

Минобрнауки России  
**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ  
ВЫСШЕГО ОБРАЗОВАНИЯ  
«ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»  
(ФГБОУ ВО «ВГУ»)**

**УТВЕРЖДАЮ**

Заведующий кафедрой  
Борисов Дмитрий Николаевич  
Кафедра информационных систем



25.02.2022

**РАБОЧАЯ ПРОГРАММА УЧЕБНОЙ ДИСЦИПЛИНЫ**

Б1.В.06 Параллельное и распределенное программирование

**1. Код и наименование направления подготовки/специальности:**

09.04.02 Информационные системы и технологии

**2. Профиль подготовки/специализация:**

Анализ и синтез информационных систем

**3. Квалификация (степень) выпускника:**

Магистратура

**4. Форма обучения:**

Очная

**5. Кафедра, отвечающая за реализацию дисциплины:**

Кафедра информационных систем

**6. Составители программы:**

Фертиков Вадим Валериевич, кандидат физ.-мат. наук, доцент

**7. Рекомендована:**

Научно-методическим советом ФКН, протокол НМС ФКН № 3 от 25.02.2022

**8. Учебный год:**

2023-2024

**9. Цели и задачи учебной дисциплины:**

формирование профессиональных компетенций будущих магистров информационных систем и технологий через изучение наиболее общих принципов организации распределенных приложений, вычислительных и операционных систем, технологий их разработки и реализации, практических приемов их применения для реализации распределенных приложений с использованием наиболее известных технологий и моделей программирования. В процессе освоения учебных материалов студент получит знание основных принципов организации распределенных приложений, вычислительных и операционных систем; представление о наиболее употребительных моделях распределенного программирования, в том числе, процедурной, компонентной и Grid, о принципах функционирования соответствующей инфраструктуры их поддержки в гетерогенной среде, о современных стандартах на последовательности протоколов разных уровней и архитектуру систем

программирования для реализации распределенных вычислений; умение применять перечисленные сведения для проектирования систем распределенных вычислений; навыки реализации распределенных приложений с использованием широко распространенных технологий и моделей программирования.

#### **10. Место учебной дисциплины в структуре ООП:**

обязательная дисциплина вариативной части магистерской программы (Б1.В); входные знания в объеме базовых курсов по программам обучения ступени бакалавриата в области информационных технологий, управления данными, языков и систем программирования, методов и средств проектирования информационных систем и технологий, а также обязательных курсов, предусмотренных учебным планом для изучения в семестрах 1 – 2: «Методы исследования и моделирования информационных процессов и технологий», «Системная инженерия».

№ п/п	Наименование дисциплин учебного плана, с которым организована взаимосвязь дисциплины рабочей программы	№ разделов дисциплины рабочей программы, связанных с указанными дисциплинами
1	Системы и сети передачи информации	2, 3, 8, 17, 19
2	Системная инженерия	2, 10, 19-21

**11. Планируемые результаты обучения по дисциплине/модулю (знания, умения, навыки), соотнесенные с планируемыми результатами освоения образовательной программы (компетенциями выпускников) и индикаторами их достижения:**

Код и название компетенции	Код и название индикатора компетенции	Знания, умения, навыки
<p>ПК-1 Способен организовывать работу программистов в группе по созданию системного ПО</p> <p>ПК-2 Способен организационно и технологически обеспечивать определение первоначальных требований</p> <p>ПК-6 Способен управлять выпуском релизов ИС</p> <p>ПК-10 Способен определять варианты структур программного обеспечения информационных систем (программного средства), необходимые информационные потоки и исследовать варианты структур с использованием моделей различного уровня</p> <p>ПК-14 Способен проектировать архитектуру программного средства</p> <p>ПК-14 Способен проектировать архитектуру программного средства</p>	<p>ПК-1.1 Умеет выполнять декомпозицию поставленной задачи и распределение подзадач между программистами</p> <p>ПК-2.2 Умеет назначать и распределять ресурсы по реализации требований к ИС</p> <p>ПК-6.1 Умеет определять состав и разрабатывать план выпуска релизов ИС</p> <p>ПК-10.2 Умеет проводить формирование вариантов структуры системы (программного средства) и и разрабатывает варианты реализации их реализации в рамках предлагаемых алгоритмических и программных решений</p> <p>ПК-14.1 Умеет определять состав компонентов программного средства</p> <p>ПК-14.2 Умеет определять способы взаимодействия между программными подсистемами программного средства</p>	<p>знать: основные принципы организации распределенных приложений, вычислительных и операционных систем; наиболее употребительные модели распределенного программирования, в том числе, процедурную, компонентную и Grid, принципы функционирования соответствующей инфраструктуры их поддержки в гетерогенной среде, стандарты на последовательности протоколов разных уровней и архитектуру систем программирования для реализации распределенных вычислений;</p> <p>уметь: применять перечисленные сведения для проектирования систем распределенных вычислений; осуществлять моделирование проектируемых распределенных приложений и их отдельных компонентов на базе стандартных средств разработки ПО (SDK);</p> <p>владеть (иметь навык(и)): навыками реализации распределенных приложений с использованием широко распространенных технологий и моделей программирования.</p>

## 12. Объем дисциплины в зачетных единицах/час:

3/108

## Форма промежуточной аттестации:

Зачет

## 13. Трудоемкость по видам учебной работы

Вид учебной работы	Семестр 4	Всего
Аудиторные занятия	36	36
Лекционные занятия	12	12
Практические занятия		0
Лабораторные занятия	24	24

Вид учебной работы	Семестр 4	Всего
Самостоятельная работа	72	72
Курсовая работа		0
Промежуточная аттестация	0	0
Часы на контроль		0
Всего	108	108

### 13.1. Содержание дисциплины

п/п	Наименование раздела дисциплины	Содержание раздела дисциплины	Реализация раздела дисциплины с помощью онлайн-курса, ЭУМК
1	Эволюция технологии доступа к данным в информационных системах.	Этапы развития технологии: модель "Host+терминалы"; настольные СУБД; их сетевые версии; архитектура "клиент/сервер"; трехзвенные и многозвенные системы; приложения для WEB-серверов; Grid-технологии.	-
2	Понятие DCE (Distributed Computing Environment).	Понятие DCE - инфраструктуры для разработки и функционирования распределенных приложений в гетерогенных системах: модель программирования, место на уровнях модели OSI-RM (Open System Interconnection Reference Model).	Дистанционное чтение лекций
3	Архитектура средств DCE OSF (Open Software Foundation).	Архитектура средств DCE OSF как интегрированного набора спецификаций стандартов функций, независимого от платформ, операционных систем и сетевых средств: нити, RPC, служба каталогов, служба безопасности, служба времени, распределенная файловая система.	Дистанционное чтение лекций

п/п	Наименование раздела дисциплины	Содержание раздела дисциплины	Реализация раздела дисциплины с помощью онлайн-курса, ЭУМК
4	Концепция вызова удаленных процедур (Remote Procedure Call, RPC) на примере RPC DCE OSF.	Концепция вызова удаленных процедур на примере RPC DCE OSF. Обеспечение прозрачности вызова. Язык определения интерфейсов (Interface Definition Language, IDL) и схема сборки распределенного приложения.	Конспект лекции.
5	Механизм вызова RPC.	Механизм вызова RPC. Динамическое связывание. Особенности протокола RPC DCE OSF. Размещение RPC по уровням модели OSI-RM.	Конспект лекции.
6	Введение в компонентное программирование.	Введение в компонентное программирование: основные низкоуровневые модели распределенных вычислений, этапы развития парадигмы программирования.	Материалы для самостоятельного изучения.
7	Стратегия Microsoft в области распределенных вычислений.	Этапы развития технологий по пути к распределенным вычислениям от DLL до Windows DNA (Distributed interNet Application) и .NET.	-
8	Основы COM (Component Object Model).	Место COM в архитектуре OSI-RM. Последовательности протоколов. Объектная модель COM.	Материалы для самостоятельного изучения.
9	COM и объектно-ориентированный подход	COM и объектно-ориентированный подход: полиморфизм, инкапсуляция, механизмы поддержки наследования.	Материалы для самостоятельного изучения.

п/п	Наименование раздела дисциплины	Содержание раздела дисциплины	Реализация раздела дисциплины с помощью онлайн-курса, ЭУМК
10	СОМ-интерфейс: идентификация, спецификация, виды реализаций.	Идентификация СОМ-интерфейсов (механизмы глобальной идентификации), спецификация (стандарты двоичного интерфейса, язык описания интерфейсов IDL), виды реализаций (интерфейс с виртуальной таблицей, диспинтерфейс, дуальный интерфейс).	Материалы для самостоятельного изучения.
11	Классы СОМ-объектов. Создание экземпляров объектов.	Классы СОМ-объектов. Библиотека СОМ: основные задачи. Фабрика класса.	Материалы для самостоятельного изучения.
12	Многопоточность в СОМ. Механизмы повторного использования СОМ-объектов.	Модели многопоточности. Механизмы повторного использования: включение, агрегирование.	Материалы для самостоятельного изучения.
13	Маршалинг - механизм взаимодействия сервера и клиента.	Маршалинг: назначение и реализация в различных конфигурациях распределенного приложения. Статическое и динамическое связывание.	Материалы для самостоятельного изучения.
14	Информация о типе. Библиотека типа.	Понятие информации о типе. Библиотека типа. Стандартные интерфейсы библиотеки типа для доступа к информации о типе.	Материалы для самостоятельного изучения.
15	Постоянные данные объектов. Сервисы перманентности. Управление перманентностью.	Постоянные данные (состояние) объектов. Сервисы перманентности (Persistent Service), структурированное хранилище. Управление перманентностью, интерфейсы IPersist*.	Материалы для самостоятельного изучения.
16	Объекты с подключением - технология реализации событий СОМ.	Объекты с подключением (Connectable Objects): реализация событий, входящие и исходящие интерфейсы, объекты-точки связи и приемники (sink).	Материалы для самостоятельного изучения.

п/п	Наименование раздела дисциплины	Содержание раздела дисциплины	Реализация раздела дисциплины с помощью онлайн-курса, ЭУМК
17	Распределенная модель компонентных объектов (Distributed COM).	Особенности создания удаленного объекта, механизмы доступа к удаленному объекту (экспортер объектов, формат объектной ссылки, инфраструктура разрешателей OXID), обеспечение контроля доступа к объектам распределенного приложения (защита активизации и защита вызовов).	Материалы для самостоятельного изучения.
18	Понятие об Enterprise JavaBeans (EJB).	Понятие об Enterprise JavaBeans - спецификации набора распределенных Java-компонентов для расширения Java средствами поддержки межплатформенной объектной технологии.	-
19	Введение в Common Object Request Broker Architecture (CORBA).	CORBA, как часть Object Management Architecture (OMA). Место CORBA в модели OSI-RM. Основные спецификации протоколов.	-
20	Объектная модель CORBA. Язык спецификации интерфейсов IDL.	Основные понятия объектной модели CORBA. Основные объектные сервисы. Язык спецификации интерфейсов IDL. Виды интерфейсов и механизмов взаимодействия объектов.	-
21	Структура клиентского приложения. Отображение IDL к языку программирования.	Структура клиентского приложения. Отображение IDL к языку программирования.	-

п/п	Наименование раздела дисциплины	Содержание раздела дисциплины	Реализация раздела дисциплины с помощью онлайн-курса, ЭУМК
22	Структура и реализации брокера ORB (Object Request Broker).	Структура брокера ORB: ядро, интерфейс динамического вызова, объектные каркасы и стабы. Объектный адаптер (Object Adapter): назначение, понятие стиля объектной реализации, базовый и целевые объектные адаптеры. Основные реализации ORB.	-
23	Основы архитектуры .NET Microsoft.	Архитектура .NET и платформа .NET Framework. Общеязыковая среда выполнения CLR (Common Language Runtime) , общий промежуточный язык CIL (Common Intermediate Language). Управляемый код и управляемые данные.	-
24	Система типов .NET.	Система типов: общая система типов CTS (Common Type System), типы-значения и ссылочные типы. Обеспечение ООП: поля, методы, свойства, события. Тип делегат.	-
25	Система метаданных CLR.	Система метаданных CLR: хранение метаданных вместе с типом, концепция отражения, механизмы динамического связывания, метапрограммирование, атрибуты – средство расширения метаданных. Состав и структура хранения сборки.	-
26	Систма выполнения CLR.	Система выполнения: JIT-компиляция и верификация, управление доменами приложений, управление памятью и алгоритм сборки мусора.	-



п/п	Наименование раздела дисциплины	Содержание раздела дисциплины	Реализация раздела дисциплины с помощью онлайн-курса, ЭУМК
27	Механизмы развертывания сборок.	Механизмы развертывания сборок. Глобальный кэш и кэш загрузки. Строгое имя сборки. Схемы цифрового подписания и проверки аутентичности сборки. Механизмы контроля версий сборки. Интернационализация и локализация приложений.	-
28	Особенности разработки распределенных приложений для функционирования в среде Microsoft RPC.	Лабораторная работа №1. Изучение концепции вызова удаленных процедур (RPC), основ языка определения интерфейсов (IDL) и схемы сборки распределенного приложения. Изучение механизма вызова RPC и динамического связывания. Изучение особенностей разработки распределенных приложений для функционирования в среде Microsoft RPC. Реализация тестового распределенного приложения и демонстрация его работы при различных способах размещения клиентской и серверной частей и различных последовательностях протоколов. Выполнение индивидуального задания по разработке распределенного приложения на основе Microsoft RPC.	Демонстрация выполнения основных операций в формате конференции.

п/п	Наименование раздела дисциплины	Содержание раздела дисциплины	Реализация раздела дисциплины с помощью онлайн-курса, ЭУМК
29	Начальные сведения о технологии OLE-автоматизации. Разработка простого контроллера автоматизации для одного из приложений Microsoft Office.	Лабораторная работа №2. Изучение основ технологии OLE-автоматизации. Реализация учебного примера контроллера автоматизации для одного из приложений Microsoft Office. Демонстрация его возможностей по формированию отчетов с использованием тестовой базы данных. Выполнение индивидуального задания по разработке контроллера автоматизации для одного из приложений Microsoft Office.	Демонстрация выполнения основных операций в формате конференции.
30	Введение в технологию Borland MIDAS (Multi-tier Distributed Application Services Suite) для разработки многозвенных распределенных приложений и их эксплуатации в корпоративных системах.	Лабораторная работа №3. Изучение основ технологии Borland MIDAS для разработки многозвенных распределенных приложений и их эксплуатации в корпоративных системах. Реализация в технологии учебного распределенного приложения: сервер приложений с доступом к тестовой базе данных и "тонкий" клиент с интерфейсом к данным. Выполнение индивидуального задания по разработке трехзвенного распределенного приложения в технологии Borland MIDAS.	Демонстрация выполнения основных операций в формате конференции.

### 13.2. Темы (разделы) дисциплины и виды занятий

№ п/п	Наименование темы (раздела)	Лекционные занятия	Практические занятия	Лабораторные занятия	Самостоятельная работа	Всего
1	Эволюция технологии доступа к данным в информационных системах.	0.5		0	2	2
2	Понятие DCE (Distributed Computing Environment).	0.5		0	2	2

№ п/п	Наименование темы (раздела)	Лекционные занятия	Практические занятия	Лабораторные занятия	Самостоятельная работа	Всего
3	Архитектура средств DCE OSF (Open Software Foundation).	0.5		0	2	2
4	Концепция вызова удаленных процедур (Remote Procedure Call, RPC) на примере RPC DCE OSF.	0.5		0	2	2
5	Механизм вызова RPC.	0.5		2	2	4
6	Введение в компонентное программирование.	0.5		0	2	2
7	Стратегия Microsoft в области распределенных вычислений.	0.5		0	2	2
8	Основы COM (Component Object Model).	0.5		1	2	3
9	COM и объектно-ориентированный подход	0.5		0	2	2
10	COM-интерфейс: идентификация, спецификация, виды реализаций.	0.5		1	2	3
11	Классы COM-объектов. Создание экземпляров объектов.	0.5		1	2	3
12	Многопоточность в COM. Механизмы повторного использования COM-объектов.	0.5		0	2	2
13	Маршалинг - механизм взаимодействия сервера и клиента.	0.5		0	2	2
14	Информация о типе. Библиотека типа.	0.5		2	2	4
15	Постоянные данные объектов. Сервисы перманентности. Управление перманентностью.	0.5		0	2	2
16	Объекты с подключением - технология реализации событий COM.	0.5		0	2	2

№ п/п	Наименование темы (раздела)	Лекционные занятия	Практические занятия	Лабораторные занятия	Самостоятельная работа	Всего
17	Распределенная модель компонентных объектов (Distributed COM).	0.5		2	2	4
18	Понятие об Enterprise JavaBeans (EJB).	0.5		0	2	2
19	Введение в Common Object Request Broker Architecture (CORBA).	0.5		0	3	3
20	Объектная модель CORBA. Язык спецификации интерфейсов IDL.	0.5		0	3	3
21	Структура клиентского приложения. Отображение IDL к языку программирования.	0.5		0	3	3
22	Структура и реализации брокера ORB (Object Request Broker).	0.5		0	3	3
23	Основы архитектуры .NET Microsoft.	0.5		0	3	3
24	Система типов .NET.	0.5		0	3	3
25	Система метаданных CLR.	0		0	3	3
26	Систма выполнения CLR.	0		0	3	3
27	Механизмы развертывания сборок.	0		0	3	3
28	Особенности разработки распределенных приложений для функционирования в среде Microsoft RPC.	0		5	3	8
29	Начальные сведения о технологии OLE-автоматизации. Разработка простого контроллера автоматизации для одного из приложений Microsoft Office.	0		5	3	8

№ п/п	Наименование темы (раздела)	Лекционные занятия	Практические занятия	Лабораторные занятия	Самостоятельная работа	Всего
30	Введение в технологию Borland MIDAS (Multi-tier Distributed Application Services Suite) для разработки многозвенных распределенных приложений и их эксплуатации в корпоративных системах.	0		5	3	8
31	Итого:	12		24	72	108

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

**При использовании дистанционных образовательных технологий** и электронного обучения выполнять все указания преподавателей, вовремя подключаться к online занятиям, ответственно подходить к заданиям для самостоятельной работы.

**Внеаудиторная самостоятельная работа** студентов включает проработку материалов лекций, изучение рекомендованной литературы, подготовку к лабораторным работам и их защитам, подготовку к устному опросу.

**Самостоятельная работа в аудитории** выполняется под непосредственным руководством преподавателя на лабораторных занятиях, призванных обеспечить выборочное использование лекционного материала для более глубокого изучения организации распределенного вычислительного процесса, либо распределенного приложения на локальной сети персональных компьютеров с использованием изучаемых средств и технологий разработки. Для повышения эффективности руководства при проведении лабораторных занятий, демонстрирующих варианты использования отдельных стандартных технологических приемов для решения соответствующих практических задач, необходимо ссылаться на материалы лекций, показывая место решаемых задач в общем плане изложения. Для обеспечения каждого студента индивидуальным комплектом из трех обязательных для выполнения задач, имеется набор заданий соответствующего объема. Проверка результатов выполнения работы включает требования возможной масштабируемости приложения на количество компонентов гетерогенной системы, а также хорошего стиля программирования, специфического для используемого языка разработки. Демонстрации каждого выполненного индивидуального лабораторного задания предшествует обязательная проверка функционирования соответствующего набора тренировочных задач, самостоятельно реализованных студентом.

Учитывая разницу темпов выполнения индивидуальных заданий, преподаватель обеспечивает выполнение студентами дополнительных заданий (общих для студентов, решивших основную задачу) с целью углубленного изучения материала.

#### **15. Перечень основной и дополнительной литературы, ресурсов интернет, необходимых для освоения дисциплины**

№ п/п	Источник
1	Чуешев, А.В. Распределенные информационные системы : учебно-методическое пособие : [16+] / А.В. Чуешев ; Кемеровский государственный университет. – Кемерово : Кемеровский государственный университет, 2019. – 252 с. : ил. – Режим доступа: по подписке. – URL: <a href="https://biblioclub.ru/index.php?page=book&amp;id=571521">https://biblioclub.ru/index.php?page=book&amp;id=571521</a>

б) дополнительная литература:

№ п/п	Источник
1	Чуешев, А.В. Интеграция данных : учебно-методическое пособие / А.В. Чуешев ; Кемеровский государственный университет. – Кемерово : Кемеровский государственный университет, 2018. – 281 с. : ил. – Режим доступа: по подписке. – URL: <a href="https://biblioclub.ru/index.php?page=book&amp;id=495177">https://biblioclub.ru/index.php?page=book&amp;id=495177</a>

в) информационные электронно-образовательные ресурсы:

№ п/п	Источник
1	<a href="http://www.lib.vsu.ru">http://www.lib.vsu.ru</a> ЗНБ ВГУ
2	<a href="https://edu.vsu.ru">https://edu.vsu.ru</a> Образовательный портал "Электронный университет ВГУ"

**16. Перечень учебно-методического обеспечения для самостоятельной работы**

№ п/п	Источник
1	Рекомендованная программой литература.
2	Конспекты лекций.
3	Комплект учебных приложений с подробными инструкциями по их сборке для первоначального освоения предлагаемых для выполнения лабораторных задач технологий.

**17. Образовательные технологии, используемые при реализации учебной дисциплины, включая дистанционные образовательные технологии (ДОТ), электронное обучение (ЭО), смешанное обучение):**

В качестве интегрированной среды разработки при проведении лабораторных работ студентам рекомендуется использование установленного в компьютерных классах факультета компьютерных наук лицензионного программного обеспечения Microsoft Visual Studio и свободно распространяемой системы программирования Borland Turbo-Delphi для реализации собственных программных разработок. В качестве учебного сервера баз данных рекомендуется использование сервера Oracle факультета компьютерных наук с учебной базой данных, разработанной для проведения занятий по управлению данными. Освоение технологии OLE-автоматизации предполагает использование лицензионного программного обеспечения Microsoft Office,

установленного на компьютерах факультетских лабораторий.

Внеаудиторная самостоятельная работа, как правило, предполагает использование студентами среды разработки Microsoft Visual Studio, официальная процедура установки которой на собственные компьютеры студентов обеспечена факультетом компьютерных наук, а также среды разработки Turbo-Delphi. Тем не менее, студенты никак не ограничиваются требованиями к их выбору средств разработки для использования в самостоятельной работе. Для имитации сервера баз данных в условиях самостоятельной работы студентам рекомендуется использование локальных таблиц с данными и соответствующих стандартных интерфейсов и компонентов доступа к ним.

При реализации дисциплины могут использоваться технологии электронного обучения и дистанционные образовательные технологии на базе портала edu.vsu.ru, а также другие доступные ресурсы сети Интернет.

#### **18. Материально-техническое обеспечение дисциплины:**

компьютерные классы факультета компьютерных наук, лицензионное программное обеспечение Microsoft Visual Studio, свободно распространяемая система программирования Borland Turbo-Delphi, учебный сервер Oracle факультета с базой данных, разработанной для проведения занятий по управлению данными, лицензионное программное обеспечение Microsoft Office.

#### **19. Оценочные средства для проведения текущей и промежуточной аттестаций**

Порядок оценки освоения обучающимися учебного материала определяется содержанием следующих разделов дисциплины:

№ п/п	Разделы дисциплины (модули)	Код компетенции	Код индикатора	Оценочные средства для текущей аттестации
1	Разделы 1 – 5, 28 Эволюция технологии доступа к данным в информационных системах. Понятие DCE (Distributed Computing Environment). Архитектура средств DCE OSF (Open Software Foundation). Концепция вызова удаленных процедур (Remote Procedure Call, RPC) на примере RPC DCE OSF. Механизм вызова RPC. Особенности разработки распределенных приложений для функционирования в среде Microsoft RPC. Разделы 10 – 16, 29 COM-интерфейс: идентификация, спецификация, виды реализаций. Классы COM-объектов. Создание экземпляров объектов. Многопоточность в COM. Механизмы повторного использования COM-объектов. Маршалинг - механизм взаимодействия сервера и клиента. Информация о типе. Библиотека типа. Начальные сведения о технологии OLE-автоматизации. Разработка простого контроллера автоматизации для одного из приложений Microsoft Office. Разделы 15 – 17, 30 Постоянные данные объектов. Сервисы перманентности. Управление перманентностью. Объекты с подключением - технология реализации событий COM. Распределенная модель компонентных объектов (Distributed COM). Введение в технологию Borland MIDAS (Multi-tier Distributed Application Services Suite) для разработки многозвенных распределенных приложений и их эксплуатации в корпоративных системах.	ПК-1 ПК-2	ПК-1.1 ПК-2.2	Комплект задач №1
		ПК-6 ПК-10	ПК-6.1 ПК-10.2	Комплект задач №2
		ПК-14 ПК-14	ПК-14.1 ПК-14.2	Комплект задач №3

Промежуточная аттестация

Форма контроля - Зачет

Оценочные средства для промежуточной аттестации

По результатам трех текущих аттестаций

**20 Типовые оценочные средства и методические материалы, определяющие процедуры оценивания**

**20.1 Текущий контроль успеваемости**



Контроль успеваемости по дисциплине осуществляется с помощью следующих оценочных средств.

### **20.1.1. Примерная структура тренировочных учебных примеров**

1. Реализация тестового распределенного приложения и демонстрация его работы при различных способах размещения клиентской и серверной частей и различных последовательностях протоколов.
2. Реализация учебного примера контроллера автоматизации для одного из приложений Microsoft Office с использованием различных механизмов связывания: позднее (динамическое), раннее (импорт информации о типе). Демонстрация его возможностей по формированию отчетов с использованием тестовой базы данных.
3. Реализация в технологии Borland MIDAS учебного распределенного приложения: сервер приложений с доступом к тестовой базе данных и "тонкий" клиент с интерфейсом к данным. Использование различных протоколов обмена клиента с сервером приложений, включая SOAP. В последнем случае сервер приложений реализуется в виде web-службы и размещается на отладочном web-сервере.

### **20.1.2. Перечень практических заданий**

Индивидуальное задание №1 Разработка специфического распределенного приложения для функционирования в среде Microsoft RPC. Основная цель – изучение концепции вызова удаленных процедур (RPC), основ языка определения интерфейсов (IDL) и схемы сборки распределенного приложения. Изучение механизма вызова RPC и динамического связывания. Оценка «зачтено» выставляется, если разработанное согласно индивидуальному заданию распределенное приложение реализует соответствующую модель информационного взаимодействия программных компонентов, при этом приложение должно включать, как минимум, два таких компонента: клиентскую и серверную части, выполненные в виде консольных приложений, соединенных по сетевому протоколу и работающих на разных компьютерах локальной сети. Пример варианта индивидуального задания из комплекта задач №1:

Средствами MS RPC реализуйте модель информационной системы с параллельно работающими несколькими серверами. Клиент непосредственно после запуска производит поиск доступного для подключения сервера.

Индивидуальное задание №2 Разработка контроллера автоматизации для одного из приложений Microsoft Office. Основная цель – изучение возможностей технологии OLE-автоматизации в контексте использования для формирования отчетов по результатам выборок из базы данных. Оценка «зачтено» выставляется, если разработанная согласно индивидуальному заданию программа, выступая в качестве контроллера автоматизации, выполняет необходимые действия по управлению OLE-сервером для формирования отчета по результатам выборки из учебной базы данных. Пример варианта индивидуального задания из комплекта задач №2:

Разработайте контроллер автоматизации, использующий механизм позднего связывания одновременно с двумя серверами: Microsoft Word и Excel; для решения следующей задачи. Таблица в книге Excel должна быть скопирована в документ Word и соответствующим образом отформатирована. При желании можно комбинировать различные варианты механизмов связывания с серверами автоматизации.

Индивидуальное задание №3 Разработка трехзвенного распределенного приложения в технологии Borland MIDAS. Цель: изучение основ технологии Borland MIDAS (Multi-tier Distributed Application Services Suite) для разработки многозвенных распределенных приложений и их эксплуатации в корпоративных системах. Оценка «зачтено» выставляется, если разработанное согласно индивидуальному заданию приложение выполняет необходимые действия в роли модельной трехзвенной информационной системы: на студента ложится ответственность за разработку звеньев тонкого клиента и сервера приложений, слой сервера базы данных необходимо

реализовать подключением к учебному факультетскому серверу, либо имитацией при помощи локальных таблиц в файлах. Пример варианта индивидуального задания из комплекта задач №3: Доработайте учебный пример трехзвенной системы так, чтобы оператор «тонкого» клиента имел возможность удобной работы в автономном режиме (без связи с сервером приложений). При этом необходимо дополнительно предусмотреть интерфейсные команды управления соединением с сервером приложений, команды сохранения кэша редактируемых таблиц в файлах на локальных дисках, восстановления содержимого кэша из файлов, а также команды на передачу результатов редактирования таблиц серверу приложений для сохранения их в базе данных (при переходе из автономного режима в режим связи с сервером приложений).

### **20.1.3. Описание технологии проведения**

Текущая аттестация проводится в соответствии с Положением о текущей аттестации обучающихся по программам высшего образования Воронежского государственного университета. Текущая аттестация проводится в формах: устного опроса (индивидуальный опрос, фронтальная беседа) по лекционному материалу, проверки знания основ предложенных технологий по результатам демонстрации тренировочных комплектов приложений и защиты лабораторных работ. При оценивании результатов устного опроса и демонстрации тренировочных комплектов приложений используется качественная шкала оценок. Оценивание результатов и защиты лабораторных работ предполагает использование количественной шкалы.

### **20.1.4. Требования к выполнению заданий (или шкалы и критерии оценивания)**

Конкретные требования к выполнению каждого практического задания изложены в п.20.1.2. Для оценивания качества выполнения заданий используется – зачтено, не зачтено.

## **20.2 Промежуточная аттестация**

Промежуточная аттестация проводится в соответствии с Положением о промежуточной аттестации обучающихся по программам высшего образования. Промежуточная аттестация по итогам освоения дисциплины проводится в форме зачета. Условиями для положительной итоговой оценки являются: выполнение всех тренировочных комплектов приложений и успешные защиты лабораторных работ, перечисленных в приложении под наименованием «индивидуальное задание».

### **20.2.1. Шкалы и критерии оценивания**

При оценивании используется следующая шкала:

5 баллов ставится, если обучающийся демонстрирует полное соответствие знаний, умений, навыков приведенным в таблицах показателям, свободно оперирует приобретенными знаниями, умениями, применяет их при решении практических задач;

4 балла ставится, если обучающийся демонстрирует соответствие знаний, умений, навыков приведенным в таблицах показателям, но допускает незначительные ошибки, неточности, испытывает затруднения при решении практических задач;

3 балла ставится, если обучающийся демонстрирует неполное соответствие знаний, умений, навыков приведенным в таблицах показателям, допускает значительные ошибки при решении практических задач;

2 балла ставится, если обучающийся демонстрирует явное несоответствие знаний, умений, навыков приведенным в таблицах показателям.

При сдаче зачета

«зачтено» - 3-5 баллов

«не зачтено» - 2 балла.

Для оценивания результатов обучения на зачете используются следующие показатели:

1. знание основных принципов организации распределенных приложений, вычислительных и

- операционных систем; наиболее употребительных моделей распределенного программирования, в том числе, процедурной, компонентной и Grid, принципов функционирования соответствующей инфраструктуры их поддержки в гетерогенной среде, стандартов на последовательности протоколов разных уровней и архитектуру систем программирования для реализации распределенных вычислений;
2. умение применять перечисленные сведения для проектирования систем распределенных вычислений;
  3. владение навыками реализации распределенных приложений с использованием широко распространенных технологий и моделей программирования.

Соотношение показателей, критериев и шкалы оценивания результатов обучения.

Критерии оценивания компетенций	Уровень сформированности компетенций	Шкала оценок
<p><i>Сформированные знания об основных принципах организации распределенных приложений, вычислительных и операционных систем; наиболее употребительных моделях распределенного программирования, в том числе, процедурной, компонентной и Grid, принципов функционирования соответствующей инфраструктуры их поддержки в гетерогенной среде, стандартов на последовательности протоколов разных уровней и архитектуру систем программирования для реализации распределенных вычислений. Сформированное умение применять перечисленные сведения для проектирования систем распределенных вычислений. Сформированные навыки реализации распределенных приложений с использованием широко распространенных технологий и моделей программирования.</i></p>	<p><i>Повышенный уровень</i></p>	<p><i>Отлично</i></p>
<p><i>Сформированные, но содержащие отдельные пробелы представления об основных принципах организации распределенных приложений, вычислительных и операционных систем; наиболее употребительных моделях распределенного программирования, в том числе, процедурной, компонентной и Grid, принципах функционирования соответствующей инфраструктуры их поддержки в гетерогенной среде, стандартах на последовательности протоколов разных уровней и архитектуру систем программирования для реализации распределенных вычислений. Успешное, но содержащее отдельные пробелы умение применять перечисленные сведения для проектирования систем распределенных вычислений. Сформированные, но содержащие отдельные пробелы навыки реализации распределенных приложений с использованием широко распространенных технологий и моделей программирования.</i></p>	<p><i>Базовый уровень</i></p>	<p><i>Хорошо</i></p>

<p><i>Неполное представление об основных принципах организации распределенных приложений, вычислительных и операционных систем; наиболее употребительных моделях распределенного программирования, в том числе, процедурной, компонентной и Grid, принципах функционирования соответствующей инфраструктуры их поддержки в гетерогенной среде, стандартах на последовательности протоколов разных уровней и архитектуру систем программирования для реализации распределенных вычислений. Успешное, но не системное умение применять перечисленные сведения для проектирования систем распределенных вычислений. Неполное владение навыками реализации распределенных приложений с использованием широко распространенных технологий и моделей программирования средств.</i></p>	<p><i>Пороговый уровень</i></p>	<p><i>Удовлетворительно</i></p>
<p><i>Фрагментарные знания или отсутствие знаний. Фрагментарные умения или отсутствие умений. Фрагментарные навыки или отсутствие навыков.</i></p>	<p><i>-</i></p>	<p><i>Неудовлетворительно</i></p>