

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ  
ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение высшего  
образования

«ДАГЕСТАНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

Факультет математики и компьютерных наук

**РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ**

Распределенные объектные технологии

Кафедра дискретной математики и информатики

факультета математики и компьютерных наук

Образовательная программа магистратуры

**02.04.02 Фундаментальная информатика и информационные технологии**

Направленность (профиль) программы  
Информационные технологии

Форма обучения

**очная**

Статус дисциплины:

входит в обязательную часть

Махачкала, 2022


Рабочая программа дисциплины «Распределенные объектные технологии» составлена в 2022 году в соответствии с требованиями ФГОС ВО - магистратура по направлению подготовки 02.04.02 Фундаментальная информатика и информационные технологии от «23» августа 2017 г. № 811.

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

Рабочая программа дисциплины одобрена:  
на заседании кафедры дискретной математики и информатики от «28» февраля 2022 г.,  
протокол № 6;

зав. кафедрой  Магомедов А. М.  
(подпись)

и  
на заседании Методической комиссии факультета математики и компьютерных наук  
от

«24» марта 2022 г., протокол № 4;  
председатель  Ризаев М. К.  
(подпись)

Рабочая программа дисциплины согласована с учебно-методическим управлением  
«31» марта 2022 г.

Начальник УМУ  Гасангаджиева А. Г.  
(подпись)

## Аннотация рабочей программы дисциплины

Дисциплина “Распределенные объектные технологии” входит в обязательную часть ОПОП магистратуры по направлению 02.04.02 Фундаментальная информатика и информационные технологии.

Дисциплина реализуется на факультете математики и компьютерных наук кафедрой дискретной математики и информатики.

Содержание дисциплины охватывает круг вопросов, связанных с основами организации распределенной объектной обработки данных и современными подходами к построению распределенных объектных систем.

Дисциплина нацелена на формирование следующих компетенций выпускника: общепрофессиональной – ОПК-2, профессиональных – ПК-3, ПК-4.

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

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

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

### Очно-заочная форма обучения

Семестр	Учебные занятия						СРС, в том числе экзамен	Форма промежуточной аттестации (зачет, дифференциро ванный зачет, экзамен
	в том числе							
	Контактная работа обучающихся с преподавателем							
	Всего	из них						
Лекц ии		Лаборат орные занятия	Практич еские занятия	КСР	консульта ции			
3	144	14	24				70+36	экзамен

## 1. Цели освоения дисциплины

Целями освоения дисциплины «Распределенные объектные технологии» являются:

- получение фундаментальных знаний по основам организации объектной распределенной обработки данных;
- формирование представлений о сервис-ориентированном подходе к организации распределенных систем.

Задачи курса:

- изучить сервис - ориентированный подход к построению распределенных вычислительных систем;
- ознакомить с основами разработки распределенных вычислительных систем на основе веб-сервисов;
- изучить основные стандарты разработки веб-сервисов.

## 2. Место дисциплины в структуре ОПОП магистратуры

Дисциплина «Распределенные объектные технологии» входит в обязательную часть ОПОП магистратуры по направлению 02.04.02 Фундаментальная информатика и информационные технологии и преподается на 2 курсе в 4 семестре. Изучение предмета завершается письменным экзаменом в конце семестра.

Дисциплина «Распределенные объектные технологии» логически и содержательно взаимосвязана с дисциплинами, в которых рассматриваются вопросы проектирования сложных распределенных программных систем: Объектно-ориентированные Case-технологии, Языки и методы программирования, «Сетевые технологии».

Для освоения данной дисциплины необходимо знание основ объектно-ориентированного моделирования и одного из объектно-ориентированных языков программирования.

Результаты освоения данной дисциплины будут востребованы на занятиях по «Анализу информационных систем», «Технологиям сети Интернет» и в проектно-производственной деятельности магистранта.

## 3. Компетенции обучающегося, формируемые в результате освоения дисциплины (перечень планируемых результатов обучения).

Код компетенции из ФГОС ВО	Наименование компетенции из ФГОС ВО	Планируемые результаты обучения	Процедура освоения
----------------------------	-------------------------------------	---------------------------------	--------------------

<p><b>ОПК-2</b> Способен применять компьютерные/ суперкомпьютерные методы, современное программное обеспечение (в том числе отечественного происхождения) для решения задач профессиональной деятельности</p>	<p>ОПК-2.1. Знает основные положения и концепции в области программирования, архитектуру языков программирования, теории коммуникации, знает основную терминологию, знаком с перечнем ПО, включенного в Единый Реестр Российских программ.</p>	<p><b>Знает:</b> принципы самостоятельного поиска необходимой литературы для подготовки реферата, проекта и сдачи экзамена. <b>Умеет:</b> самостоятельно выполнять домашние задания, готовить реферат; анализировать и оценивать основные распределенные технологии. <b>Владеет:</b> навыками анализа основных распределенных систем.</p>	<p>Конспектирование лекций, участие в дискуссиях. Разработка и реализация проектов на лабораторных занятиях.</p>
	<p>ОПК-2.2. Умеет анализировать типовые языки программирования, составлять программы.</p>	<p><b>Знает:</b> Основные парадигмы программирования <b>Умеет:</b> составлять программы в структурном и объектно-ориентированном стилях <b>Владеет:</b> навыками анализа основных распределенных систем.</p>	

	<p>ОПК-2.3. Имеет практический опыт решения задач анализа) интеграции различных типов программного обеспечения, анализа типов коммуникации.</p>	<p><b>Знает:</b> Архитектуру современных распределенных систем.</p> <p><b>Умеет:</b> составлять программы с использованием API известных распределенных приложений.</p> <p><b>Владеет:</b> навыками создания веб-приложений.</p>	
<p><b>ПК-3.</b> Способность проектировать распределенные информационные системы, их компоненты и протоколы их взаимодействия, собирать, обрабатывать и интерпретировать экспериментальные данные, необходимые для проектной и производственно-технологической деятельности; способность к разработке новых алгоритмических, методических и технологических</p>	<p><b>ПК-3.1.</b> Знает основы проектирования и элементы архитектурных решений информационных систем.</p>	<p><b>Знает:</b> международные и российские стандарты жизненного цикла ПО.</p> <p><b>Умеет:</b> разрабатывать техническое задание для проекта по заданному шаблону.</p> <p><b>Владеет:</b> навыками объектно-ориентированного проектирования задач в различных предметных областях.</p>	<p>Конспектирование лекций, подготовка и защита реферата, участие в дискуссиях. Разработка и реализация проектов на лабораторных занятиях.</p>

<p>решений в конкретной сфере профессиональной деятельности.</p>	<p><b>ПК-3.2.</b>          Умеет применять в практической деятельности профессиональные стандарты в области информационных технологий.</p>	<p><b>Знает:</b> стандарт моделирования UML 2.5.x и стандарт проектирования программных систем – RUP.  <b>Умеет:</b> разрабатывать техническое задание для проекта с использованием диаграмм UML.  <b>Владеет:</b> навыками объектно-ориентированного моделирования на языке UML.</p>
	<p><b>ПК-3.3.</b>          Имеет практический опыт составления технического задания на разработку информационной системы.</p>	<p><b>Знает:</b> стандарт проектирования программных систем RUP.  <b>Умеет:</b> разрабатывать техническое задание SRS  <b>Владеет:</b> навыками объектно-ориентированного моделирования на языке UML.</p>

<p><b>ПК-4</b> Способность применять в профессиональной деятельности современные языки программирования и методы параллельной обработки данных, операционные системы, электронные библиотеки и пакеты программ, сетевые технологии</p>	<p>ПК-4.1. Знает современные языки программирования и методы параллельной обработки данных.</p>	<p><b>Знает:</b> основы проектирования информационных систем на основе международного стандарта UML.</p> <p><b>Умеет:</b> отображать графическую нотацию языка UML в программы на языках программирования Delphi, C#, Java, C++ (по выбору студента), а также представлять программы на языках программирования Delphi, C#, Java, C++ (по выбору студента) с помощью графической нотации UML.</p> <p><b>Владеет:</b> навыками использования языка UML с помощью CASE-инструментов при проектировании программного обеспечения.</p>	<p>Конспектирование лекций, подготовка и защита реферата, участие в дискуссиях. Разработка и реализация проектов на лабораторных занятиях.</p>
--	---	--	--



	<p>ПК-4.2. Умеет реализовывать численные методы решения прикладных задач в профессиональной сфере деятельности, пакеты программного обеспечения, операционные системы, электронные библиотеки, сетевые технологии.</p>	<p><b>Знает:</b> основы проектирования информационных систем на основе международного стандарта RUP.</p> <p><b>Умеет:</b> применять известные распределенные технологии для проектирования в различных предметных областях.</p> <p><b>Владеет:</b> навыками использования языка UML и CASE-инструментов при проектировании сложного программного обеспечения.</p>	
	<p>ПК-4.3. Имеет практический опыт разработки интеграции информационных систем.</p>	<p><b>Знает:</b> основы создания RMI-приложений, сервисов REST и WCF и других клиент-серверных приложений.</p> <p><b>Умеет:</b> оптимизировать и отлаживать программный код распределенной системы.</p> <p><b>Владеет:</b> навыками интеграции различных технологий для создания работающего</p>	

		приложения.	

**4. Объем, структура и содержание дисциплины.**

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

4.2 Структура дисциплины

4.2.1 Структура дисциплины в очно-заочной форме

<b>№ п/п</b>	<b>Разделы и темы дисциплины</b>	<b>Семестр</b>	<b>Неделя</b>	<b>Виды учебной работы, включая самостоятельную работу студентов и трудоемкость (в часах)</b>	<b>Формы контроля успеваемости и (по</b>
------------------	--------------------------------------	----------------	---------------	---	--

				Лекции	Практические занятия	Лабораторные занятия	Самостоятельная работа	Контроль сам. работы	неделям семестра) Форма промежуточной аттестации (по семестрам)
<b>Модуль 1. Основы организации объектной распределенной обработки данных</b>									
1	Введение в распределенные вычислительные системы	4	1-2	2			2		Прием лабораторных работ (ЛР) и реферата (Р)
2	Клиент-серверная архитектура	4	3-4	2		4	8		ЛР, Р
3	Методы переноса кода и агентные технологии	4	5-6	2		4	4		ЛР, Р
4	Технология CORBA	4	7-8	2		4	4		ЛР, Р
<b>Итого по модулю 1:</b>			36	8		12	16		<b>Модуль 1</b>
<b>Модуль 2. Разработка распределенных вычислительных систем</b>									
5	Технология RMI	4		2		2	14		ЛР, Р
6	Основы компонентных программных систем	4	9-10	2		2	14		ЛР, Р
<b>Итого по модулю 2:</b>			36	4		4	28		<b>Модуль 2</b>
<b>Модуль 3. Разработка распределенных систем на основе сервисов</b>									
7	Сервис-ориентированная архитектура	4	11-12	2		2	8		ЛР, Р
7	Принципы архитектуры REST	4	13-14			2	8		ЛР, Р
8	Основы технологии WCF	4	15-16			4	2		ЛР, Р
<b>Итого по модулю 3:</b>			36	2		8	26		<b>Модуль 3</b>
Подготовка к экзамену							36		<b>Экзамен</b>
<b>ИТОГО:</b>			144	14		24	70		36

## **4.3 Содержание дисциплины, структурированное по темам (разделам)**

### **4.3.1 Содержание лекционных занятий по дисциплине**

#### **Модуль 1. Основы организации объектной распределенной обработки данных**

##### **Тема 1. Введение в распределенные вычислительные системы.**

Определение распределенной вычислительной системы (РВС). Организация связи между компонентами. Удаленный вызов процедур. Организация связи с использованием удаленных объектов.

##### **Тема 2. Клиент-серверная архитектура.**

Разделение приложений по уровням. Типы клиент-серверной архитектуры. Классификация распределенных вычислительных систем. Модель взаимодействия «Клиент-сервер».

##### **Тема 3. Методы переноса кода и агентные технологии.**

Модели переноса кода. Понятие программного агента. Безопасность в системах мобильных агентов.

##### **Тема 4. Технология CORBA.**

Основные понятия CORBA: OMG, OMA, ORB, CORBA. Брокер объектных запросов.

Язык определения интерфейса IDL. Достоинства и недостатки подхода CORBA.

#### **Модуль 2. Разработка распределенных вычислительных систем**

##### **Тема 5. Основы компонентных программных систем.**

Определение программного компонента. Технологии OLE, COM, ActiveX. Технология Enterprise JavaBeans.

##### **Тема 6. Технология RMI.**

Клиент и сервер RMI. Удаленные интерфейсы, объекты и методы. Организация доступа к классам в сети.

#### **Модуль 3. Разработка распределенных систем на основе сервисов**

##### **Тема 7. Сервис-ориентированная архитектура.**

История веб-сервисов и сервис-ориентированного подхода. Определение и составляющие сервис-ориентированной архитектуры. Слабо- и сильносвязанные программные системы.

Основные принципы сервис-ориентированного подхода.

##### **Тема 8. Принципы архитектуры REST.**

Шесть ограничений REST. Методы протокола HTTP. Форматы HTML, XML и JSON.

Примеры.

## **Тема 9. Основы технологии WCF**

Основные стандарты веб-сервисов: WSDL, SOAP, WS-Security, WS-Addressing, WSRF.

Основные принципы WCF.

### **4.3.2 Содержание лабораторно-практических занятий по дисциплине**

Лабораторные занятия предусмотрены по всем темам модулей и их содержание совпадает с содержанием тем модулей. Целью каждой лабораторной работы является построение несложного распределенного приложения с использованием соответствующей технологии.

#### **Модуль 1. Основы организации объектной распределенной обработки данных**

**Лабораторная работа 1. Удаленный вызов процедур.**

Цель работы: организовать связь с использованием удаленных объектов.

**Лабораторная работа 2. Клиент-серверная архитектура.**

Цель работы: спроектировать систему на основе модели взаимодействия «Клиент-сервер».

**Лабораторная работа 3. Методы переноса кода и агентные технологии.**

Цель работы: спроектировать систему на основе модели переноса кода.

**Лабораторная работа 4. Технология CORBA.**

Цель работы: спроектировать систему на основе модели взаимодействия CORBA

#### **Модуль 2. Разработка распределенных вычислительных систем**

**Лабораторная работа 5. Основы компонентных программных систем.**

Цель работы: определить программный компонент на основе одной из технологий: OLE, COM, ActiveX, технологии Enterprise JavaBeans.

**Лабораторная работа 6. Технология RMI.**

Цель работы: определить компоненты RMI и организовать доступ к классам в сети.

#### **Модуль 3. Разработка распределенных систем на основе сервисов**

**Лабораторная работа 7. Сервис-ориентированная архитектура.**

Цель работы: определить составляющие сервис-ориентированной архитектуры.

**Лабораторная работа 8. Принципы архитектуры REST.**

Цель работы: создать приложение на основе методов протокола HTTP. Форматы HTML, XML и JSON.

**Лабораторная работа 9. Основы технологии WCF**

Цель работы: создать WCF-службу.

## 5. Образовательные технологии

Процесс изложения учебного материала сопровождается презентациями и демонстрацией решения задач в интерактивном режиме с использованием мультимедийного проектора. Предусмотрено регулярное общение с лектором и представителями российских и зарубежных компаний по электронной почте и по скайпу.

## 6. Учебно-методическое обеспечение самостоятельной работы студентов.

Пакет заданий для самостоятельной работы выдается в начале семестра, определяются предельные сроки их выполнения и сдачи. Самостоятельная работа студентов складывается из проработки лекционного материала (настоятельно рекомендуется самостоятельное практическое решение всех разобранных на лекциях упражнений), материала учебника, видео-лекций и соответствующих форумов интернет, решения всех заданий из индивидуальных заданий, решения рекомендуемых задач, подготовки к сдаче промежуточных отчетов и экзамена.

№	Вид самостоятельной работы	Вид контроля	Учебно-методическое обеспечение
1.	Проработка лекционного материала.	Контрольный фронтальный опрос.	См. разделы 7.3, 8, 9 данного документа
2.	Изучение рекомендованной литературы и материалов соответствующих форумов интернет.	Контрольный фронтальный опрос, прием и представление рефератов.	См. разделы 7.3, 8, 9 данного документа
3.	Подготовка к отчетам по лабораторным работам.	Проверка выполнения работ, опрос по теме работы.	См. разделы 7.3, 8, 9 данного документа
4.	Подготовка к сдаче промежуточных форм контроля.	Контрольные работы по каждому модулю и прием рефератов.	См. разделы 7.3, 8, 9 данного документа

Пакет заданий для самостоятельной работы выдается в начале семестра, определяются предельные сроки их выполнения и сдачи.

## 7. Фонд оценочных средств для проведения текущего контроля успеваемости, промежуточной аттестации по итогам освоения дисциплины.

### 7.1 Типовые контрольные задания

#### 7.1.1 Вопросы для самостоятельной работы

1. Определение распределенной вычислительной системы. Классификация распределенных вычислительных систем.
2. Модель взаимодействия «Клиент-сервер». Уровни клиент-серверной архитектуры.
3. Технология CORBA и распределенные объектные технологии. Основные понятия CORBA: OMG, OMA, ORB, CORBA.
4. Достоинства и недостатки подхода CORBA.
5. Определение программного компонента. Технологии OLE, COM, ActiveX.
6. Определение и составляющие сервис – ориентированной архитектуры.
7. Слабо- и сильносвязанные программные системы.
8. Основные принципы сервис – ориентированного подхода.
9. Основные стандарты веб-сервисов: WSDL, SOAP.
10. Второе поколение стандартов веб-сервисов: WS-Security, WS-Addressing, WSRF.
11. Основные принципы WCF.
12. Основные принципы REST.
13. Основы платформы BOINC.
14. Основы платформы Google Cloud Platform.
15. Основы платформы Windows Azure.
16. Основы платформы Amazon Compute Cloud.

### **7.1.2 Контрольные вопросы к экзамену**

1. Определение распределенной вычислительной системы (PBC).
2. Организация связи между компонентами.
3. Удаленный вызов процедур.
4. Организация связи с использованием удаленных объектов.
5. Модель взаимодействия «Клиент-сервер». Уровни клиент-серверной архитектуры.
6. Модели переноса кода. Понятие программного агента.
7. Безопасность в системах мобильных агентов.
8. Технология CORBA и распределенные объектные технологии.
9. Основные понятия CORBA: OMG, OMA, ORB, CORBA.
10. Язык определения интерфейса IDL
11. Определение программного компонента. Технологии OLE, COM, ActiveX.
12. Технология Enterprise JavaBeans.
13. Клиент и сервер RMI. Удаленные интерфейсы, объекты и методы.
14. Организация доступа к классам в сети.

15. Определение и составляющие сервис – ориентированной архитектуры.
16. Слабо- и сильносвязанные программные системы.
17. Основные принципы сервис – ориентированного подхода.
18. Основные стандарты веб-сервисов: WSDL, SOAP, WS-Security, WS-Addressing, WSRF.
19. Основные принципы WCF.
20. Основные принципы REST.
21. Методы протокола HTTP.
22. Форматы HTML, XML и JSON. Примеры.

### **7.1.3 Примерный контрольный тест по первому модулю**

1. Изобретателем URI, URL, HTTP, HTML является:
  1. Джон Бардин
  2. Сеймур Крей
  3. Сэр Тимоти Джон Бернерс-Ли
  4. Джон фон-Нейман
2. Выберите неверное утверждение:
  1. На базе CORBA возможно было реализовать концептуальные модели архитектуры распределенной вычислительной среды;
  2. CORBA не обеспечивала взаимосвязь с языком UML;
  3. Популярность CORBA пошла на спад из-за сложности процесса разработки на основе объектно-ориентированной модели и высокой стоимости приобретения и поддержки;
  4. Для борьбы с гетерогенностью вычислительной среды CORBA реализует высокоуровневые стандарты взаимодействия,
  5. JVM (Java Virtual Machine – Виртуальная Машина Java), в совокупности с Java приложениями и сервисами, преодолевают проблемы гетерогенности вычислительных систем, предоставляя методы удаленного вызова процедур посредством технологии RMI (Remote Method Invocation – Удаленный Вызов Методов).
3. SOAP – это:
  1. основанный на XML протокол взаимодействия веб-сервисов;
  2. язык описания веб-сервисов;
  3. метод описания, поиска, взаимодействия и использования веб-сервисов;
  4. парадигма крупномасштабных распределенных вычислений, основанная на эффекте масштаба.
4. WSDL – это:
  1. основанный на XML протокол взаимодействия веб-сервисов;
  2. язык описания веб-сервисов;



3. метод описания, поиска, взаимодействия и использования веб-сервисов;
4. парадигма крупномасштабных распределенных вычислений, основанная на эффекте масштаба.
5. UDDI – это:
  1. основанный на XML протокол взаимодействия веб-сервисов;
  2. язык описания веб-сервисов;
  3. метод описания, поиска, взаимодействия и использования веб-сервисов;
  4. парадигма крупномасштабных распределенных вычислений, основанная на эффекте масштаба.
6. Облако –это:
  1. основанный на XML протокол взаимодействия веб-сервисов;
  2. язык описания веб-сервисов;
  3. метод описания, поиска, взаимодействия и использования веб-сервисов;
  4. парадигма крупномасштабных распределенных вычислений, основанная на эффекте масштаба.
7. Выберите верное утверждение:
  1. Процессы, реализующие некоторый сервис, например, сервис файловой системы или базы данных, называются клиентами.
  2. Процессы, запрашивающие сервисы у серверов путем отправки запроса и последующего ожидания ответа от сервера, называются серверами.
  3. Уровень пользовательского интерфейса обычно реализуется на серверах.
  4. На «толстых» клиентах возлагаются основные задачи по отображению информации пользователю и обработке всех данных.
8. Выберите неверное утверждение:
  1. Изменения в компонентную программную систему вносятся путем создания новых компонентов или изменения старых, а не путем рефакторинга существующего кода.
  2. Программный компонент – это автономный элемент программного обеспечения, предназначенный для многократного использования, который может распространяться для использования в других программах в виде скомпилированного кода.
  3. В программе, использующей компонент, нельзя назначать обработчики событий, на которые умеет реагировать компонент.
  4. Все методы, предоставляемые в интерфейсе программного компонента, должны быть качественно документированы, с учетом всех возможных вариантов их использования в сторонних приложениях.
9. Выберите неверное утверждение:
  1. Веб-сервисы – это слабосвязанные программные компоненты, поддерживающие многократное использование, которые семантически инкапсулируют отдельные функциональные возможности и программным образом доступны по стандартным протоколам Интернета.
  2. Веб-сервис – это среда программирования.
  3. Сильная связанность возникает, когда зависимый класс содержит ссылку непосредственно на определенный класс, предоставляющий некоторые возможности.

4. Сильная связанность возникает, когда зависимый класс содержит ссылку на интерфейс, который может быть реализован одним или несколькими конкретными классами.
5. Основная цель использования концепции слабосвязанных программных систем – это уменьшение количества зависимостей между компонентами.

10. Выберите неверное утверждение:

1. Веб-сервисы – это слабосвязанные программные компоненты, поддерживающие многократное использование, которые инкапсулируют отдельные функциональные возможности и программным образом доступны по стандартным протоколам Интернета.
2. Сильная связанность возникает, когда зависимый класс содержит ссылку непосредственно на определенный класс, предоставляющий некоторые возможности.
3. Сильная связанность возникает, когда зависимый класс содержит ссылку на интерфейс, который может быть реализован одним или несколькими конкретными классами.
4. Основная цель использования концепции слабосвязанных программных систем – это уменьшение количества зависимостей между компонентами.

11. Выберите неверное утверждение:

1. Сервисы должны поддерживать повторное использование.
2. Контракт сервиса предоставляет конечную точку (service endpoint): адрес, по которому можно обратиться к данному сервису;
3. Сервисы должны поддерживать обнаружение.
4. Контракт сервиса предоставляет основные операции, предоставляемые сервисом;
5. Сервисы должны быть слабосвязаны.
6. Сервисы должны быть спроектированы таким образом, чтобы поддерживать возможность их использования в качестве элементов другого сервиса.

12. Выберите неверное утверждение:

1. Клиент BOINC – это набор PHP-сценариев для организации и управления проектом: регистрация участников, распределение заданий, получение результатов;
2. Клиент BOINC – это пользовательское приложение, позволяющее участвовать в одном или нескольких проектах.
3. WSDL написан на XML и является XML-документом.
4. Стандарт WSDL обеспечивает описание веб-сервиса в виде сообщений, которые может отправить или же принять веб-сервис.
5. Проект SETI@home (распределенный поиск внеземных цивилизаций) так же, как и Skype, основан на технологии одноранговых сетей P2P.
6. Основным отличием AWS (Amazon Web Services) от всех остальных платформ является то, что каждый разработчик может использовать собственную среду исполнения приложений.

13. Выберите неверное утверждение:

1. Облачные вычисления предоставляют только базу для развертывания вычислительной инфраструктуры.
2. Облачные вычисления предоставляют интегрированный подход на всех уровнях предоставления информационных ресурсов: IaaS, PaaS, SaaS.
3. Общественное облако предоставляет ресурсы облака широкому кругу пользователей по принципу «оплата по мере использования» (pay-as-you-go).
4. Облако – это большой пул легко используемых и легкодоступных виртуализованных ресурсов (таких как аппаратные комплексы, сервисы и др.).
5. Облачные платформы фокусируются на подходе «всё как сервис».
6. Google App Engine может запускать приложения, написанные на Python, Java, Jython, Scala и т.п.

#### **7.1.4 Темы рефератов**

1. Методы распределенной обработки больших данных: концепция MapReduce, платформа Hadoop, ApacheSpark.
2. Облачная платформа (Amazon Elastic Compute Cloud|Microsoft Azure | Google Cloud Platform).
3. Технологии интернет-вычислений. Платформа BOINC.
4. Технологии DevOps: понятие и основные парадигмы.
5. REST API. Технология WCF.
6. Архитектура, принципы разработки, тестирования и/или масштабирования высоконагруженных систем (на примере Netflix, Twitter, Facebook и др.).

#### **7.2. Методические материалы, определяющие процедуру оценивания знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций.**

Общий результат выводится как интегральная оценка, складывающаяся из текущего контроля - 50% и промежуточного контроля - 50%. Текущий контроль по дисциплине включает:

- посещение занятий - 10 баллов,
- подготовка реферата – 20 баллов,
- выполнение лабораторных заданий – 20 баллов,
- выполнение домашних (аудиторных) контрольных работ - 50 баллов. Промежуточный контроль по дисциплине включает:

- прием лабораторных работ - 30 баллов,
- письменная контрольная работа - 50 баллов,
- тестирование - 20 баллов.

## **8. Перечень основной и дополнительной учебной литературы, необходимой для освоения дисциплины**

а) адрес сайта курса

<https://explorecoursesit.blogspot.com/>

б) основная литература:

1. Таненбаум Э., Ван-Стеен М. Распределенные системы. Принципы и парадигмы. [Текст] / Э. Таненбаум. Спб.: Питер, 2003. 877 с.
2. Berners-Lee T. Universal Resource Identifiers – Axioms of Web Architecture.  
URL: <http://www.w3.org/DesignIssues/Axioms.html> (дата обращения: 10.10.2018)
3. Радченко Г.И. Распределенные вычислительные системы. ЮУрГУ, 2012 г. 182 с.:  
[http://glebradchenko.ru/doc/Radchenko\\_Distributed\\_Computer\\_Systems.pdf](http://glebradchenko.ru/doc/Radchenko_Distributed_Computer_Systems.pdf)
4. Распределённые информационные системы [Электронный ресурс] : учебно-методическое пособие по дисциплине Сетевые технологии / . — Электрон. текстовые данные. — М. : Московский технический университет связи и информатики, 2016. — 16 с. — 2227-8397. — Режим доступа: <http://www.iprbookshop.ru/61537.html>
5. Сычев А.В. Web-технологии [Электронный ресурс] / А.В. Сычев. — Электрон. текстовые данные. — М. : Интернет-Университет Информационных Технологий (ИНТУИТ), 2016. — 184 с. — 2227-8397. — Режим доступа: <http://www.iprbookshop.ru/56344.html>

в) дополнительная литература:

6. Robert Daigneau. Service Design Patterns: Fundamental Design Solutions for SOAP/WSDL and RESTful Web Services. AddisonWesley Professional, 2011. 352 p.
7. Thomas Erl. Service-Oriented Architecture (SOA): Concepts, Technology, and Design. Prentice Hall, 2005. 792 p.
8. Sukumar Ghosh. Distributed Systems: An Algorithmic Approach. CRC Press, 2010. 424 p.
9. Мухамедзянов Р.Р. JAVA. Серверные приложения [Электронный ресурс] / Р.Р. Мухамедзянов. — Электрон. текстовые данные. — М. : СОЛОН-ПРЕСС, 2010. — 336 с. — 5-93455-134-5. — Режим доступа: <http://www.iprbookshop.ru/65089.html>

## **9. Перечень ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины.**

Видеокурсы лекций:

- 1) <https://www.coursera.org/>
- 2) <https://www.udacity.com/>
- 3) <https://www.intuit.ru/>

Форумы и блоги по компьютерным наукам и программированию:

- 1) [www.stackoverflow.com](http://www.stackoverflow.com)
- 2) <http://www.cyberforum.ru/>
- 3) <https://explorecoursesit.blogspot.com/>

#### **10. Методические указания для обучающихся по освоению дисциплины.**

Все упражнения, приведенные на лекции с решениями, следует прорабатывать сразу после лекции. Самостоятельная работа студентов складывается из:

- проработки лекционного материала (настоятельно рекомендуется самостоятельное практическое решение всех разобранных на лекциях упражнений);
- изучения рекомендованной литературы и материалов соответствующих форумов интернет;
- подготовки к отчетам по лабораторным работам;
- подготовки к сдаче промежуточных форм контроля (контрольных работ и сдаче реферата).

Пакет лабораторных заданий рассчитан на семестр. Рекомендуется выполнять и сдавать задания своевременно с прохождением соответствующего материала.

Модули и темы для самостоятельного изучения	Виды и содержание самостоятельной работы
Модуль 1. Основы организации объектной распределенной обработки данных. Тема 1. Введение в распределенные вычислительные системы.	Проработка лекционного материала. Изучение рекомендованной литературы и материалов соответствующих форумов интернет для подготовки реферата.
Модуль 1. Основы организации объектной распределенной обработки данных. Тема 2. Клиент-серверная архитектура. Разделение приложений по уровням.	Проработка лекционного материала. Изучение рекомендованной литературы и материалов соответствующих форумов интернет для подготовки реферата.

Модуль 1. Основы организации объектной распределенной обработки данных. Тема 3. Методы переноса кода и агентные технологии.	Проработка лекционного материала. Подготовка к отчетам по лабораторным работам. Подготовка к сдаче промежуточных форм контроля.
Модуль 1. Основы организации объектной распределенной обработки данных. Тема 4. Технология CORBA.	Проработка лекционного материала. Изучение рекомендованной литературы и материалов соответствующих форумов интернет для подготовки реферата. Подготовка к отчетам по лабораторным работам
Модуль 2. Разработка распределенных вычислительных систем. Тема 1. Основы компонентных программных систем.	Проработка лекционного материала. Изучение рекомендованной литературы и материалов соответствующих форумов интернет для подготовки реферата. Подготовка к отчетам по лабораторным работам. Подготовка к сдаче промежуточных форм контроля.
"	Проработка лекционного материала.
Модуль 3. Разработка распределенных вычислительных систем на основе вебсервисов. Тема 2. Сервисориентированная архитектура.	Изучение рекомендованной литературы и материалов соответствующих форумов интернет для подготовки реферата. Подготовка к отчетам по лабораторным работам.
Модуль 3. Разработка распределенных вычислительных систем на основе веб-сервисов. Тема 3. Принципы архитектуры REST.	Проработка лекционного материала. Изучение рекомендованной литературы и материалов соответствующих форумов интернет для подготовки отчетов по лабораторным работам.
Модуль 3. Разработка распределенных вычислительных систем на основе веб-сервисов. Тема 4. Основы технологии WCF.	Проработка лекционного материала. Изучение рекомендованной литературы и материалов соответствующих форумов интернет для подготовки отчетов по лабораторным работам.

**11. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине, включая перечень программного обеспечения и информационных справочных систем.**

При осуществлении образовательного процесса студентами и профессорско-преподавательским составом используются следующее программное обеспечение: Microsoft Windows, Ubuntu Linux, Microsoft Visual Studio Ultimate, Rational Rose, Skype.

Также студентам предоставляется доступ к российским и международным электронным библиотекам через компьютеры университета.

**12. Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине.**

Имеется необходимая литература в библиотеке, медиапроектор и компьютер для проведения лекций-презентаций.

Лабораторные занятия проводятся в компьютерных классах с необходимым программным обеспечением.

Вся основная литература предоставляется студенту в электронном формате.