Волновые алгоритмы распространения информации	26	2	0	4	20
11.	Тема 11. Волновые алгоритмы обхода сайтов	20	2	0	4	14
12.	Тема 12. Алгоритмы выбора лидера	18	2	0	2	14
	<b>Раздел 3. Технологии распределенной обработки данных</b>	<b>97</b>	<b>5</b>	<b>0</b>	<b>12</b>	<b>80</b>
13.	Тема 13. Технологии клиент-сервер	44	2	0	6	36
14.	Тема 14. Технологии WCF	27	1	0	2	24
15.	Тема 15. GRID-технология	9	1	0	0	8
16.	Тема 16. Cloud-технологии	17	1	0	4	12
	<b>Итого:</b>	<b>252</b>	<b>20</b>	<b>0</b>	<b>40</b>	<b>192</b>

## 6. Формы контроля знаний студентов

Тип контроля	Форма контроля	1 год				Параметры
		1	2	3	4	
Текущий (неделя)	Домашнее задание			9		Отчет о выполнении домашнего задания
Итоговый	Экзамен			*		Письменный экзамен (80 минут)

### 6.1. Критерии оценки знаний, навыков

Текущий контроль предусматривает выполнение домашнего задания, заключающегося в разработке распределенного приложения. Оценки по всем формам текущего контроля выставляются по 10-ти балльной шкале.

Тематика заданий домашнего задания приведена в пункте 9.1.



Критерии оценки выполнения заданий:

Требование к заданию	Максимальное количество баллов
Распределенное приложение спроектировано с учетом особенностей предметной области. Выбрана наиболее подходящая модель распределенной системы. Если используется распределенная база данных, то тиражирование данных имеет подходящий для данной предметной области механизм и архитектуру.	1
Архитектура системы является оптимальной для заданных при разработке критериев. В отчете присутствует обоснование выбора данного типа архитектуры.	1
Приложение обеспечивает параллельную работу нескольких клиентов и серверов, в том числе на одном компьютере. Серверы распределенной системы выполняют различные функции.	2/3
Приложение является масштабируемым, позволяет добавлять новых участников взаимодействия без переписывания кода и перезапуска приложений.	2/3
Существует возможность динамической балансировки загрузки системы.	5/3
Для организации взаимодействия компонент распределенной системы используется не менее четырех различных средств коммуникации. В отчете присутствует четкое обоснование выбора средств взаимодействия для каждого конкретного случая.	4/3
Система является отказоустойчивой. В случае если один и/или несколько компонент системы аварийно завершают свою работу.	1
Распределенное приложение продолжает работать и в случае, если после аварийного завершения некоторого компонента, он восстановлен на другом узле вычислительной сети.	1
Отчет содержит подробное описание архитектуры каждого компонента распределенного приложения.	1
В отчете описана структура передаваемых данных, формат сообщений и вид протокола, используемого для этого.	2/3

Предусматривается возможность «защиты» выполненного домашнего задания. В ходе защиты студент должен продемонстрировать знание профессиональной терминологии в рамках соответствующей темы, продемонстрировать знание теоретического материала по теме, а также умение оценивать эффективность решений.

Итоговый экзамен включает как теоретические вопросы, так и практические задания по темам всего курса. Примерный перечень вопросов для подготовки к экзамену по дисциплине представлен в разделе 9.2.

### 6.1. Порядок формирования оценок по дисциплине

Преподаватель оценивает работу студентов на практических занятиях: активность студентов при ответах на вопросы преподавателя, правильность решение задач. Оценки за работу на практических занятиях преподаватель выставляет в рабочую ведомость. Оценка по 10-ти балльной шкале за работу на семинарских и практических занятиях определяется перед промежуточным или итоговым контролем и называется –  $O_{аудиторная}$ .

**Оценка за текущий контроль** ( $O_{текущий}$ ) рассчитывается как взвешенная сумма всех форм текущего контроля:

$$O_{текущий} = n_1 \cdot O_{дз},$$

при этом  $n_1 = 1,0$ .



Способ округления оценки за текущий контроль: арифметический.

**Накопленная оценка** за текущий контроль учитывает результаты студента по текущему контролю следующим образом:

$$O_{\text{накопленная}} = 2/3 * O_{\text{текущий}} + 1/3 * O_{\text{аудиторная}}.$$

Способ округления накопленной оценки текущего контроля: арифметический.

**Результирующая оценка** за дисциплину рассчитывается следующим образом:

$$O_{\text{результирующая}} = 0,6 * O_{\text{накопленная}} + 0,4 * O_{\text{экс}}$$

Способ округления накопленной оценки итогового контроля в форме экзамена: арифметический.

На пересдаче студенту не предоставляется возможность получить дополнительный балл для компенсации оценки за текущий контроль.

На экзамене студент может получить дополнительный вопрос (дополнительную практическую задачу, выполнить к пересдаче домашнее задание), ответ на который оценивается в 1 балл.

В диплом выставляется результирующая оценка по учебной дисциплине, которая формируется равной результирующей оценке ( $O_{\text{результирующая}}$ ).

## 7. Содержание дисциплины

### Раздел 1. Основные принципы организации распределенных систем обработки информации

#### **Тема 1. Теоретические предпосылки создания распределенных систем обработки информации**

Использование ресурсов нескольких вычислительных узлов. Расширяемость. Повышение производительности. Совместная работа географически удаленных друг от друга пользователей. Повышение надежности системы.

#### **Тема 2. Задачи и свойства распределенных систем**

Назначение распределенных систем. Основные классы задач, которые позволяют решить распределенные вычисления. Проблема распределения функций и данных между узлами сети. Свойства распределенных систем: масштабируемость, логическая целостность данных, безопасность, отказоустойчивость, эффективность.

#### **Тема 3. Достоинства и недостатки распределенных систем различных классов**

Классификация распределенных вычислительных систем: локальные сети, глобальные сети, мультимастерные вычислительные системы, взаимодействующие процессы, которые выполняются на одном компьютере.

Характеристики локальных и глобальных вычислительных систем: параметры надежности, время, затрачиваемое на передачу сообщения, однородность, безопасность.

Назначение многопроцессорных компьютеров.

Кластер. Вычислительные системы корпоративного уровня.

Глобальные системы (грид-система). Достоинства и недостатки.

Cloud-компьютинг. Достоинства и недостатки, особенности организации вычислений в облаке. Примеры использования.

#### **Тема 4. Особенности разработки программного обеспечения для распределенных систем**

Особенности реализации распределенного программного обеспечения для глобальных сетей, исходя из следующих характеристик:

- надежность обмена данными по типу точка-точка;
- выбор путей коммуникации;
- контроль перегрузок;
- предотвращение тупиков;
- безопасность.





Проблемы реализации программного обеспечения для локальных сетей. Проблемы распределённого управления процессами, выполняющимися на разных узлах.

#### **Тема 5. Особенности реализации распределенных алгоритмов**

Неполные знания о глобальном состоянии. Отсутствие глобального кадра времени. Недетерминизм.

#### **Тема 6. Клиент-серверная архитектура**

Основные понятия и особенности организации вычислительных систем с клиент-серверной архитектурой.

Структура вычислительных систем с архитектурой клиент-сервер.

#### **Тема 7. Компонентное программирование и сервис-ориентированная архитектура, основные понятия**

Основные понятия, связанные с компонентами, роль интерфейса.

Понятие сервиса, особенности организации систем с сервис-ориентированной архитектурой.

#### **Тема 8. Распределенные интеллектуальные системы на основе агентов**

Определение и особенности программного агента. Структура программного агента. Координация, кооперация, коммуникация агентов. Реактивные и интеллектуальные агенты. Примеры применения многоагентных систем.

#### **Тема 9. Распределенная обработка данных**

Распределенные базы данных, понятия, характеристики, классическая распределенная система БД. Общее с распределенными файловыми системами.

Критерии разделения данных между узлами сети.

Определение Дэйта, двенадцать свойств Дэйта.

Лекции: 9 часов.

Практические занятия: 18 часов.

Самостоятельная работа: 64 часа.

На лекционных и практических занятиях рассматриваются примеры распределенных систем, особенности их организации, особенности алгоритмов в распределенных системах. Время, отведенное для самостоятельных занятий, студенты используют для закрепления материала, изученного на лекциях и на практических занятиях.

#### **Литература по разделу:**

1. Афанасьев К.Е., Завозкин С.Ю., Трофимов С.Н., Власенко А.Ю. Основы высокопроизводительных вычислений. – Кемерово: Кемеровский государственный университет, 2011. – 246 с.
2. Тель Ж. Введение в распределенные алгоритмы. М.: МЦНМО, 2009. – 616 с.
3. Таненбаум Э. Распределенные системы: принципы и парадигмы. – СПб: Питер, 2003. – 877 с.

### **Раздел 2. Фундаментальные распределенные алгоритмы**

#### **Тема 10. Волновые алгоритмы распространения информации**

Определение волнового алгоритма. Кольцевой волновой алгоритм. Древоидный волновой алгоритм. Алгоритм голосования. Алгоритм «Эхо». Алгоритм Финна.

#### **Тема 11. Волновые алгоритмы обхода сайтов**

Алгоритм обхода полного графа. Алгоритм обхода тора. Алгоритм обхода гиперкуба. Алгоритм Тарри.





## **Тема 12. Алгоритмы выбора лидера**

Алгоритм смещения. Алгоритм для древовидных структур. Алгоритм для кольцевых структур (Алгоритм Ле-Ланна, Алгоритм Чанга Робертса).

Лекции: 6 часов.

Практические занятия: 10 часов.

Самостоятельная работа: 48 часов.

На лекционных и практических занятиях рассматриваются классические распределенные алгоритмы, особенности их применения в распределенных системах. Время, отведенное для самостоятельных занятий, студенты используют для закрепления материала, изученного на лекциях и на практических занятиях и выполнения индивидуальных занятий.

### **Литература по разделу:**

1. Миков А.И., Замятина Е.Б. Распределенные системы и алгоритмы. – М.: Интернет-Университет Информационных Технологий, 2008. – 377 с.
2. Афанасьев К.Е., Завозкин С.Ю., Трофимов С.Н., Власенко А.Ю. Основы высокопроизводительных вычислений. – Кемерово: Кемеровский государственный университет, 2011. – 246 с.
3. Тель Ж. Введение в распределенные алгоритмы. М.: МЦНМО, 2009. – 616 с.
4. Таненбаум Э. Распределенные системы: принципы и парадигмы. – СПб: Питер, 2003. – 877 с.

## **Раздел 3. Технологии распределенной обработки данных**

### **Тема 13 Технологии клиент-сервер**

*Развитие архитектуры распределенных приложений:*

- прикладные программы с монолитной структурой;
- архитектура "клиент-сервер" со связями "один к одному";
- структура приложений "один к одному" для сети;
- многопоточковые серверы;
- архитектура с виртуальным сервером (диспетчером);
- архитектура с несколькими многопоточковыми серверами (multi-threaded, multi-servers architecture) и схемы организации взаимодействия клиентов и серверов.

*Требования к программированию приложений "клиент-сервер":*

- не использовать монолитное кодирование (приложения с модульной структурой, разрабатываемые "сверху вниз");
- глобальные переменные неприемлемы;
- "поклиентное обслуживание";
- использование архитектуры с "независимыми средствами";
- обеспечить переносимость через устранение зависимости от ОС и СУБД.

Архитектура сервера: коммуникационный модуль, диспетчер, исполнительные (обслуживающие, обрабатывающие) модули.

Архитектура клиента: коммуникационный модуль и интерфейс пользователя.

Режимы взаимодействия клиента и сервера (синхронный и асинхронный режим).

Поддержка технологий распределенных приложений на уровне ОС: средства управления процессами и потоками, синхронизация вычислений.

Варианты распределения взаимодействия между приложениями:

- файловая система;
- общий сервер, реализующий обслуживание клиентов;
- виртуальный сервер.

Преимущества и недостатки каждого варианта.



Подходы к формированию каталогов сервисов в сети:

- статическое;
- динамическое, от клиента по запросу функции;
- динамическое, от клиента по отказу сервера;
- динамическое, предложение от сервера;
- динамическое, с централизованным каталогом на одном узле.

#### **Тема 14. Технологии WCF**

Основные принципы технологии WCF. Преимущества и недостатки WCF. Распределенное исполнение. Сервис-ориентированная архитектура. Предпосылки появления WCF.

WCF-сервисы и клиенты. Параметры точки доступа. Конфигурирование WCF-сервисов. Связывание. Виды связывания.

#### **Тема 15. GRID-технология**

Понятие GRID-технологии. Примеры проектов, созданных с помощью данной технологии. Причины развития GRID. Решаемые с помощью данной технологии задачи. Типы GRID-систем. Компоненты хранилища данных Amazon.com. Инструментарий Globus Toolkit и его основные компоненты.

#### **Тема 16. Cloud-технологии**

Технологии Cloud-вычислений. Основные понятия Cloud-вычислений. Мультиэнтная архитектура. Технологии Cloud-вычислений.

Лекции: 1 час.

Практические занятия: 12 часов.

Самостоятельная работа: 8 часов.

На лекционных и практических занятиях рассматриваются технологии распределенных вычислений, особенности их применения. Практические занятия используются для подробного рассмотрения облачных технологий (AZURE). Отведенное на самостоятельную работу время используется для закрепления лекционного и практического материала и для выполнения индивидуальных заданий.

#### **Литература по разделу:**

1. Миков А.И., Замятина Е.Б. Распределенные системы и алгоритмы. – М.: Интернет-Университет Информационных Технологий, 2008. – 377 с.
2. Афанасьев К.Е., Завозкин С.Ю., Трофимов С.Н., Власенко А.Ю. Основы высокопроизводительных вычислений. – Кемерово: Кемеровский государственный университет, 2011. – 246 с.
3. Тель Ж. Введение в распределенные алгоритмы. М.: МЦНМО, 2009. – 616 с.
4. Таненбаум Э. Распределенные системы: принципы и парадигмы. – СПб: Питер, 2003. – 877 с.

## **8. Образовательные технологии**

Используется «проблемное» чтение лекций по дисциплине с использованием компьютерного мультимедийного оборудования.

Лекционные занятия предназначены для того, чтобы:

- создать теоретические предпосылки для практического применения методов, алгоритмов и технологий распределенного программирования при решении прикладных задач;



- познакомить студентов с особенностями использования технологий и методов распределенных вычислений в различных областях будущей профессиональной деятельности.

На практике используются инструментальные средства Microsoft Visual Studio, поддерживающие WCF-технологии, Microsoft AZURE.

### **8.1. Методические рекомендации преподавателю**

На лекциях используется «проблемный» подход к изложению материала: материал каждой лекции иллюстрируется примерами, рассматриваются нестандартные ситуации, требующие решения с использованием рассматриваемого материала. При этом студенты должны активно участвовать в обсуждении вопросов, выработке решений, предлагаемые студентами решения, обсуждаются, анализируются и оцениваются в ходе лекции. Предлагается рассматривать не только «верные», оптимальные решения, но и решения, приводящие к ошибкам. По каждому рассматриваемому на лекции вопросу следует предложить задачи для самостоятельного решения и вопросы для самостоятельного изучения.

На практических занятиях используются следующие методы обучения и контроля усвоения материала:

- 1) выполнение практических работ по теме занятия сопровождается контрольным опросом;
- 2) обсуждение различных вариантов решения, предложенных студентами, сравнение решений, анализ возможных ситуаций.

Рекомендуется использовать «защиту» выполненного домашнего задания.

### **8.2. Методические указания студентам**

Студенту рекомендуется следующая схема *подготовки к практическому занятию*:

- 1) проработать конспект лекций;
- 2) проанализировать основную и дополнительную литературу, рекомендованную по изучаемому разделу;
- 3) при необходимости найти дополнительную информацию в сети Интернет, на сайтах электронных библиотек;
- 4) проанализировать варианты решений, предложенные преподавателем, найденные в дополнительных источниках;
- 5) при затруднениях сформулировать вопросы к преподавателю.

Студенту рекомендуется следующая схема *подготовки к лекции*:

- 1) проработать конспект лекций;
- 2) изучить материал, предложенный для самостоятельного изучения;
- 3) выполнить предложенные преподавателем задания;
- 4) при затруднениях задать вопросы к преподавателю при проведении индивидуальных консультаций.

Рекомендуется при выполнении домашнего задания рассмотреть возможность защиты предложенных решений, подготовить документацию и «презентацию» работы.

Для самостоятельного изучения и подготовки к лекциям предлагается использовать электронные ресурсы, размещаемые на сервере НИУ ВШЭ – Пермь.

## **9. Оценочные средства для текущего контроля и аттестации студента**

### **9.1. Тематика заданий текущего контроля**

Задание выполняется в группе (не более трех студентов) или индивидуально. Каждый студент отчитывается по каждому пункту задания индивидуально.



Требуется разработать распределенное приложение, позволяющее работать в сети.  
Требования к выполнению работы:

- Приложение должно обеспечивать параллельную работу нескольких клиентов и серверов. Дополнительное требование: возможность запуска нескольких серверов на одном компьютере.
- Клиентские приложения должны автоматически находить серверы для обслуживания и выполнения заданных функций.
- Серверы системы могут выполнять различные функции.
- При разрыве сеанса приложения должны автоматически восстанавливать свою работоспособность.
- Для хранения данных и доступа к ним применить ADO и/или ADO.NET.
- Приложения должны поддерживать возможность взаимодействия в различных режимах.
- Для организации взаимодействия нужно использовать различные средства коммуникации (именованные каналы, мейлслоты, сокет, MSMQ, .Net Remoting, web-сервисы, WCF-сервисы), сравнив их возможности.

По окончании выполнения задания каждая группа студентов должна подготовить отчет и доклад на 7-10 минут. Выступление студентов сопровождается показом презентации, отражающей основные этапы разработки.

Отчет по выполнению задания должен включать:

1. Общее описание приложения. Постановка задачи, введение в предметную область.
2. Архитектура системы. Обоснование выбора данного типа архитектуры распределенного приложения. Алгоритм работы приложения в целом.
3. Архитектура каждого из логических компонент системы (серверы, клиенты, диспетчеры). Подходы к реализации. Алгоритмы работы. Многопоточность, обоснование.
4. Методы коммуникаций компонент системы (клиент→сервер, сервер→клиент и т.д.). Обоснование выбора этих методов коммуникации.
5. Способ передачи данных (синхронная / асинхронная, однонаправленная / двунаправленная и т.д.). Обоснование.
6. Структура передаваемых данных. Вид протоколов, обоснование выбора.
7. Отказоустойчивость системы. Как система поведет себя, если «исчезнет» один или несколько ее компонент. Что произойдет с системой, если «исчезнувший» компонент будет восстановлен на другом узле сети.
8. Работа с базой данных (если используется). Обоснование.
9. Исходный код приложений с комментариями.

## 9.2. Вопросы для оценки качества освоения дисциплины

1. Дайте определение распределенной системы.
2. Каковы предпосылки для перехода к распределенной обработке данных?
3. Дайте определение распределенной системы.
4. Перечислите компоненты, составляющие канонические распределенные системы.
5. Какова роль промежуточного программного обеспечения?
6. Какие ВС можно рассматривать как распределенные?
7. Какие классы ВС (SISD, MISD, SIMD, MIMD) классификации Флинна нельзя отнести к распределенным системам?
8. Приведите примеры распределенных систем.
9. Перечислите задачи, которые стоят перед распределенными системами.
10. Дайте определение прозрачности распределенной системы.



11. Что такое прозрачность доступа? Приведите примеры.
12. Что такое прозрачность переноса и смены местоположения? Приведите примеры.
13. Объясните, что такое прозрачность репликации. Как прозрачность репликации связана с прозрачностью местоположения?
14. Что такое прозрачность параллельного доступа и как можно достичь этой прозрачности?
15. Объясните, что такое прозрачность отказов, и какие трудности возникают при реализации прозрачности отказов?
16. Что такое степень прозрачности? Следует ли всегда придерживаться максимальной степени прозрачности? Приведите примеры.
17. Что такое открытая распределенная система? Каковы преимущества открытой распределенной системы?
18. Дайте определение интероперабельности.
19. Что вы понимаете под переносимостью системы?
20. Расскажите, что вы понимаете под гибкостью системы?
21. Что такое масштабируемая система?
22. Перечислите показатели масштабируемой системы?
23. Подробнее расскажите о географической масштабируемости и о проблемах географической масштабируемости.
24. Приведите примеры масштабируемости по размеру и расскажите о проблемах, которые возникают при её реализации.
25. Что такое административная прозрачность. Каковы пути реализации административной прозрачности.
26. Расскажите о методиках, с помощью которых можно добиться масштабируемости распределенной системы.
27. Дайте определение, что такое протокол?
28. Какова схема передачи сообщения в сети?
29. Перечислите протоколы нижнего уровня.
30. Расскажите о назначении физического уровня протоколов?
31. Что вы можете рассказать о канальном уровне протоколов?
32. Перечислите функции сетевого уровня протоколов и дайте им краткое пояснение.
33. Что такое надежность связи?
34. Приведите перечень причин ошибок связи, объясните, почему происходят ошибки связи?
35. На каком уровне протоколов работает протокол скользящего окна?
36. На каком уровне протоколов работает алгоритм, основанный на таймере.
37. Каков размер пакета, используемый алгоритмом, основанном на таймере?
38. Приведите перечень упрощений для алгоритма, основанного на таймере?

### 9.3. Примеры заданий итогового контроля

1. Разработать централизованный алгоритм балансировки. Решение о переносе объекта с одного вычислительного узла распределенной системы на другой выполняется одним из процессов, который предварительно получает сообщения от всех вычислительных узлов об их загрузке. Сеть имеет древовидную топологию.
2. Разработать централизованный алгоритм балансировки. Решение о переносе объекта с одного вычислительного узла распределенной системы на другой выполняется одним из процессов, который предварительно получает сообщения от всех вычислительных узлов об их загрузке. Сеть имеет произвольную топологию и является ориентированной. Схема сети прилагается.



3. Разработать централизованный алгоритм балансировки. Решение о переносе объекта с одного вычислительного узла распределенной системы на другой выполняется одним из процессов, который предварительно получает сообщения от всех вычислительных узлов об их загрузке. Сеть имеет произвольную топологию и является неориентированной. Схема сети прилагается.
4. Разработать децентрализованный алгоритм балансировки. В этом случае процессы, выполняющиеся на вычислительном узле, при обнаружении чрезмерной загрузки переносят объекты со своего процессора на соседние.

## **10. Учебно-методическое и информационное обеспечение дисциплины**

### **10.1. Базовый учебник**

1. Миков А.И., Замятина Е.Б. Распределенные системы и алгоритмы. – М.: Интернет-Университет Информационных Технологий, 2008. – 377 с.

### **10.2. Основная литература**

2. Афанасьев К.Е., Завозкин С.Ю., Трофимов С.Н., Власенко А.Ю. Основы высокопроизводительных вычислений. – Кемерово: Кемеровский государственный университет, 2011. – 246 с.

### **10.3. Дополнительная литература**

3. Тель Ж. Введение в распределенные алгоритмы. М.: МЦНМО, 2009. – 616 с.
4. Таненбаум Э. Распределенные системы: принципы и парадигмы. – СПб: Питер, 2003. – 877 с.

### **10.4. Справочники, словари, энциклопедии**

Не предусмотрены.

### **10.5. Программные средства**

Для успешного освоения дисциплины, студент использует следующие программные средства:

- Microsoft Visual Studio 2010/2013.
- Microsoft Azure.
- Microsoft Office.

### **10.6. Дистанционная поддержка дисциплины**

Дистанционная поддержка курса предусмотрена в рамках информационной образовательной среды LMS.

## **11. Материально-техническое обеспечение дисциплины**

Для проведения лекционных занятий используется компьютер с установленным программным обеспечением для демонстрации презентаций и проектор.

Практические занятия проводятся в компьютерных классах с установленным программным обеспечением, перечисленным выше.