Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Артеменков Михаил Николаевич
Должность: Ректор
Дата подписания: 09.11.2026 18:40:57
Уникальный программный ключ:
a63b74166e8f4fc17bd1e0462725f79e393bfdf9

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Смоленский государственный университет»

Кафедра прикладной математики и информатики

«Утверждаю»
Проректор по учебно-
методической деятельности
_____ Ю.А. Устименко
«25» декабря 2025 г.

Рабочая программа дисциплины
Б1.В.02Взаимодействие в распределенных программных системах

Направление подготовки**01.04.02 Прикладная математика и информатика**
Направленность (профиль) **Математическое и программное обеспечение цифровых радиотехнических систем и комплексов**
Форма обучения очная
Курс – 1
Семестр – 2
Всего зачетных единиц –3,часов – 108
Форма отчетности: зачет – 2 семестр

Программу разработал: доктор технических наук, доцент В.И. Мунерман

Одобрена на заседании кафедры
«20» ноября 2025 г., протокол № 5

Заведующий кафедрой _____ С.В. Козлов

Смоленск
2025

1 Место дисциплины в структуре ОП

Дисциплина включена в часть, формируемую участниками образовательных отношений, образовательной программы по направлению подготовки01.04.02 Прикладная математика и информатика (направленность (профиль) – Математическое и программное обеспечение цифровых радиотехнических систем и комплексов).

Дисциплина «Взаимодействие в распределенных программных системах» находится в содержательной и методической взаимосвязи с такими дисциплинами, как «Параллельное программирование прикладных задач» и «Проектирование сетевых информационных систем».

К исходным требованиям, необходимым для изучения дисциплины «Взаимодействие в распределенных программных системах», относятся знания, умения и навыки предыдущего уровня образования, сформированные в процессе изучения языков и методов программирования, современных компьютерных технологий.

Освоение дисциплины «Взаимодействие в распределенных программных системах» является необходимой основой для последующего изучения таких дисциплин, как «Параллельное программирование прикладных задач», «Проектирование сетевых информационных систем». Знания, умения и навыки, полученные при изучении дисциплины, понадобятся при написании магистерской диссертации, а также в практической деятельности, связанной с проектированием автоматизированных информационных систем.

2 Планируемые результаты обучения по дисциплине

Компетенция	Индикаторы достижения
ПК-2. Способен проектировать программное и информационное обеспечение, вычислительные модели и модели данных для реализации элементов новых (или известных) программных продуктов в области цифровых радиотехнических систем и комплексов	**Знает:** структуру программного обеспечения, основные требования к его проектированию, методики разработки и анализа блок-схем алгоритмов, современные языки программирования, технологии программирования. **Умеет:** проектировать программное и информационное обеспечение в области цифровых радиотехнических систем и комплексов, формулировать требования к программным продуктам, создавать блок-схемы алгоритмов функционирования разрабатываемых программных продуктов, оценивать их вычислительную сложность. **Владеет:** методами и приемами формализации и алгоритмизации поставленных проектных задач по созданию программного обеспечения.
ПК-3. Способен разрабатывать программное обеспечение цифровых радиотехнических систем и комплексов	**Знает:** архитектуру аппаратной платформы, синтаксис, принципы и особенности программирования (в том числе кроссплатформенного, распределенного и параллельного программирования), стандартные библиотеки выбранного языка программирования, стандарты реализации интерфейсов устройств, технологии разработки и отладки программных продуктов, принципы информационного построения сетевого взаимодействия, методики тестирования программного обеспечения. **Умеет:** использовать методы и приемы формализации и алгоритмизации поставленных задач, применять выбранный язык программирования для написания программного кода, осуществлять отладку программного обеспечения. **Владеет:** основными методами и приемами разработки и отладки программных продуктов.

3 Содержание дисциплины

Методы и средства взаимодействия в программных системах. Понятие кроссплатформенности. Способы передачи данных между программами. Библиотеки программ.

Взаимодействие с MSOfficevMSVisualStudio. ОбъектыApplication, Document, WorkBook, WorkSheet, Range. Создание объектов-серверов для взаимодействия с MS Word и MS Excel.

Взаимодействие с базами данных в MS Visual Studio. Понятие провайдера СУБД. Основные провайдеры: ODBC, ADO. Основные объекты: набор данных, источник данных, соединение с базой данных. Методы отображения и изменения базы данных.

Библиотечные методы взаимодействия. Понятие библиотечной программы. Сборка и динамический вызов подпрограмм. Перекрытие. Динамически связываемые библиотеки.

Взаимодействие с низкоуровневыми системами программирования. Основные понятия языка Ассемблера. Связь Ассемблера с Object Pascal и MS Visual Studio C#, C++. Отладка программ с ассемблерными вставками.

4 Тематический план

№ п/п	Разделы и темы	Всего часов	Формы занятий		
			лекции	лабораторные занятия	самостоятельная работа
1	Методы и средства взаимодействия в программных системах	14	2	2	10
2	Взаимодействиес MS Office в MS Visual Studio (MS Word)	14	2	2	10
3	Взаимодействиес MS Office в MS Visual Studio (MS Excel)	14	2	2	10
4	Взаимодействие с базами данных в MS Visual Studio. Основные объекты: набор данных, источник данных, соединение с базой данных.	14	2	2	10
5	Взаимодействие с базами данных в MS Visual Studio. Методы отображения и изменения базы данных.	19	2	2	12
6	Библиотечные методы взаимодействия	14	2	2	10
7	Взаимодействие с низкоуровневыми системами программирования	19	4	4	14
	ИТОГО	108	16	16	76

5 Виды образовательной деятельности

Занятия лекционного типа

Лекция №1. Методы и средства взаимодействия в программных системах. Понятие кроссплатформенности. Способы передачи данных между программами. Библиотеки программ.

Лекция №2. Взаимодействие с MS Office в MS Visual Studio. Объекты Application, Document, WorkBook, WorkSheet, Range.

Лекция №3. Взаимодействие с MSOfficevMSVisualStudio. Создание объектов-серверов для взаимодействия с MS Word и MS Excel.

Лекция №4. Взаимодействие с базами данных в MS Visual Studio. Понятие провайдера СУБД. Основные провайдеры: ODBC, ADO. Основные объекты: набор данных, источник данных, соединение с базой данных.

Лекция №5. Взаимодействие с базами данных в MS Visual Studio. Методы отображения и изменения базы данных.

Лекция №6. Библиотечные методы взаимодействия. Понятие библиотечной программы. Сборка и динамический вызов подпрограмм. Перекрытие. Динамически связываемые библиотеки.

Лекция №7. Взаимодействие с низкоуровневыми системами программирования. Основные понятия языка Ассемблера. Связь Ассемблера с ObjectPascal и MSVisualStudioC#, C++.

Лекция №8. Взаимодействие с низкоуровневыми системами программирования. Отладка программ с ассемблерными вставками.

Занятия семинарского типа – лабораторные занятия

Задания к лабораторным работам размещены в ЭИОС СмолГУ https://cdo.smolgu.ru/local/crw/course.php?id=7696

Лабораторная работа №1.*Методы и средства взаимодействия в программных системах*

1. Понятие кроссплатформенности.
2. Способы передачи данных между программами. Библиотеки программ.
3. Библиотеки программ.
4. Виды библиотек.

Лабораторная работа №2,3.*Взаимодействие с MSOffice в MSVisualStudio*

1. Объект Application.
2. Объект Document.
3. Объект WorkBook.
4. Объект WorkSheet.
5. Объект Range.
6. Структура объекта-сервера связи.

Лабораторная работа №4.*Взаимодействие с базами данных в Object Pascal*

1. Объекты: набор данных, источник данных, соединение с базой данных.
2. Объект DataGridView.

Лабораторная работа №5.*Взаимодействие с базами данных в MSVisualStudio*

1. Объекты связи с базой данных.
2. Объекты чтения данных.
3. Технология BulkCopy.

Лабораторная работа №6.*Библиотечные методы взаимодействия*

1. Структура библиотечной программы.
2. Передача параметров и возврат значений.
3. Организация DLL.
4. Организация DLL в C#.

Лабораторная работа №7-8.*Взаимодействие с низкоуровневыми системами программирования*

1. Основные понятия языка Ассемблера.
2. Связь Ассемблера с и MSVisualStudioC#, C++.
3. Отладка программ с ассемблерными вставками.

Самостоятельная работа

Самостоятельная работа направлена на углубление и закрепление знаний студентов и развитии практических умений. Она заключается в работе с лекционными материалами, поиске и обзоре литературы и электронных источников, информации по заданным темам курса, в изучении тем, вынесенных на самостоятельную проработку и самостоятельном выполнении заданий.

Темы для самостоятельного изучения

1. Модели взаимодействия компонентов распределённых систем.
2. Механизмы удалённого вызова процедур (RPC) и обращения к объектам.
3. Веб-сервисы и сервисно-ориентированная архитектура (SOA).
4. Обеспечение целостности и согласованности данных.
5. Безопасность и аутентификация в распределённых системах.
6. Асинхронное взаимодействие и очереди сообщений.
7. Обнаружение и регистрация сервисов (Service Discovery).
8. Управление трафиком и балансировка нагрузки.
9. Мониторинг и трассировка распределённых систем.
10. Согласованность и репликация в гео-распределённых системах.
11. Управление состояниями и сеансами в распределённых системах
12. Обработка сбоев и обеспечение отказоустойчивости

Задания длясамостоятельной работы

Задание 1.Смоделировать распределённую транзакцию (например, перевод денег между счетами в разных банках) и описать шаги 2PC.

Задание 2.Зарегистрировать 2-3 микросервиса (например, на Spring Boot) и реализоватьавтоматическое обнаружение соседей;проверку здоровья сервисов.

Задание 3.Настроить Nginx или HAProxy как балансировщик для 3 инстансов сервиса.

Задание 4.Настроить репликацию MongoDB между 2 дата-центрами (например, локально через Docker Compose).

Задание 5.Разработать stateless-сервис (например, API для получения списка товаров) на Flask/FastAPI.

6 Критерии оценивания результатов освоения дисциплины (модуля)

6.1 Оценочные средства и критерии оценивания для текущей аттестации

1. Выполнение лабораторной работы

Образец задания для лабораторной работы

Разработать DLL в C++ для умножения двух матриц и соединить ее с программой C#.

№ п/п	Критерий	Балл
1.	Задание выполнено в срок (на соответствующем занятии)	1
2.	Задание выполнено студентом самостоятельно (без помощи и подсказок преподавателя)	1
3.	Результат строго соответствует заданию, служит прямым ответом на соответствующий вопрос задания	1
4.	Представленный результат проходит проверку (тестирование) на разных наборах входных данных	1
5.	Студент дает правильные пояснения о том, как выполнялось задание и каковы его результаты	1
6.	Работа оформлена в соответствии с требованиями (структура, форматирование, подписи, комментарии в коде и т. п.)	1
7.	Алгоритм/решение выстроено последовательно, блоки кода или расчётов логически разделены и обоснованы	1

8.	Использованы оптимальные методы и алгоритмы; решение не содержит избыточных или неоправданно сложных шагов	1
9.	В работе (особенно в коде или расчётах) присутствуют понятные пояснения, облегчающие восприятие	1
10.	Студент уверенно отвечает на дополнительные вопросы по заданию, демонстрирует понимание смежных аспектов задачи	1
	Итоговая (суммарная) оценка	Max - 10

Оценка «зачтено» - 5баллов и более; оценка «не зачтено» - менее 5 баллов.

2. Теоретические вопросы

Теоретические вопросы предлагаются студенту во время сдачи лабораторного задания и направлены на проверку знания студентом основных теоретических положений дисциплины.

Примеры теоретических вопросов:

1. Что такое распределённая система? Приведите 2–3 формальных определения (например, по Э. Таненбауму, Л. Лэмпорту).
2. В чём отличие распределённой системы от параллельной?
3. Каковы основные признаки (свойства) распределённой системы?
4. Почему распределённые системы *не подразумевают* обязательный параллелизм вычислений?
5. Что понимается под «узлом» распределённой системы?
6. Какова роль промежуточного программного обеспечения (middleware) в распределённых системах?
7. Опишите модель «клиент-сервер». В чём разница между двухзвенной и трёхзвенной архитектурой?
8. Приведите примеры сценариев, где оправдана P2P-архитектура (peer-to-peer).
9. Чем отличается синхронное взаимодействие от асинхронного? Назовите плюсы и минусы каждого.
10. Что такое «промежуточная среда» (middleware) и какие задачи она решает?
11. Как реализуется обмен данными между компонентами: через разделяемую память или сообщения? Почему в распределённых системах преобладает второй способ?
12. Что такое гетерогенная распределённая система? Чем она отличается от гомогенной?
13. Перечислите и кратко раскройте 5–6 ключевых требований к распределённым системам (открытость, масштабируемость, устойчивость и др.).
14. Что означает «открытость» системы? Почему важны общедоступные стандарты протоколов?
15. В чём суть масштабируемости? Как связана масштабируемость с балансировкой нагрузки?
16. Как обеспечивается поддержание логической целостности данных (атомарность операций)?
17. Что понимается под «устойчивостью» системы? Приведите пример уникальной точки сбоя и способы её устранения.
18. Почему безопасность — обязательное требование? Какие угрозы она призвана нейтрализовать?
19. Каковы базовые примитивы работы с очередями сообщений (на примере MQ-систем)?
20. В чём преимущество систем очередей сообщений над прямым обменом через сокеты?
21. Что такое удалённый вызов процедур (RPC)? На каком уровне он реализуется относительно TCP/IP?
22. Как соотносятся низкоуровневый обмен сообщениями (сокеты) и высокоуровневые протоколы (HTTP, RPC)?
23. Приведите примеры промежуточных сред для обмена сообщениями (не менее 3).
24. Почему в распределённых системах невозможно иметь единое точное время? Как это влияет на координацию процессов?

25. Что такое частичные отказы? Как система должна обрабатывать сбои отдельных узлов?

26. В чём сложность поддержания идентичности реплик при географическом распределении?

27. Почему полная прозрачность распределения ресурсов практически недостижима? Приведите пример компромисса между прозрачностью и производительностью.

28. Что означает «отсутствие общей совместно используемой памяти» в распределённой системе? Как это сказывается на программировании?

29. Что такое программная компонента в контексте распределённых систем? Каковы её границы выполнения?

30. Как компоненты обнаруживают друг друга в системе?

31. В чём разница между сервисом, предоставляемым компонентой, и её внутренним состоянием?

32. Почему важно, чтобы состояние процесса было закрыто для других процессов?

33. Как реализуется мобильность пользователя в распределённых системах?

34. Какие протоколы транспортного уровня чаще всего используются в распределённых системах (приведите 2–3 примера)?

35. В чём преимущества использования системных сокетов (unix sockets) или именованных каналов внутри одного узла?

36. Как промежуточная среда может скрывать различия между локальным и удалённым вызовом?

37. Приведите примеры открытых стандартов, обеспечивающих совместимость компонентов в распределённых системах.

38. Каковы типичные накладные расходы при использовании middleware?

39. Что такое распределённая разделяемая память (DSM)? В каких случаях она применяется?

40. В чём особенность архитектуры SIMD и почему она *не считается* распределённой системой?

41. Опишите паттерн «публикация-подписка» (pub/sub). Где он полезен?

42. Что такое балансировка нагрузки (load balancing) и как она влияет на масштабируемость?

43. Как реализуются механизмы аутентификации и авторизации в распределённых системах?

44. Почему задержка при обращении к удалённым ресурсам всегда больше, чем к локальным?

45. Как непредсказуемость времени доставки сообщений влияет на проектирование распределённых алгоритмов?

46. Что такое «время ответа» (response time) и «пропускная способность» (throughput) в контексте распределённых систем?

47. Как географическое распределение узлов влияет на латентность?

48. Как облачные платформы (AWS, GCP, Azure) решают проблемы распределённых систем (масштабируемость, отказоустойчивость)?

49. В чём особенности взаимодействия микросервисов по сравнению с монолитными архитектурами?

50. Как контейнеризация (Docker, Kubernetes) влияет на развёртывание и взаимодействие компонентов?

51. Какие новые протоколы и стандарты появляются для улучшения взаимодействия в распределённых системах (например, gRPC, GraphQL)?

Критерии оценки ответов на теоретические вопросы:

Показатель	Количество баллов
1) Полнота раскрытия темы (ответ охватывает все ключевые аспекты вопроса, не упускает существенные детали)	1

2) Точность терминологии (корректно использованы понятия и термины дисциплины; отсутствуют смысловые искажения)	1
3) Логичность и структура (ответ выстроен последовательно: есть введение, основная часть, выводы; мысли связаны между собой)	1
4) Глубина понимания (показано осмысление материала: не просто пересказ, а объяснение причинно-следственных связей, принципов, механизмов)	1
5) Аргументированность (положения подкреплены примерами, ссылками на источники, формулы, теоремы или экспериментальные данные)	1
6) Самостоятельность изложения (ответ не сводится к дословному воспроизведению учебника/лекции; присутствует индивидуальная интерпретация)	1
7) Актуальность и современность (упомянуты современные исследования, технологии или дискуссии по теме (если применимо))	1
8) Корректность примеров (приведённые иллюстрации точно соответствуют теории, не содержат ошибок и не вводят в заблуждение)	1
9) Ясность и доступность (текст понятен без дополнительных уточнений; сложные моменты разъяснены; нет избыточной перегруженности)	1
10) Грамотность и оформление (отсутствуют орфографические, пунктуационные и стилистические ошибки; формулы, схемы, цитаты оформлены по правилам)	1
Итоговая (суммарная) оценка	Max - 10

Оценка «зачтено» - 5баллов и более; оценка «не зачтено» - менее 5 баллов.

3. Выполнение заданий для самостоятельной работы (перечень заданий приведен в разделе 5 «Самостоятельная работа»).

Критерии оценивания выполнениязаданий:

№ п/п	Критерий	Балл
1.	Задание выполнено в срок (на соответствующем занятии)	1
2.	Задание выполнено студентом самостоятельно (без помощи и подсказок преподавателя)	1
3.	Результат строго соответствует заданию, служит прямым ответом на соответствующий вопрос задания	1
4.	Представленный результат проходит проверку (тестирование) на разных наборах входных данных	1
5.	Студент дает правильные пояснения о том, как выполнялось задание и каковы его результаты	1
6.	Работа оформлена в соответствии с требованиями (структура, форматирование, подписи, комментарии в коде и т. п.)	1
7.	Алгоритм/решение выстроено последовательно, блоки кода или расчётов логически разделены и обоснованы	1
8.	Использованы оптимальные методы и алгоритмы; решение не содержит избыточных или неоправданно сложных шагов	1
9.	В работе (особенно в коде или расчётах) присутствуют понятные пояснения, облегчающие восприятие	1
10.	Студент уверенно отвечает на дополнительные вопросы по заданию, демонстрирует понимание смежных аспектов задачи	1
	Итоговая (суммарная) оценка	Max - 10

Оценка «зачтено» - 5баллов и более; оценка «не зачтено» - менее 5 баллов.

6.2. Оценочные средства и критерии оценивания для промежуточной аттестации

Форма промежуточной аттестации – **зачет**

Для определения уровня сформированности компетенций применяется процедура независимой оценки. Независимая оценка уровня сформированности компетенций осуществляется комиссией (с возможностью привлечения внешних экспертов со стороны работодателей). В состав комиссии входят 3 преподавателя: заведующий кафедрой – председатель комиссии и независимый эксперт, преподаватель, ведущий лабораторные занятия по дисциплине «Взаимодействие в распределенных программных системах», преподаватель кафедры – независимый эксперт. Зачет проходит в форме просмотра портфолио.

Требования к структуре и содержанию портфолио

Портфолио студента включает в себя следующие виды деятельности и их оценки:

№ п/п	Вид деятельности	Количество заданий	Пороговая оценка	Максимальная оценка
1.	Выполнение и защита заданий лабораторных работ	8 лабораторных работ	40	80
2.	Ответы на теоретические вопросы	7 блоков вопросов	35	70
3.	Выполнение заданий для самостоятельной работы	5 заданий	25	50

Критерии и показатели оценивания портфолио по уровням

Уровень	Критерии и показатели оценивания
10 баллов	По результатам оценки портфолио набрано 93-100% от максимально возможного количества баллов, при этом по всем видам деятельности достигнуты пороговые значения
9 баллов	По результатам оценки портфолио набрано 86-92% от максимально возможного количества баллов, при этом по всем видам деятельности достигнуты пороговые значения
8 баллов	По результатам оценки портфолио набрано 79-85% от максимально возможного количества баллов, при этом по всем видам деятельности достигнуты пороговые значения
7 баллов	По результатам оценки портфолио набрано 71-78% от максимально возможного количества баллов, при этом по всем видам деятельности достигнуты пороговые значения
6 баллов	По результатам оценки портфолио набрано 64-70% от максимально возможного количества баллов, при этом по всем видам деятельности достигнуты пороговые значения
5 баллов	По результатам оценки портфолио набрано 57-63% от максимально возможного количества баллов, при этом по всем видам деятельности достигнуты пороговые значения
4 балла	По результатам оценки портфолио набрано 50-56% от максимально возможного количества баллов, при этом по всем видам деятельности достигнуты пороговые значения
3 балла	По результатам оценки портфолио набрано 36-49% от максимально возможного количества баллов и/или хотя бы по одному из видов деятельности не достигнуто пороговое значение
2 балла	По результатам оценки портфолио набрано 21-45% от максимально

	возможного количества баллов и/или хотя бы по одному из видов деятельности не достигнуто пороговое значение
1 балл	По результатам оценки портфолио набрано менее 21% от максимально возможного количества баллов и/или хотя бы по одному из видов деятельности не достигнуто пороговое значение

Таблица перевода оценок из 10-ти бальной системы в бинарную систему

Оценка по 10-ти бальной системе	Оценка по бинарной системе
10, 9, 8, 7,6,5,4	зачтено
3, 2, 1	не зачтено

7. Перечень основной и дополнительной учебной литературы

7.1. Основная литература

1. Лаврищева Е.М. Программная инженерия. Парадигмы, технологии и CASE-средства: учебник для вузов – М.: Издательство Юрайт, 2025. – 280 с. – URL: https://urait.ru/bcode/561899.
2. Маркин, А. В. Программирование на SQL: учебник и практикум для вузов – М.: Издательство Юрайт, 2025. – 805 с. – URL: https://urait.ru/bcode/568900.

7.2. Дополнительная литература

1. Корняков В.Н. Программирование документов и приложений MS Office в Delphi. – СПб.: БХВ-Петербург, 2005. – 496 с.
2. Автоматически реализуемые свойства (Руководство по программированию на C#). – http://msdn.microsoft.com/ru-ru/library/bb384054.aspx
3. Д. Л. Осипов Базы данных и Delphi. Теория и практика. – СПб, БХВ-Петербург, 2011. – 752 с.
4. Oliver Sturm Functional Programming in C#: Classic Programming Techniques for Modern Projects. – Wiley. 2011. – 290 с.
5. А.Шкрыль, Разработка клиент - серверных приложений в DELPHI. С-Пет., 2006.
6. Кривошеев С.Е.Разработка DLL в среде Borland Delphi (в помощь Delphi-разработчику). – http://citforum.ru/programming/digest/dlldelphi.shtml
7. Практическое руководство. Создание и использование библиотек DLL на языке C# (Руководство по программированию на C#). – http://msdn.microsoft.com/ru-ru/library/3707x96z(v=vs.90).aspx
8. Guido Gybels "Использование ассемблера в Дельфи". Электронная книга. – http://delphiexpert.ru/view_download.php?id=7
9. Доступ к COM серверам Microsoft Office. – http://www.delphisources.ru/pages/faq/base/access_to_microsoft_com_servers.html
10. Интеграция C# с Microsoft Office. – http://easyprog.ru/index.php?option=com_content&task=view&id=818&Itemid=54
11. Сорокин А.В. Delphi. Разработка базы данных. – СПб, Питер, 2005. – 474 с.
12. Работа с базами данных в Delphi с помощью ADO. – http://www.delphisources.ru/pages/articles/2006_year/art_27_11_06/art_27_11_06.html
13. Работа с базами данных на языке C#. Технология ADO.NET: Учебное пособие / сост. О. Н. Евсеева, А. Б. Шамшев. - Ульяновск: УлГТУ, 2009. - 170 с.
14. Zeeshan Amjad Using Unmanaged code and assembler in C#. – http://www.codeproject.com/Articles/1392/Using-Unmanaged-code-and-assembler-in-C

7.3. Перечень ресурсов информационно-телекоммуникационной сети «Интернет»

1. Автоматически реализуемые свойства (Руководство по программированию на C#). – http://msdn.microsoft.com/ru-ru/library/bb384054.aspx

2. Oliver Sturm Functional Programming in C#: Classic Programming Techniques for Modern Projects. – Wiley. 2011. – 290 c.

3. Практическое руководство. Создание и использование библиотек DLL на языке C# (Руководство по программированию на C#). – http://msdn.microsoft.com/ru-ru/library/3707x96z(v=vs.90).aspx

4. Guido Gybels "Использование ассемблера в Дельфи". Электронная книга. – http://delphiexpert.ru/view_download.php?id=7

5. Доступ к COM серверам Microsoft Office. – http://www.delphisources.ru/pages/faq/base/access_to_microsoft_com_servers.html

6. Интеграция C# с Microsoft Office. – http://easyprog.ru/index.php?option=com_content&task=view&id=818&Itemid=54

7. Работа с базами данных в Delphi с помощью ADO. – http://www.delphisources.ru/pages/articles/2006_year/art_27_11_06/art_27_11_06.html

8. Zeeshan Amjad Using Unmanaged code and assembler in C#. – http://www.codeproject.com/Articles/1392/Using-Unmanaged-code-and-assembler-in-C

8. Материально-техническое обеспечение

Учебная аудитория для проведения занятий лекционного типа, оснащенная стандартной учебной мебелью, интерактивной доской, мультимедиапроектором, ноутбуком и колонками.

Учебная аудитория для проведения занятий семинарского типа, групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации - компьютерная аудитория с выходом в Интернет.

Помещение для самостоятельной работы – компьютерный класс с доступом к сети «Интернет» и ЭИОС СмолГУ.

9. Программное обеспечение

KasperskyEndpointSecurity для бизнеса Стандартный АО «Лаборатория Касперского», лицензия 1FB6-161215-133553-1-6231.

Microsoft Open License, лицензия 49463448 всоставе: Microsoft Windows Professional 7 Russian; Microsoft Office 2010 Russian.

MicrosoftVisualStudio.