

Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Бадауров Насим Лидович
Должность: Врио ректора
Дата подписания: 28.03.2022 12:05:01
Уникальный программный ключ:
b261c06f25acbb0d1e6de5fc04abdfed0091d138

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
ФГБОУ ВО «ДАГЕСТАНСКИЙ ГОСУДАРСТВЕННЫЙ
ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

РЕКОМЕНДОВАНО

К УТВЕРЖЕНИЮ

Декан факультета

магистерской подготовки


Ашуралиева Р.К.
«02» 03 2022г.

УТВЕРЖДАЮ

Врио ректора ДГТУ,

Председатель методического
совета ДГТУ


Суракатов Н.С.
«03» 03 2022г.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

Дисциплина М1.В.ДВ.1 Объектно-ориентированное программирование
код и наименование дисциплины по ООП

для направления 09.04.01 «Информатика и вычислительная техника»
код и направление направления подготовки

по профилю Сети ЭВМ и телекоммуникации
наименование магистерской программы

факультет Магистерской подготовки
наименование факультета, где ведется дисциплина (практика)

кафедра Управление и информатика в технических системах и вычислительной техники
наименование кафедры, за которой закреплена дисциплина (практика)

Квалификация выпускника (степень) Магистр
бакалавр, магистр (специалист)

Форма обучения очная курс 1 семестр (ы) 1
очная, заочная, др

Всего трудоемкость в зачетных единицах (часах) 3 ЗЕТ (108)

лекции 9 экзамен -
час семестр


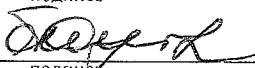
практические (семинарские) занятия 17 зачет 1
час семестр

лабораторные занятия 17 самостоятельная работа 65
час час

курсовой проект (работа, РГР) 1
семестр

И.о. зав. кафедрой

/Начальник УО


подпись

подпись

Асланов Т.Г.

Магомаева Э.В.

Программа составлена в соответствии с требованиями ФГОС ВО с учетом рекомендаций ООП ВО по направлению подготовки 09.04.01 «Информатика и вычислительная техника»

Программа одобрена на заседании выпускающей кафедры от «28» 02 2020 года, протокол № 6.

И.о. зав. кафедрой по данному направлению


_____ подпись

Асланов Т.Г.

ОДОБРЕНО

**Методической комиссией
по УГС(Н)**

09.00.00 – Информатика и
вычислительная техника

Председатель М.К.


_____ подпись Абдулгалимов А.М.

«28» 02 2020.

АВТОР ПРОГРАММЫ

К.т.н., ст. преп. Т.Г. Асланов


_____ подпись

1. Цели освоения дисциплины

Целью изучения дисциплины «Объектно-ориентированное программирование» является теоретическая и практическая подготовка студентов в области разработки программного обеспечения с использованием объектно-ориентированной модели современных языков программирования. Знания, полученные в результате освоения дисциплины, помогут при разработке системных программных компонентов современных информационных и расчетных программ, в проектировании и реализации системных компонентов операционных систем в такой степени, чтобы студенты могли самостоятельно выбирать средства реализации, находить необходимые программные и технологические решения для практически важных системных и предметно-ориентированных задач.

2. Место дисциплины в структуре ООП магистратуры

Дисциплина «Объектно-ориентированное программирование» представляет собой вариативную часть дисциплин по выбору учебного плана.

Дисциплина «Объектно-ориентированное программирование» является предшествующей для дисциплины «Сетевые протоколы».

3. Компетенции обучающегося, формируемые в результате освоения дисциплины «Объектно-ориентированное программирование»

Выпускник, освоивший программу магистратуры, должен обладать следующими компетенциями:

общекультурными (ОК):

– способностью совершенствовать и развивать свой интеллектуальный и общекультурный уровень (ОК-1);

– способностью понимать роль науки в развитии цивилизации, соотношение науки и техники, иметь представление о связанных с ними современных социальных и этических проблемах, понимать ценность научной рациональности и ее исторических типов (ОК-2);

общепрофессиональными (ОПК):

– владением, по крайней мере, одним из иностранных языков на уровне социального и профессионального общения, способностью применять специальную лексику и профессиональную терминологию языка (ОПК-4);

– владением методами и средствами получения, хранения, переработки и трансляции информации посредством современных компьютерных технологий, в том числе, в глобальных компьютерных сетях (ОПК-5);

– способностью анализировать профессиональную информацию, выделять в ней главное, структурировать, оформлять и представлять в виде аналитических обзоров с обоснованными выводами и рекомендациями (ОПК-6);

профессиональными компетенциями (ПК):

– владением существующими методами и алгоритмами решения задач распознавания и обработки данных (ПК-4);

– владением существующими методами и алгоритмами решения задач цифровой обработки сигналов (ПК-5);

– способностью к созданию программного обеспечения для анализа,

распознавания и обработки информации, систем цифровой обработки сигналов (ПК-15);

– способностью к разработке программного обеспечения для создания трехмерных изображений (ПК-18).

В результате освоения дисциплины обучающийся должен:

Знать: способы совершенствования и развития своего интеллектуального и общекультурного уровня; роль науки в развитии цивилизации; основы профессионального общения на иностранном языке; современные тенденции развития в области получения, передачи, хранения и переработки данных; научные основы применения компьютерных технологий; методы и алгоритмы решения задач распознавания и обработки данных; методы и алгоритмы решения задач цифровой обработки сигналов; методы создания программного обеспечения для анализа, распознавания и обработки информации, систем цифровой обработки сигналов; методы разработки программного обеспечения для создания трехмерных изображений.

Уметь: совершенствовать знания в выбранной области для решения профессиональных задач; анализировать социальные и этические проблемы науки и техники; формулировать основные положения технической документации на иностранном языке; применять вычислительную технику для решения практических задач; использовать типовые программные продукты, ориентированные на решение научных, проектных и технологических задач; решать задачи распознавания и обработки данных; решать задачи цифровой обработки сигналов; создавать программное обеспечение для анализа, распознавания и обработки информации, систем цифровой обработки сигналов; разрабатывать программное обеспечение для создания трехмерных изображений.

Владеть: технологиями анализа, синтеза и оценки общекультурных и профессиональных знаний; навыками рационального мышления; навыком применения одного из иностранных языков на уровне социального и профессионального общения; методами и средствами работы в глобальных компьютерных сетях; методиками сбора, переработки и представления научно-технических материалов по результатам исследований к опубликованию в печати, а также в виде обзоров, рефератов, отчетов, докладов и лекций; существующими методами и алгоритмами решения задач распознавания и обработки данных; существующими методами и алгоритмами решения задач цифровой обработки сигналов; навыками создания программного обеспечения для анализа, распознавания и обработки информации, систем цифровой обработки сигналов; навыками разработки программного обеспечения для создания трехмерных изображений.

4. Содержание программирование»

дисциплины

«Объектно-ориентированное

4.1. Содержание дисциплины

№	Раздел дисциплины Тема лекции и вопросы	Семестр	Неделя семестра	Виды учебной работы, включая самостоятельную работу студентов и трудоемкость (в часах)				Формы текущего контроля успеваемости (по срокам текущих аттестаций в семестре) Форма промежуточной аттестации (по семестрам)
				ЛК	ПЗ	ЛБ	СР	
а	б	в	г	д	е	ж	з	и
1	Лекция 1 ТЕМА: Парадигмы программирования. Основы объектно-ориентированного программирования. 1. Парадигма объектно-ориентированного программирования. 2. Понятия объектов и классов. 3. Конструктор и деструктор. 4. Модификаторы доступа.	1	1	2	4	4	13	Входная контрольная работа
2	Лекция 2 ТЕМА: Разработка классов, создание экземпляров класса. Указатели на классы. 1. Разработка собственных классов. 2. Инкапсуляция. 3. Абстракция. 4. Конструктор копий. 5. Переопределение операторов.		5	2	4	4	13	Аттестационная контрольная работа 1
3	Лекция 3 ТЕМА: Организация иерархии классов. Интерфейсы. Множественное наследование. 1. Наследование. 2. Виртуальные методы. 3. Полиморфизм. 4. Динамическое связывание. 5. Абстрактный класс.		9	2	4	4	13	Аттестационная контрольная работа 2
4	Лекция 4 ТЕМА: Создание и использование шаблонов классов. 1. Обобщенное программирование. 2. Шаблоны классов и особенности их применения.		13	2	4	4	13	Аттестационная контрольная работа 3
5	Лекция 5 ТЕМА: Использование стандартных библиотек классов при создании программ на языке высокого уровня. 1. Применение классов стандартной библиотеки для создания проектов, содержащих графический интерфейс пользователя.		17	1	1	1	13	

	2. Классы стандартной библиотеки, предназначенные для формирования дополнительных потоков и распределения вычислительной нагрузки между ними.							
Итого:			9	17	17	65	Зачет, курсовая работа	

4.2 Содержание практических занятий

№ п/п	№ по содержанию дисциплины	Наименование практического занятия	Количество часов	Рекомендуемая литература и методические разработки (№ источника из списка литературы)
1	1	Изучение парадигмы объектно-ориентированного программирования.	4	1-5
2	2	Изучение конструктора копий.	4	1-5
3	3	Изучение динамического связывания. Абстрактный класс.	4	1-5
4	4	Изучение шаблонов классов и особенности их применения.	2	1-5
5	5	Изучение классов стандартной библиотеки, предназначенных для формирования дополнительных потоков и распределения вычислительной нагрузки между ними.	3	1-5
Итого:			17	

4.3 Содержание лабораторных занятий

№ п/п	№ по содержанию дисциплины	Наименование лабораторного занятия	Количество часов	Рекомендуемая литература и методические разработки (№ источника из списка литературы)
1	1	Создание простых классов на объектно-ориентированном языке программирования. Разграничение доступа к элементам класса. Описание полей и методов класса.	4	1-5
2	2	Реализация методов класса. Переопределение методов. Перегрузка методов. Переопределение операторов. Дружественные функции. Разработка программы, содержащей класс для реализации прикладной математической задачи.	4	1-5

3	3	Создание классов-наследников. Выполнение проекта с организацией иерархии классов.	4	1-5
4	4	Создание программ, содержащих обобщенные классы. Применение обобщенных классов для организации хранения объектов произвольного типа.	2	1-5
5	5	Создание проектов с графическим интерфейсом пользователя для решения прикладных математических задач и их визуализации.	3	1-5
Итого:			17	

4.4 Тематика для самостоятельной работы студента

№ п/п	Тематика по содержанию дисциплины, выделенная для самостоятельного изучения	Количество часов из содержания дисциплины	Рекомендуемая литература и источники информации	Формы контроля СРС
1	Парадигма объектно-ориентированного программирования. Понятия объектов и классов. Конструктор и деструктор. Модификаторы доступа.	13	1-5	Опрос, курсовая
2	Разработка собственных классов. Инкапсуляция. Абстракция. Конструктор копий. Переопределение операторов.	13	1-5	Опрос, курсовая
3	Наследование. Виртуальные методы. Полиморфизм. Динамическое связывание. Абстрактный класс.	13	1-5	Опрос, курсовая
4	Обобщенное программирование. Шаблоны классов и особенности их применения.	13	1-5	Опрос, курсовая
5	Применение классов стандартной библиотеки для создания проектов, содержащих графический интерфейс пользователя. Классы стандартной библиотеки, предназначенные для формирования дополнительных потоков и распределения вычислительной нагрузки между ними.	13	1-5	Опрос, курсовая

5. Образовательные технологии

В ходе проведения занятий используются такие методы обучения как презентация, применение компьютерной техники.

6. Оценочные средства для текущего контроля успеваемости, промежуточной аттестации по итогам освоения дисциплины и учебно-методическое обеспечение самостоятельной работы студентов

6.1 Перечень вопросов по проверке входных знаний студентов

1. Парадигма объектно-ориентированного программирования.
2. Понятия объектов и классов.
3. Конструктор и деструктор.
4. Модификаторы доступа.
5. Разработка собственных классов.
6. Инкапсуляция.
7. Абстракция.
8. Конструктор копий.
9. Переопределение операторов.
10. Наследование.
11. Виртуальные методы.
12. Полиморфизм.
13. Динамическое связывание.
14. Абстрактный класс.
15. Обобщенное программирование.
16. Шаблоны классов и особенности их применения.
17. Применение классов стандартной библиотеки для создания проектов, содержащих графический интерфейс пользователя.
18. Классы стандартной библиотеки, предназначенные для формирования дополнительных потоков и распределения вычислительной нагрузки между ними.

6.2. Задания для текущих аттестаций

6.2.1. Контрольные вопросы для первой аттестации

1. В чем состоят недостатки процедурного и модульного программирования?
2. Что такое объектно-ориентированное программирование?
3. Каковы преимущества и цели объектно-ориентированного программирования?
4. Дайте определения следующих терминов: класс, объект, поведение.
5. Каким образом объекты обмениваются информацией?
6. Что такое конструктор?
7. Какие подходы к программированию Вам известны?
8. Каковы основные понятия объектно-ориентированного подхода к

программированию?

9. Что такое класс?
10. Что такое объект?
11. Что такое атрибут класса?
12. Что такое метод?
13. Что отличает объектно-ориентированный подход к программированию от компонентно-ориентированного?
14. В чем Вы видите преимущества языков объектно-ориентированного программирования?
15. В чем состоят особенности объектно-ориентированного проектирования?
16. Почему при объектно-ориентированном проектировании необходимо использовать инструментальные средства?
17. В чем Вы видите преимущества инструментального средства Rational Rose для объектно-ориентированного проектирования?
18. В чем состоят особенности языков программирования на основе функционального подхода?
19. В чем состоят особенности языков объектно-ориентированного программирования?
20. В чем состоят особенности языков логического программирования?
21. Какие языки программирования основаны на структурном подходе?
22. Какие цели объектно-ориентированного программирования реализует абстракция?
23. Дайте определение понятию «абстракция» в программировании.
24. Какие математические теории, формализующие понятие абстракции, Вам известны?

6.2.2. Контрольные вопросы для второй аттестации

1. Приведите пример применения абстракции.
2. Почему для достижения эффективной абстракции важно четко определить степень детализации предметной области?
3. В чем состоит особенность абстрактных типов данных?
4. В чем Вы видите недостатки абстракции?
5. Возможно ли применение в объектно-ориентированной программе различных степеней абстракции? Приведите пример.
6. Какие цели объектно-ориентированного программирования реализует инкапсуляция?
7. Дайте определение понятию «инкапсуляция»
8. Приведите пример применения инкапсуляции.
9. Дайте определение понятию «реализация».
10. Дайте определение понятию «интерфейс».
11. В чем состоит различие между интерфейсом и реализацией?
12. Почему для достижения эффективной инкапсуляции важно четко распределить ответственность?
13. Что такое тип?
14. В чем состоит особенность инкапсулированных типов данных?

15. В чем Вы видите недостатки инкапсуляции?
16. Что такое наследование?
17. Что такое единичное (простое) наследование?
18. Что такое множественное наследование?
19. Что такое базовый класс?
20. Что такое производный класс?
21. В чем состоят ограничения на использование множественного наследования?
22. Какие элементы класса может наследовать другой класс?
23. В чем основное различие между базовым классом и производным классом?
24. В чем состоит проблема «хрупких» базовых классов?

6.2.3. Контрольные вопросы для третьей аттестации

1. Что такое полиморфизм?
2. Какими средствами реализуется полиморфизм?
3. Что такое отложенное (позднее) связывание?
4. Что такое раннее связывание?
5. Каковы преимущества и недостатки применения полиморфизма времени выполнения?
6. Каковы преимущества и недостатки применения полиморфизма времени компиляции?
7. Что такое параметрический полиморфизм?
8. Что такое переопределение метода? В каких случаях оно применяется?
9. Что такое перегрузка метода?
10. Что такое «ленивые» вычисления?
11. В чем преимущества и недостатки «ленивых» вычислений?
12. Какие трудности проведения объектно-ориентированного анализа Вам известны?
13. Что такое технология разработки программного обеспечения?
14. Что такое итеративная технология проектирования?
15. Как итеративная технология проектирования соотносится со спиральной моделью?
16. В чем состоит результат объектно-ориентированного анализа?
17. Что описывают требования к системе?
18. В чем важность повторного использования?
19. Какие элементы проекта могут быть повторно использованы?
20. Что такое шаблон проектирования?
21. Какие элементы входят в шаблон проектирования?
22. Как правильно выбрать шаблон проектирования?
23. Гарантирует ли использование шаблонов высокую долю повторного использования?
24. В чем состоит важность дисциплины проекта для повышения доли повторного использования?
25. Как связан выбор модели жизненного цикла с повторным использованием?

6.3. Перечень вопросов по проверке остаточных знаний

1. В чем состоят недостатки процедурного и модульного программирования?
2. Что такое объектно-ориентированное программирование?
3. Каковы преимущества и цели объектно-ориентированного программирования?
4. Дайте определения следующих терминов: класс, объект, поведение.
5. Каким образом объекты обмениваются информацией?
6. Что такое конструктор?
7. Какие подходы к программированию Вам известны?
8. Каковы основные понятия объектно-ориентированного подхода к программированию?
9. Что такое класс?
10. Что такое объект?
11. Что такое атрибут класса?
12. Что такое метод?
13. Что отличает объектно-ориентированный подход к программированию от компонентно-ориентированного?
14. В чем Вы видите преимущества языков объектно-ориентированного программирования?
15. В чем состоят особенности объектно-ориентированного проектирования?
16. Почему при объектно-ориентированном проектировании необходимо использовать инструментальные средства?
17. В чем Вы видите преимущества инструментального средства Rational Rose для объектно-ориентированного проектирования?
18. В чем состоят особенности языков программирования на основе функционального подхода?
19. В чем состоят особенности языков объектно-ориентированного программирования?
20. В чем состоят особенности языков логического программирования?
21. Какие языки программирования основаны на структурном подходе?
22. Какие цели объектно-ориентированного программирования реализует абстракция?
23. Дайте определение понятию «абстракция» в программировании.
24. Какие математические теории, формализующие понятие абстракции, Вам известны?
25. Приведите пример применения абстракции.
26. Почему для достижения эффективной абстракции важно четко определить степень детализации предметной области?
27. В чем состоит особенность абстрактных типов данных?
28. В чем Вы видите недостатки абстракции?
29. Возможно ли применение в объектно-ориентированной программе различных степеней абстракции? Приведите пример.
30. Какие цели объектно-ориентированного программирования реализует инкапсуляция?
31. Дайте определение понятию «инкапсуляция»

32. Приведите пример применения инкапсуляции.
33. Дайте определение понятию «реализация».
34. Дайте определение понятию «интерфейс».
35. В чем состоит различие между интерфейсом и реализацией?
36. Почему для достижения эффективной инкапсуляции важно четко распределить ответственность?
37. Что такое тип?
38. В чем состоит особенность инкапсулированных типов данных?
39. В чем Вы видите недостатки инкапсуляции?
40. Что такое наследование?
41. Что такое единичное (простое) наследование?
42. Что такое множественное наследование?
43. Что такое базовый класс?
44. Что такое производный класс?
45. В чем состоят ограничения на использование множественного наследования?
46. Какие элементы класса может наследовать другой класс?
47. В чем основное различие между базовым классом и производным классом?
48. В чем состоит проблема «хрупких» базовых классов?
49. Что такое полиморфизм?
50. Какими средствами реализуется полиморфизм?
51. Что такое отложенное (позднее) связывание?
52. Что такое раннее связывание?
53. Каковы преимущества и недостатки применения полиморфизма времени выполнения?
54. Каковы преимущества и недостатки применения полиморфизма времени компиляции?
55. Что такое параметрический полиморфизм?
56. Что такое переопределение метода? В каких случаях оно применяется?
57. Что такое перегрузка метода?
58. Что такое «ленивые» вычисления?
59. В чем преимущества и недостатки «ленивых» вычислений?
60. Какие трудности проведения объектно-ориентированного анализа Вам известны?
61. Что такое технология разработки программного обеспечения?
62. Что такое итеративная технология проектирования?
63. Как итеративная технология проектирования соотносится со спиральной моделью?
64. В чем состоит результат объектно-ориентированного анализа?
65. Что описывают требования к системе?
66. В чем важность повторного использования?
67. Какие элементы проекта могут быть повторно использованы?
68. Что такое шаблон проектирования?
69. Какие элементы входят в шаблон проектирования?
70. Как правильно выбрать шаблон проектирования?

71. Гарантирует ли использование шаблонов высокую долю повторного использования?

72. В чем состоит важность дисциплины проекта для повышения доли повторного использования?

73. Как связан выбор модели жизненного цикла с повторным использованием?

6.4. Задания для промежуточной аттестации

6.4.1. Контрольные вопросы для проведения зачета

1. Парадигма объектно-ориентированного программирования.
2. Понятия объектов и классов.
3. Конструктор и деструктор.
4. Модификаторы доступа.
5. Разработка собственных классов.
6. Инкапсуляция.
7. Абстракция.
8. Конструктор копий.
9. Переопределение операторов.
10. Наследование.
11. Виртуальные методы.
12. Полиморфизм.
13. Динамическое связывание.
14. Абстрактный класс.
15. Обобщенное программирование.
16. Шаблоны классов и особенности их применения.
17. Применение классов стандартной библиотеки для создания проектов, содержащих графический интерфейс пользователя.
18. Классы стандартной библиотеки, предназначенные для формирования дополнительных потоков и распределения вычислительной нагрузки между ними.

6.5. Темы курсовых работ

1. Библиотека классов, реализующих графические примитивы на плоскости с возможностью аффинных преобразований.
2. Библиотека классов, реализующих графические примитивы на плоскости с реализацией операций над множествами.
3. Библиотека классов, реализующих графические примитивы в трёхмерном пространстве с возможностью аффинных преобразований.
4. Классы для описания оконного графического интерфейса, аналогичного интерфейсу MS Windows. Должны быть реализованы обработчики событий с использованием виртуальных функций.
5. Библиотека классов, реализующих комплексную арифметику.
6. Библиотека классов, реализующих векторы в n-мерном пространстве.
7. Реализация строк и операций над ними, включая работу с регулярными выражениями.
8. «Обобщённый массив» (позволяющий хранить данные произвольных типов).

9. Ассоциативный массив (хэш-массив) (с возможностью хранения данных произвольных типов).
10. Реализация различных типов графов и операций над ними.
11. Система классов для обеспечения работы с абонентами телефонной компании.
12. Система классов для обеспечения работы деканата.
13. Система классов, описывающих сотрудников предприятия/организации с их функциями (сотрудник, менеджер, ...).
14. Система классов, описывающих различные транспортные средства.
15. Моделирование замкнутой биологической системы (корм, травоядное, хищник).
16. Моделирование муравейника (несколько типов муравьёв, источники питания, внешние раздражители,...).
17. Моделирование дорожного движения на заданной карте дорог.
18. Моделирование компьютерной сети (стационарной).
19. Моделирование компьютерной сети с изменяющейся топологией.
20. Моделирование планетарной системы.
21. Система классов для реализации матричной арифметики.
22. Библиотека классов для реализации игрового приложения.

7. Учебно-методическое и информационное обеспечение дисциплины (модуля)

Рекомендуемая литература и источники информации (основная и дополнительная)

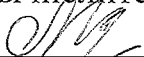
Зав. библиотекой

№ п/п	Необходимая учебная, учебно-методическая (основная и дополнительная) литература, программное обеспечение и Интернет ресурсы	Автор(ы)	Издательство и год издания	Количество изданий	
				В библиотеке	На кафедре
ОСНОВНАЯ					
1	Практикум по объектно-ориентированному программированию	Бабушкина И.А.	Москва : Лаборатория знаний, 2020	IPR BOOKS iprbookshop.ru/ 12254.html	
2	Объектно-ориентированное программирование : конспект лекций	Лисицин Д.В.	Новосибирск : Новосибирский государственный технический университет, 2010	IPR BOOKS iprbookshop.ru/ 44970.html	
ДОПОЛНИТЕЛЬНАЯ					
3	Объектно-ориентированное	Николаев Е.И.	Ставрополь :	IPR BOOKS	

	программирование. Часть 1 : лабораторный практикум		Северо-Кавказский федеральный университет, 2015	iprbookshop.ru/62966.html	
4	Объектно-ориентированное программирование : учебное пособие	Николаев Е.И.	Ставрополь : Северо-Кавказский федеральный университет, 2015	IPR BOOKS iprbookshop.ru/62967.html	
5	Объектно-ориентированное программирование: учебное пособие. Курс лекций	Сорокин А.А.	Ставрополь : Северо-Кавказский федеральный университет, 2014	IPR BOOKS iprbookshop.ru/63110.html	

8. Материально-техническое обеспечение дисциплины (модуля)

Семинарские занятия по дисциплине проводятся в аудитории с презентационной техникой и учебной мебелью.

Программа составлена в соответствии с требованиями ФГОС ВО с учетом рекомендаций ООП ВО по направлению и профилю подготовки 09.04.01 – Информатика и вычислительная техника. Рецензент от выпускающей кафедры по направлению  Меркухин Е.Н.