

МЕЖГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
«БЕЛОРУССКО-РОССИЙСКИЙ УНИВЕРСИТЕТ»

Кафедра «Автоматизированные системы управления»

РАСПРЕДЕЛЕННЫЕ СИСТЕМЫ ОБРАБОТКИ ИНФОРМАЦИИ

*Методические рекомендации к самостоятельной работе
для студентов специальности
1-28 01 02 «Электронный маркетинг»
заочной формы обучения*



Могилев 2022

УДК 004.43
ББК 32.973.2
Р24

Рекомендовано к изданию
учебно-методическим отделом
Белорусско-Российского университета

Одобрено кафедрой «Автоматизированные системы управления»
«08» декабря 2021 г., протокол № 5

Составитель ст. преподаватель Д. А. Денисевич

Рецензент канд. техн. наук, доц. В. В. Кутузов

Даны методические рекомендации к самостоятельной работе по дисциплине
«Распределенные системы обработки информации». Приведены сведения по
изучаемым темам, вопросы к экзамену, список литературы.

Учебно-методическое издание

РАСПРЕДЕЛЕННЫЕ СИСТЕМЫ ОБРАБОТКИ ИНФОРМАЦИИ

Ответственный за выпуск	А. И. Якимов
Корректор	Т. А. Рыжикова
Компьютерная верстка	Н. П. Полевничая

Подписано в печать . Формат 60×84/16. Бумага офсетная. Гарнитура Таймс.
Печать трафаретная. Усл. печ. л. . Уч.-изд. л. . Тираж 26 экз. Заказ №

Издатель и полиграфическое исполнение:
Межгосударственное образовательное учреждение высшего образования
«Белорусско-Российский университет».
Свидетельство о государственной регистрации издателя,
изготовителя, распространителя печатных изданий
№ 1/156 от 07.03.2019.
Пр-т Мира, 43, 212022, г. Могилев.

© Белорусско-Российский
университет, 2022

Содержание

Введение.....	4
1 Содержание программы дисциплины.....	5
2 Требования к содержанию и оформлению аудиторных контрольных работ	8
3 Перечень вопросов к первой аудиторной контрольной работе	8
4 Пример выполнения аудиторной контрольной работы № 1	10
5 Перечень вопросов ко второй аудиторной контрольной работе.....	12
6 Пример выполнения аудиторной контрольной работы № 2	13
Список литературы	14

Введение

Дисциплина «Распределенные системы обработки информации» предоставляет базовые знания для моделирования, проектирования и программной разработки распределенных задач, на основе современных методов, лучших решений и современных технологий.

Цель методических рекомендаций – предоставление материалов для подготовки студентов к выполнению аудиторной контрольной работы, в которых рассматриваются расширенные возможности и перспективы применения языка Java в распределенных корпоративных системах и веб-сервисах. Изучаются также основы архитектуры и программирования в .NET с применением языка C# и современных технологий на основе .NET.

Аудиторная контрольная работа (АКР) является формой промежуточного контроля результатов межсессионной самостоятельной работы студентов заочной (в том числе сокращенной) форм обучения по дисциплинам учебного плана специальности, одним из элементов системы контроля за качеством обучения. АКР организуется в период проведения лабораторно-экзаменационной сессии в соответствии с расписанием.

1 Содержание программы дисциплины

Введение. Предмет и содержание дисциплины. Структура курса. Цель и основные изучаемые направления и границы предметной области. Особенности разработки распределенных информационных и корпоративных систем.

Архитектура и основные механизмы современных платформ разработки распределенных приложений. Распределенные информационные системы. Понятие и классификация распределенных систем. Определение, задачи, основные концепции и принципы аппаратных, программных и архитектурных решений распределенных информационных систем. Архитектурные программные решения. Связывание, принципы и основные аспекты взаимодействия. Проблемы и общие принципы реализации, основные методы и технологии, типизация и стандартизации лучших решений и процедур.

Синхронное и асинхронное взаимодействие в распределенных приложениях. Методы и механизмы взаимодействия элементов распределенной информационной системы. Взаимодействие с распределенными объектами. Статистическое и динамическое установления связей и организации вызовов. Понятие, общие принципы и механизмы, сравнение возможностей, преимуществ и недостатков синхронных и асинхронных взаимодействий. Понятие и применение Message-Oriented Middleware (MOM) и основные модели обмена сообщениями. Основные решения для конкретных платформ и механизмов.

Транспортные протоколы и промежуточные средства обеспечения взаимодействий приложений. Методы, средства и протоколы установления связи и обеспечения взаимодействия распределенных элементов системы. Реализации обращений к удаленным объектам и связи посредством передачи сообщений. Основные принципы и механизмы передачи сообщений. Протоколы взаимодействия и передачи сообщений. Потoki и процессы организации взаимодействия. Общие принципы и программная реализация разработки. Использование интерфейсов, перенос кода и применение программных агентов для взаимодействия элементов системы. Методы и программные алгоритмы синхронизации. Распределенные системы объектов, основные принципы реализации, архитектурные решения, методы обеспечения системной целостности и надежности выполнения прикладных программных решений.

Архитектурные решения и технологии разработки сервисно-ориентированных приложений. Веб-сервисы, определение, назначение и основные решаемые задачи. Основы архитектуры и используемые технологии, взаимодействие с прикладными системами. Основные принципы разработки сервисно-ориентированной (SOA) архитектуры. Веб-службы, методы и механизмы. Обнаружение, описание, обработка и передача данных.

Разработка веб-сервисов на основе современных платформ и языков программирования. Создание сервисов на основе Java. Написание и ручное развертывание WSD. Передача данных. Создание и использование WSDL. Протоколы связывания и передача данных по SOAP. SOAP и HTTP. Вызов веб-служб. Обработка, отображения и привязки в SOAP. Публикация и обнаружение веб-служб. Технология и модель данных UDDI. Основные программные

интерфейсы UDDI. Сценарии использования и применение WSDL и SOAP в UDDI. Интеграция и организация взаимодействия веб-служб и основных технологий. Организация и технологии документно-ориентированного взаимодействия.

Распределенные системы согласования. Распределенные системы согласования. Архитектура Jini. Основные элементы и их назначение. Использование событий в Jini. Обнаружение сервисов. Сервис JavaSpaces. Служба поиска Jini. Посредники сервиса. Модель аренды в Jini (Jini Technology's Distributed Leasing Model). Синхронизация и модель организации транзакций Jini (Jini Technology's Distributed Transaction Model). Реализации сервиса и клиента Jini. Вспомогательные службы и утилиты.

Архитектурные решения и технологии разработки сервисно-ориентированных приложений. Веб-сервисы, определение, назначение и основные решаемые задачи. Основы архитектуры и используемые технологии, взаимодействие с прикладными системами. Основные принципы разработки сервисно-ориентированной (SOA) архитектуры. Веб-службы, методы и механизмы. Обнаружение, описание, обработка и передача данных.

Платформа .NET фирмы Microsoft. Понятие Framework: определение, назначение, архитектура и особенности реализации программных приложений на используемых платформах. Инициатива Microsoft. Базовая операционная система .NET Framework: назначение, характеристики, архитектура. Основные принципы функционирования. Управляемый и неуправляемый код. Выполнение и стыковка приложений. Платформа .NET Framework, назначение архитектурные решения, функционирование и обеспечивающие механизмы.

Управляемые модули, сборки, исполнение программ. NET Framework: виды приложений, общая система типов. Назначение и определение свойств классов в .NET. Компоновка, упаковка, развертывание и администрирование приложений и типов данных.

Современные языковые и технологические средства разработки программных приложений. Проектирование типов. Методы, свойства и события. Обработка и управление основными типами данных. Создание собственных типов. Понятие и особенности разработки и использования интерфейсов. Управление типами и памятью. Виды и технология создания приложений.

Особенности разработки программ на C#. Концептуальные особенности разработки программных приложений на C#. Основные элементы и абстракции языка C#. Особенности синтаксиса и семантики языка. Встроенные и создаваемые типы. Строковые данные. Особенности разработки приложений на C#. Делегаты и события. Поток, сборки и удаленные вызовы. Разработка программных приложений на C# в среде .NET для Windows.

Технологии работы с данными и системами управления базами данных (СУБД) в применяемых современных платформах программирования. Концепция и архитектура доступа к данным ADO.NET. Объектная модель ADO.NET. Понятие и принципы отсоединенной обработки данных. Основные классы и объекты. Источники данных в ADO.NET(провайдеры). Классы SqlConnection и OleDbConnection. Соединения, пул соединений. Класс Command и выполнение

команд. DataAdapter.

Разработка распределенных информационных приложений на базе применяемых современных платформ, средств и систем и используемой платформы. Основы и особенности разработки приложений на платформе ASP.NET. Среда и средства разработки. Привязка данных к пользовательскому интерфейсу. Элементы управления и разработка интерфейса. Основы и методы управления состоянием. Модель безопасности и масштабируемость разработки.

Лучшие типовые решения и паттерны проектирования для разработки программных систем. Современные средства эффективного проектирования архитектурных и программных решений. Стандарты и средства UML. Виды диаграмм. Статические и поведенческие диаграммы. Уровни абстракции. Сквозное проектирование моделей. Виды паттернов. Общая структура и описание.

Применение паттернов, не зависящих и ориентированных на платформу программирования. Паттерны проектирования, понятие и применение паттернов в информатике. Паттерны не зависящие и ориентированные на программную платформу. Типы паттернов. Паттерны проектирования и применения паттернов для разработки архитектурных решений. Паттерны тестирования. Архитектурные паттерны. Разработка и преимущества разработки эффективных программных систем с применением паттернов. Понятие и примеры антипаттернов.

Объектные распределенные системы. Вызов удаленных процедур. Организация связи с использованием удаленных объектов . CORBA.

Агентные технологии. Понятие программного агента. Мультиагентные системы. Безопасность в системах мобильных агентов.

Компонентные системы. Основы компонентных программных систем. Концепция JavaBeans.

Технологии одноранговых сетей. Основы технологии одноранговых сетей. Алгоритмы работы P2P сетей. Применение технологий P2P. Достоинства и недостатки P2P.

Технологии Грид. Архитектура Грид. Стандарты Грид. Система Globus. Система UNICORE. Параметрические модели производительности Грид.

Облачные вычисления. Определение облачных вычислений. Многослойная архитектура облачных приложений. Компоненты облачных приложений. Достоинства и недостатки облачных вычислений. Классификация облаков. Наиболее распространенные облачные платформы. Сравнение Грид и облачных вычислений.

Возможности, архитектура и средства разработки приложений развивающихся перспективных технологий. Развивающиеся платформы и приложения. Концепция принципы и основные проектные решения. Архитектурная модель. Основные классы, объекты и методы реализации. Инструментальные средства.

2 Требования к содержанию и оформлению аудиторных контрольных работ

По дисциплине выполняется две аудиторные контрольные работы (АКР).

На аудиторную контрольную работу № 1 выносятся вопросы, указанные в перечне вопросов к первой аудиторной контрольной работе.

На аудиторную контрольную работу № 2 выносятся вопросы, указанные в перечне вопросов ко второй аудиторной контрольной работе.

Аудиторные контрольные работы включают по одному вопросу из представленного перечня вопросов.

АКР оформляются на стандартных бланках объёмом один лист формата А4, выдаваемых преподавателем. Ответы на вопросы должны конспективно раскрывать суть вопросов. При этом рекомендуется приводить примеры вычислений, команд, программного кода и т. д. Объём ответа на каждый вопрос определяется студентом с учетом вышесказанного.

Продолжительность выполнения АКР по данной дисциплине составляет два академических часа.

3 Перечень вопросов к первой аудиторной контрольной работе

1 Особенности разработки распределенных информационных и корпоративных систем. Понятие и классификация распределенных систем.

2 Определение, задачи, основные концепции и принципы аппаратных, программных и архитектурных решений распределенных информационных систем.

3 Архитектурные программные решения. Связывание, принципы и основные аспекты взаимодействия. Проблемы и общие принципы реализации, основные методы и технологии, типизация и стандартизации лучших решений и процедур.

4 Методы и механизмы взаимодействия элементов распределенной информационной системы. Взаимодействие с распределенными объектами. Статистическое и динамическое установления связей и организации вызовов.

5 Понятие, общие принципы и механизмы, сравнение возможностей, преимуществ и недостатков синхронных и асинхронных взаимодействий.

6 Понятие и применение Message-Oriented Middleware (МОО) и основные модели обмена сообщениями. Основные решения для конкретных платформ и механизмов.

7 Методы, средства и протоколы установления связи и обеспечения взаимодействия распределенных элементов системы. Реализации обращений к удаленным объектам и связи посредством передачи сообщений.

8 Основные принципы и механизмы передачи сообщений. Протоколы взаимодействия и передачи сообщений. Потoki и процессы организации

взаимодействия.

9 Общие принципы и программная реализация разработки. Использование интерфейсов, перенос кода и применение программных агентов для взаимодействия элементов системы.

10 Общие принципы и программная реализация разработки. Методы и программные алгоритмы синхронизации.

11 Распределенные системы объектов, основные принципы реализации, архитектурные решения, методы обеспечения системной целостности и надежности выполнения прикладных программных решений.

12 Веб-сервисы, определение, назначение и основные решаемые задачи.

13 Основные принципы разработки сервисно-ориентированной (SOA) архитектуры. Веб-службы, методы и механизмы. Обнаружение, описание, обработка и передача данных.

14 Создание сервисов на основе Java.

15 Написание и ручное развертывание WSDD. Передача данных. Создание и использование WSDL.

16 Протоколы связывания и передача данных по SOAP. SOAP и HTTP. Вызов веб-служб. Обработка, отображения и привязки в SOAP.

17 Публикация и обнаружение веб-служб. Технология и модель данных UDDI. Основные программные интерфейсы UDDI.

18 Сценарии использования и применение WSDL и SOAP в UDDI. Интеграция и организация взаимодействия веб-служб и основных технологий.

19 Распределенные системы согласования. Архитектура Jini. Основные элементы и их назначение.

20 Использование событий в Jini (Jini Technology's Distributed Transaction Model). Обнаружение сервисов. Сервис JavaSpaces.

21 Служба поиска Jini (Jini Technology's Distributed Transaction Model). Посредники сервиса. Модель аренды в Jini. Синхронизация и модель организации транзакций Jini. Реализации сервиса и клиента Jini. Вспомогательные службы и утилиты.

22 Веб-сервисы, определение, назначение и основные решаемые задачи.

23 Основы архитектуры и используемые технологии, взаимодействие с прикладными системами. Основные принципы разработки сервисно-ориентированной (SOA) архитектуры.

24 Веб-службы, методы и механизмы. Обнаружение, описание, обработка и передача данных.

25 Инициатива Microsoft. Базовая операционная система .NET Framework: назначение, характеристики, архитектура. Основные принципы функционирования.

26 Управляемый и неуправляемый код. Выполнение и стыковка приложений. Платформа .NET Framework: назначение, архитектурные решения, функционирование и обеспечивающие механизмы.

27 NET Framework: виды приложений, общая система типов. Назначение и определение свойств классов в .NET. Компоновка, упаковка, развертывание и администрирование приложений и типов данных.

28 NET Framework. Проектирование типов. Методы, свойства и события. Обработка и управление основными типами данных. Создание собственных типов. Понятие и особенности разработки и использования интерфейсов. Управление типами и памятью. Виды и технология создания приложений.

29 Концептуальные особенности разработки программных приложений на С#. Основные элементы и абстракции языка С#. Особенности синтаксиса и семантики языка. Встроенные и создаваемые типы. Строковые данные.

30 Особенности разработки приложений на С#. Делегаты и события. Поток, сборка и удаленные вызовы. Разработка программных приложений на С# в среде .NET для Windows.

31 Концепция и архитектура доступа к данным ADO.NET. Объектная модель ADO.NET. Понятие и принципы отсоединенной обработки данных. Основные классы и объекты. Источники данных в ADO.NET(провайдеры).

32 Классы SqlConnection и OleDbConnection. Соединения. Пул соединений. Класс Command и выполнение команд. DataAdapter.

33 Основы и особенности разработки приложений на платформе ASP.NET. Среда и средства разработки. Привязка данных к пользовательскому интерфейсу. Элементы управления и разработка интерфейса. Основы и методы управления состоянием. Модель безопасности и масштабируемость разработки.

4 Пример выполнения аудиторной контрольной работы № 1

Веб-сервисы – это часть бизнес-логики приложения, размещенной в интернете, которая обеспечивает доступ по стандартным интернет-протоколам, таким как HTTP или SMTP.

Главные технологии веб-сервисов:

– SOAP (Simple Object Access Protocol) – стандарт «упаковки» XML-документов для транспортировки по протоколам SMTP, HTTP или FTP;

– WSDL (Web Service Description Language) – это XML-технология, описывающая интерфейсы веб-сервисов. Помогает клиентам автоматически понять, как работать с сервисом;

– UDDI (Universal Description, Discovery and Integration) – обеспечивает всемирную регистрацию web-сервисов. Используется для обнаружения сервиса путем поиска по имени, категории и др.

Взаимодействие между технологиями:

– клиент запрашивает у репозитория UDDI сервис по его имени или идентификатору;

– клиент получает информацию о размещении WSDL-документа от UDDI-репозитория. Он содержит информацию о том, как связаться с сервисом, и формат запроса в XML;

– согласно найденной в WSDL информации клиент создает SOAP-сообщение и посылает на хост сервиса.

Провайдеры публикуют информацию (метаданные) о сервисе в репозитории. Согласно документу Web Services Conceptual Architecture компании IBM для публикации используются несколько различных механизмов.

Механизм публикации прямой.

Инициатор запроса ищет описание сервиса прямо у провайдера сервиса, используя e-mail, FTP, CD. Провайдер доставляет описание сервиса и одновременно делает сервис доступным для инициатора запроса. Как такового репозитория (брокера) здесь нет.

Механизм публикации через HTTP Get-запрос.

Этот механизм используется на www.xmethods.com. Это публичный репозиторий веб-сервисов. Инициатор запросов ищет описание сервиса прямо у провайдера, используя Get-запрос по протоколу HTTP. Эта модель имеет регистр (веб-репозиторий), хотя только в неполном смысле этого слова.

Механизм публикации с помощью динамического обнаружения.

Этот механизм использует локальные публичные репозитории, чтобы хранить и программно отыскивать описания сервисов. Наиболее часто используемый репозиторий – UDDI, хотя есть и другие (например, ebXML R). Эта модель наиболее универсальная и представляет наибольший интерес.

Apache Axis (Apache eXtensible Interaction System) – система для конструирования SOAP-процессоров, таких как клиенты, серверы, шлюзы и др. SOAP – это механизм для коммуникации приложений посредством интернета.

Однако Axis не просто «движок» SOAP, он также включает:

- простой самостоятельный сервер;
- сервер, встраиваемый в контейнеры сервлетов;
- расширенную поддержку WSDL;
- инструменты, генерирующие Java-классы из WSDL;
- примеры программ;
- инструмент для отслеживания TCP/IP-пакетов.

Axis конвертирует Java-объекты в данные SOAP, когда посылает их по сети или получает результаты. Все ошибки, генерируемые сервером, Axis преобразовывает в Java-исключения.

Axis реализует JAX-RPC API – один из стандартных способов программирования Java-сервисов.

Axis поставляется в виде набора jar-библиотек. Реализация JAX-RPC API представлена в файлах *jaxrpc.jar* и *saaj.jar*. Также для работы приложений необходимы вспомогательные библиотеки для логгирования, обработки WSDL. Для работы Axis'a требуется наличие XML-парсера, совместимого с JAXP 1.1 (Java API for XML Processing), предпочтительно Xerces.

5 Перечень вопросов ко второй аудиторной контрольной работе

- 1 Современные средства эффективного проектирования архитектурных и программных решений.
- 2 Стандарты и средства UML.
- 3 UML. Виды диаграмм. Статические и поведенческие диаграммы.
- 4 Уровни абстракции. Сквозное проектирование моделей.
- 5 Виды паттернов. Общая структура и описание.
- 6 Паттерны проектирования, понятие и применение паттернов в информатике. Паттерны, не зависящие и ориентированные на программную платформу.
- 7 Типы паттернов. Паттерны проектирования и применения паттернов для разработки архитектурных решений.
- 8 Паттерны тестирования.
- 9 Архитектурные паттерны.
- 10 Разработка и преимущества разработки эффективных программных систем с применением паттернов.
- 11 Понятие и примеры антипаттернов.
- 12 Вызов удаленных процедур. Организация связи с использованием удаленных объектов. CORBA.
- 13 Понятие программного агента. Мультиагентные системы. Безопасность в системах мобильных агентов.
- 14 Основы компонентных программных систем. Концепция JavaBeans.
- 15 Основы технологии одноранговых сетей.
- 16 Алгоритмы работы P2P сетей. Применение технологий P2P.
- 17 Достоинства и недостатки P2P.
- 18 Архитектура Грид. Стандарты Грид.
- 19 Система Globus. Система UNICORE. Параметрические модели производительности Грид.
- 20 Определение облачных вычислений. Многослойная архитектура облачных приложений.
- 21 Компоненты облачных приложений. Достоинства и недостатки облачных вычислений.
- 22 Классификация облаков. Наиболее распространенные облачные платформы. Сравнение Грид и облачных вычислений.
- 23 Развивающиеся платформы и приложения. Концепция принципы и основные проектные решения.
- 24 Развивающиеся платформы и приложения. Архитектурная модель. Основные классы, объекты и методы реализации. Инструментальные средства.
- 25 Основы и особенности разработки приложений на платформе ASP.NET. Среда и средства разработки.
- 26 Архитектура MVC. Компоненты MVC приложения.

- 27 Проектирование и разработка веб-приложений в модели ASP.NET MVC.
 28 Архитектура доступа к данным ADO.NET. Объектная модель.

6 Пример выполнения аудиторной контрольной работы № 2

ASP.NET – это часть технологии .NET, используемая для написания мощных клиент-серверных интернет-приложений. Она позволяет создавать динамические страницы HTML на основе множества готовых элементов управления, применяя которые можно быстро создавать интерактивные веб-сайты.

Формы ASP.NET.

Основным средством взаимодействия веб-приложения с пользователем являются веб-формы, представляющие собой аналог оконных форм Windows.

При создании формы программист вначале рисует форму в визуальном редакторе с помощью набора элементов управления, а затем настраивает обработку событий для формы. Обработчики могут относиться как к уровню страницы, так и к отдельному компоненту формы.

Рассмотрим простой пример формы:

```
<%@ Page Language="C#" AutoEventWireup="true" CodeFile="Default2.aspx.cs"
Inherits="Default2" %>
```

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

```
<!doctype html>
<head runat="server">
</head>
<body>
<form id="form1" runat="server">
<div>
<asp:Label ID="Label1" runat="server" Text="Введите свое имя"> </asp:Label>
<asp:TextBox ID="TextBox1" runat="server" Height="46px" Width="257px">
</asp:TextBox>
<asp:Button ID="Button1" runat="server" Text="Отправить" onclick="Button1_Click" />
</div>
<p>
<asp:Label ID="Label2" runat="server" Text=""></asp:Label>
</p>
</form>
</body>
</html>
```

В данной форме присутствуют четыре компонента: метка Label1, содержащая текст «Введите свое имя»; поле ввода TextBox1 для ввода имени; кнопка Button1; метка Label2. Вначале она пуста, а в процессе работы в нее выводится сообщение. В параметрах кнопки указан обработчик нажатия Button1_Click. При нажатии на кнопку введенное в поле ввода имя внедряется в

сообщение, которое выводится в метке Label2.

Код обработки данной формы приведен далее.

```
using System;
using System.Collections.Generic; using System.Linq;
using System.Web; using System.Web.UI;
using System.Web.UI.WebControls;

public partial class Default2 : System.Web.UI.Page
{
protected void Page_Load(object sender, EventArgs e)
{
}

protected void Button1_Click(object sender, EventArgs e)
{
Label2.Text = "Привет, " + TextBox1.Text + "!"; }}
```

Таким образом осуществляется создание web-приложений на базе ASP.NET.

Список литературы

- 1 **Арлоу, Д.** UML и унифицированный процесс. Практический объектно-ориентированный анализ и проектирование / Д. Арлоу, А. Нейштадт. – Москва: Символ, 2016. – 624 с.
- 2 **Гробов, И. Д.** ASP.NET 2.0: теория и практика / И. Д. Гробов. – Москва: Диалог-МИФИ, 2007. – 608 с.
- 3 **Гуриков, С. Р.** Информатика : учебник / С. Р. Гуриков. – Москва: ФОРУМ; ИНФРА-М, 2018. – 463 с.
- 4 **Дадян, Э. Г.** Методы, модели, средства хранения и обработки данных: учебник / Э. Г. Дадян, Ю. А. Зеленков. – Москва: Вузовский учебник; ИНФРА-М, 2017. – 168 с.
- 5 **Курняван, Б.** Программирование WEB-приложений на языке Java / Б. Курняван. – Москва: Лори, 2014. – 880 с.
- 6 **Флэнаган, Д.** JavaScript: карманный справочник / Д. Флэнаган. – 3-е изд. – Москва : Вильямс, 2015. – 314 с.
- 7 **Шилдт, Г.** Java 8. Полное руководство / Г. Шилдт. – Москва : Вильямс, 2018. – 1375 с.