Минобрнауки России

Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Оренбургский государственный университет имени В.А. Бондаренко»

Кафедра программного обеспечения вычислительной техники и автоматизированных систем

РАБОЧАЯ ПРОГРАММА

ДИСЦИПЛИНЫ

«Б1.Д.Б.11 Основы программирования»

Уровень высшего образования

БАКАЛАВРИАТ

Направление подготовки

02.03.02 Фундаментальная информатика и информационные технологии
(код и наименование направления подготовки)

Разработка корпоративных систем и цифровых решений
(наименование направленности (профиля) образовательной программы)

Квалификация

Бакалавр

Форма обучения

Очная

Год набора 2026

Рабочая программа дисциплины «*Б1.Д.Б.11 Основы программирования*» рассмотрена и утверждена на заседании кафедры

Кафедра программного обеспечения вычислительной техники и автоматизированных систем
наименование кафедры

протокол № _____ от "___" _____ 20__ г.

Заведующий кафедрой
Кафедра программного обеспечения вычислительной техники и автоматизированных систем
наименование кафедры *подпись* Д.В. Горбачев *расшифровка подписи*

Исполнители:

доцент *подпись* Л.Ф. Тагирова
должность *подпись* *расшифровка подписи*

доцент *подпись* Е.Н. Чернопрудова
должность *подпись* *расшифровка подписи*

СОГЛАСОВАНО:
Председатель методической комиссии по направлению подготовки
02.03.02 Фундаментальная информатика и информационные технологии
код наименование *личная подпись* *расшифровка подписи*

Заведующий отделом формирования фонда и научной обработки документов
личная подпись С.А. Бикчентаева *расшифровка подписи*

Уполномоченный по качеству института
личная подпись С.Н. Морозова *расшифровка подписи*

№ регистрации _____

© Тагирова Л.Ф., Чернопрудова Е.Н., 2026
© ОГУ, 2026

1 Цели и задачи освоения дисциплины

Целью освоения дисциплины является формирование у студентов компетенций в области алгоритмизации, программирования на языках высокого уровня, разработки, отладки и тестирования программных продуктов с использованием современных сред разработки.

Задачи:

Задачами дисциплины являются освоение методов алгоритмизации и стандартов разработки схем алгоритмов, освоение синтаксиса и семантики языков программирования высокого уровня и технологий программирования, освоение стандартных библиотек при разработке программ, освоение работы в современных средах разработки, а также освоение тестирования программ и создания прототипов программных продуктов.

2 Место дисциплины в структуре образовательной программы

Дисциплина относится к базовой части блока Д «Дисциплины (модули)»

Пререквизиты дисциплины: *Б1.Д.Б.10 Информатика*

Постреквизиты дисциплины: *Б1.Д.Б.15 Математическое моделирование, Б1.Д.Б.16 Методы оптимизации, Б1.Д.Б.17 Операционные системы, Б1.Д.Б.18 Вычислительные системы, сети и телекоммуникации, Б1.Д.Б.19 Системное администрирование, Б1.Д.Б.20 Информационная безопасность, Б1.Д.Б.21 Основы проектной деятельности, Б1.Д.Б.22 Дискретная математика, Б1.Д.Б.23 Структуры данных и алгоритмы, Б1.Д.Б.24 Системы управления базами данных, Б1.Д.Б.25 Тестирование цифровых решений, Б1.Д.В.1 Проектная деятельность, Б1.Д.В.2 Технологии программирования корпоративных систем, Б1.Д.В.4 Искусственный интеллект и машинное обучение, Б1.Д.В.Э.2.2 Низкоуровневое программирование, ФДТ.1 WEB-дизайн, ФДТ.4 Методология кибериммунитета*

3 Требования к результатам обучения по дисциплине

Процесс изучения дисциплины направлен на формирование следующих результатов обучения

Код и наименование формируемых компетенций	Код и наименование индикатора достижения компетенции	Планируемые результаты обучения по дисциплине, характеризующие этапы формирования компетенций
ОПК-3 Способен к разработке алгоритмических и программных решений в области системного и прикладного программирования, математических, информационных и имитационных моделей, созданию информационных ресурсов глобальных сетей, образовательного контента, прикладных баз данных, тестов и средств	ОПК-3-В-1 Разрабатывает алгоритмические и программные решения, тесты и средства тестирования, информационные ресурсы, прикладные базы данных	**Знать:** основные парадигмы программирования; состав и функции систем программирования; способы записи алгоритмов; алфавит, синтаксис и семантику языков программирования C++ и Python; основные операторы и типы данных языков программирования.

Код и наименование формируемых компетенций	Код и наименование индикатора достижения компетенции	Планируемые результаты обучения по дисциплине, характеризующие этапы формирования компетенций
тестирования систем и средств на соответствие стандартам и исходным требованиям		**Уметь:** разрабатывать алгоритмы решения поставленных задач; разрабатывать программы с использованием среды программирования с текстовым и графическим интерфейсом; реализовывать принципы объектно-ориентированного программирования.
		Владеть: навыками использования возможностей среды программирования при написании прикладных программ на языках C++ и Python.
ПК*-2 Способен разрабатывать и отлаживать программный код	ПК*-2-В-1 Выполняет формализацию и алгоритмизацию поставленных задач для разработки программного кода ПК*-2-В-4 Разрабатывает программный код с использованием языков программирования ПК*-2-В-5 Оформляет программный код в соответствии с установленными требованиями ПК*-2-В-7 Проверяет и отлаживает программный код	**Знать:** - алгоритмы решения типичных задач, области и способы их применения, нотации и программное обеспечение для графического отображения алгоритмов, методы и приемы алгоритмизации поставленных задач. - синтаксис выбранного языка программирования, особенности программирования на этом языке, стандартные библиотеки языка программирования, методологии разработки компьютерного программного обеспечения, технологии программирования. - нормативно-технические документы, определяющие требования к оформлению программного кода, основные стандарты оформления технической документации на компьютерное программное обеспечение. - методы и приемы отлад-

Код и наименование формируемых компетенций	Код и наименование индикатора достижения компетенции	Планируемые результаты обучения по дисциплине, характеризующие этапы формирования компетенций
		ки программного кода, типы и форматы сообщений об ошибках, предупреждений. **Уметь:** - использовать методы и приемы формализации и алгоритмизации поставленных задач, применять алгоритмы решения типовых задач в соответствующих областях. - применять выбранные языки программирования для написания программного кода, использовать выбранную среду программирования, использовать возможности имеющейся технической и/или программной архитектуры для написания программного кода. - применять заданные стандарты и шаблоны для составления и оформления технической документации, нормативно-технические документы, определяющие требования к оформлению программного кода, инструментарий для создания и актуализации исходных текстов программ. - выявлять ошибки в программном коде, отлаживать программный код на уровне программных модулей и межмодульных взаимодействий и взаимодействий с окружением. **Владеть:** - навыками использования методов и приемов формализации и алгоритмизации поставленных задач, - навыками применения выбранных языков про-

Код и наименование формируемых компетенций	Код и наименование индикатора достижения компетенции	Планируемые результаты обучения по дисциплине, характеризующие этапы формирования компетенций
		граммирования для написания программного кода, использовать выбранную среду программирования, - навыками применения заданных стандартов и шаблонов для оформления технической документации, - навыками выявления ошибок в программном коде, отладки программного кода.

4 Структура и содержание дисциплины

4.1 Структура дисциплины

Общая трудоемкость дисциплины составляет 10 зачетных единиц (360 академических часов).

Вид работы	Трудоемкость, академических часов		
	1 семестр	2 семестр	всего
Общая трудоёмкость	**216**	**144**	**360**
Контактная работа:	**69,25**	**86,5**	**155,75**
Лекции (Л)	34	34	68
Практические занятия (ПЗ)		16	16
Лабораторные работы (ЛР)	34	34	68
Консультации	1	1	2
Индивидуальная работа и инновационные формы учебных занятий		1	1
Промежуточная аттестация (зачет, экзамен)	0,25	0,5	0,75
Самостоятельная работа: *- выполнение курсовой работы (КР);* *- выполнение расчетно-графического задания (РГЗ);* *- самоподготовка (проработка и повторение лекционного материала и материала учебников и учебных пособий;* *- изучение разделов курса в системе электронного обучения;* *- подготовка к лабораторным занятиям;* *- подготовка к практическим занятиям;* *- подготовка к рубежному контролю.*	**146,75** +	**57,5** +	**204,25**
Вид итогового контроля (зачет, экзамен, дифференцированный зачет)	**экзамен**	**экзамен**	

Разделы дисциплины, изучаемые в 1 семестре

№	Наименование разделов	Количество часов

раздела		всего	аудиторная работа			внеауд. работа
			Л	ПЗ	ЛР	
1	Основы алгоритмизации и программирования. Классификация алгоритмов	18	4			14
2	Основы программирования на языке С++. Базовые конструкции языка С++: операторы и функции	22	4		4	14
3	Обработка массивов в С++	20	2		4	14
4	Работа со строками в С++	20	2		4	14
5	Динамическая память и указатели в С++	24	4		4	16
6	Пользовательские функции в С++	24	4		4	16
7	Обработка данных в файлах средствами С++	24	4		4	16
8	Визуальное программирование на С++	26	4		6	16
9	Составные типы данных и структуры в С++	22	4		4	14
10	Стандартные библиотеки С++	16	2			14
	Итого:	216	34		34	148

Разделы дисциплины, изучаемые в 2 семестре

№ раздела	Наименование разделов	Количество часов				
		всего	аудиторная работа			внеауд. работа
			Л	ПЗ	ЛР	
11	Основы программирования на языке Python. Базовые конструкции языка Python: операторы и функции	26	8	6	4	8
12	Пользовательские функции в Python	22	6	2	6	8
13	Работа со строками на языке Python	20	6	-	6	8
14	Структуры данных языка Python	24	6	2	6	10
15	Файлы и обработка данных в Python	31	6	4	8	13
16	Разработка приложений с графическим пользовательским интерфейсом.	21	2	2	4	13
	Итого:	144	34	16	34	60
	Всего:	360	68	16	68	208

4.2 Содержание разделов дисциплины

1. **Основы алгоритмизации и программирования.** Понятие технологии программирования; этапы создания программы; структурное программирование; процедурное программирование; модульное программирование; процедуры и функции; объектно-ориентированное программирование; визуальное программирование; событийно-ориентированное программирование. Классификация алгоритмов. Классификация алгоритмов. Основные понятия алгоритмизации. Понятие схем-алгоритма. Алгоритмы решения задач. Разработка разветвляющихся структур. Разработка циклических структур.

2. **Основы программирования на языке С++.** Базовые конструкции языка С++: операторы и функции. Состав языка С++. Алфавит языка. Идентификаторы. Ключевые слова. Знаки операций. Константы. Комментарии. Стандартные типы данных языка С++. Переменные и выражения. Переменные. Выражения. Структура программы. Программная реализация. Операторы. Математические операторы. Побитовые операторы. Операторы присваивания. Логические операторы. Операторы сравнения. Оператор ветвления if. Оператор switch. Цикл с параметром for. Оператор goto. Оператор break. Операторы continue и return. Стандартные функции С++.

3. **Обработка массивов в C++.** Массивы. Одномерные массивы. Объявление и инициализация одномерных массивов. Ввод и вывод элементов массива. Доступ к элементам массива. Основные операции над одномерными массивами: поиск, суммирование, нахождение минимального и максимального элементов, сортировка, реверс. Многомерные массивы. Двумерные массивы. Объявление и инициализация двумерных массивов. Обход двумерных массивов.

4. **Работа со строками в C++.** Строки в стиле C (char*). Строки в стиле C++ (std::string). Объявление и инициализация строк. Ввод и вывод строк. Основные операции над строками: копирование, сравнение, конкатенация, определение длины, обработка символов, разбиение на подстроки, поиск и замена, преобразование строк в числа и чисел в строки. Массивы строк. Передача строк в функции. Возврат строк из функций. Форматирование строк.

5. **Динамическая память и указатели в C++.** Выделение и освобождение динамической памяти. Операторы new и delete. Динамические массивы. Указатели и динамическая память. Утечки памяти и способы их предотвращения.

6. **Пользовательские функции в C++.** Объявление и определение функций. Возвращаемое значение. Параметры функции. Передача массивов в качестве параметров. Передача имен функций в качестве параметров. Рекурсивные функции. Перегрузка функции. Шаблоны функций.

7. **Обработка данных в файлах средствами C++.** Файлы. Текстовые и бинарные файлы. Открытие и закрытие файлов. Режимы работы с файлами. Файлы и потоки. Соединение и отсоединение потока от файла. Перенаправление потока. Работа с индикаторами ошибки, позиции и конца файла. Блочный ввод-вывод. Символьный ввод-вывод. Работа с буферами. Стандартные потоки. Служебные функции для работы с файлами.

8. **Визуальное программирование на C++.** Использование библиотеки визуальных компонентов. Главная форма программы. Основные понятия и термины при разработке Windows приложений. Создание простейшего Windows-приложения с использованием Win API. Создание окна на основе класса окна. Меню. Окна сообщений. Элементы управления и диалоги. Классы MFC для элементов управления. Элементы управления Windows. Списки. Поле ввода. Контрольные переключатели. Полосы прокрутки. Графический вывод. Введение в Qt Creator. Структура проекта Qt. Сигналы и слоты. Основные виджеты Qt. Создание окон и диалогов. Обработка событий в Qt.

9. **Составные типы данных и структуры в C++.** Типы данных, определяемые пользователем. Переименование типов (typedef). Перечисления (enum). Структуры (struct). Битовые поля. Объединения (union). Динамические структуры данных. Линейные списки. Стеки. Очереди. Бинарные деревья. Реализация динамических структур с помощью массивов.

10. **Стандартные библиотеки C++.** Состав и назначение стандартной библиотеки C++. Библиотека ввода-вывода (iostream). Библиотека работы со строками (string). Библиотека контейнеров (vector, list, map, set). Библиотека алгоритмов (algorithm). Библиотека математических функций (cmath). Библиотека работы с памятью (memory). Библиотека работы со временем и датой (ctime). Использование пространства имён std. Подключение и применение стандартных библиотек в программах.

11. **Основы программирования на языке Python. Базовые конструкции языка Python: операторы и функции.** Язык Python: история, область применения (автоматизация, анализ и визуализация данных, машинное обучение, веб-разработка), сопоставление с C++: интерпретация и компиляция, динамическая и статическая типизация, управление памятью. Интерпретатор, интерактивный режим, запуск сценариев; среды разработки PyCharm и VS Code; виртуальные окружения и установка пакетов. Синтаксис: отступы, идентификаторы, комментарии; соглашение о стиле PEP 8. Объекты и переменные-ссылки, изменяемые и неизменяемые типы. Числовые типы, логический тип, None; операции и выражения; преобразование типов. Ввод и вывод: input, print, f-строки. Управляющие конструкции: if–elif–else, циклы while и for, функция range, break, continue.

12. **Пользовательские функции в Python.** Функции: определение, позиционные и именованные аргументы, значения по умолчанию, произвольное число аргументов, возврат нескольких значений, области видимости, лямбда-выражения, документирующие строки.

13. **Работа со строками в Python.** Строки: индексация и срезы, методы строк, форматирование, неизменяемость.

14. **Структуры данных языка Python.** Встроенные структуры данных: списки, кортежи, словари, множества – создание, основные методы, особенности, вычислительная сложность операций и области применения; вложенные структуры. Срезы, распаковка, генераторы списков, словарей и

множеств; итераторы и генераторы; функции enumerate, zip, sorted. Копирование объектов: присваивание, поверхностное и глубокое копирование.

15. **Файлы и обработка данных в Python.** Файловый ввод-вывод: функция open, режимы, кодировки, менеджер контекста with; построчная обработка текстовых файлов. Форматы обмена данными: CSV (модуль csv), JSON (модуль json), сериализация структур данных. Обработка исключений: try–except–else–finally, иерархия исключений, raise, проверка пользовательского ввода и входных данных. Классы и объекты в Python (обзор). Отладка и проверка программ: отладчик, assert; основы тестирования на Python: простейшие тесты с pytest/unittest для проверки кода и запуск тестов. Основы анализа данных: чтение набора данных, фильтрация, сортировка, группировка, описательные статистики средствами стандартной библиотеки; обзор библиотек NumPy, pandas и matplotlib; построение простейшего графика.

16. **Разработка приложений с графическим пользовательским интерфейсом.** Основные подходы к разработке приложений с визуальным интерфейсом: построение пользовательского взаимодействия, обработка событий, отображение данных и организация связи интерфейса с логикой программы.

4.3 Лабораторные работы

№ ЛР	№ раздела	Наименование лабораторных работ	Кол-во часов
1	2	Реализация различных видов алгоритмов на языке программирования высокого уровня C++	4
2	3	Использование возможностей языка C++ для обработки массивов	4
3	4	Работа со строками символов в языке C++	4
4	5	Работа с динамической памятью. Указатели	4
5	6	Создание собственных функций пользователя в языке C++	4
6	7	Использование возможностей языка C++ для работы с файлами	4
7	8	Создание приложений с использованием графических интерфейсов	6
8	9	Составные типы данных и структуры в C++	4
		Итого:	34

4.3 Лабораторные работы 2 семестр

№ ЛР	№ раздела	Наименование лабораторных работ	Кол-во часов
1	11	Реализация различных видов алгоритмов на языке Python	4
2	12	Создание пользовательских функций на языке Python	6
3	13	Работа со строками на языке Python	6
4	14	Работа со базовыми структурами данных (списки, кортежи, словари, множества)	6
5	15	Работа с файлами.	4
6	15	Основы анализа данных	4
7	16	Разработка GUI: основы Tkinter/PyQt, виджеты, обработка событий	4
		Итого:	34

4.4 Практические занятия (семинары)

№ занятия	№ раздела	Тема	Кол-во часов
1	11	Базовые конструкции Python: операторы, ветвления, циклы	4
2	11	Постановка задачи курсовой работы (индивидуального комплексного задания)	2
3	12	Пользовательские функции: аргументы, возврат, области видимости	2
4	14	Структуры данных Python: списки, кортежи, словари, множества	2
5	15	Файлы и обработка данных: чтение/запись, CSV, JSON, обработка исключений. Обработка набора данных	4
6	16	Тестирование	2
		Итого:	16

4.5 Расчетно-графическое задание 1 семестр

Тема расчетно-графического задания: "Разработка визуальных приложений на языке C++".

Цель: закрепление теоретических знаний и получение практических навыков программирования на языке высокого уровня при работе с одномерными и многомерных массивами, строками, указателями, а также пользовательскими функциями, также работа с окнами в языке C++.

Задание: разработать **визуальное приложение на языке C++** для обработки многомерных массивов с использованием пользовательских функций языка высокого уровня языке C++. В программе необходимо использовать работу с динамической памятью и строками.

Вариант 1. Протабулировать функцию y на заданном отрезке, затем записать в одномерный массив B все значения x, а в целочисленный массив C – все значения функции до минимального ее значения, округленные до целого.

y =3cos5x+2sinx [-π, π], h =π/10.

Далее найти максимальный элемент в массиве B, а в массиве C – минимальный. Из полученных массива B и массива C сформировать матрицу K, элементы главной диагонали матрицы - единицы, а вне главной диагонали должны вычисляться по формуле: K[I][J] = B[I]*C[J] – B[J]*C[I].

Размерность полученной матрицы – минимальная из размерностей массивов B и C. В полученной матрице K определить количество отрицательных элементов. В случае, если количество отрицательных элементов больше трети всех элементов массива, необходимо заменить все отрицательные элементы на нули, а если меньше, то оставить без изменения. Преобразовать элементы матрицы K в строку.

Затем все значения исходной функции на отрезке вывести на печать в виде таблицы "X Y", массивы B, C до и после изменений в виде строк с заголовками, матрицу K до и после изменения, все искомые результаты.

4.5 Курсовая работа (2 семестр)

Курсовая работа проводится по темам, востребованным в рамках будущей профессиональной деятельности обучающихся.

Обобщенная тема курсовой работы: «Разработка учета и анализа данных на основе обработки файлов».

Написать приложение на языке C++ или Python с использованием современных технологий визуального программирования.

При разработке приложения использовать элементы структурного программирования, реализовать работу с пользовательскими функциями, массивами, строками и динамической памятью. Пользовательские типы данных следует сохранить в отдельных файлах.

Для взаимодействия с пользователем программы необходимо разработать оконную форму для ввода запросов. Результат выполнения запросов программы необходимо сохранить в текстовый файл. При создании приложения следует разработать удобный пользовательский интерфейс.

Примерный вариант предметных областей для курсовой работы

Вариант № 1 (Гостиница).

Требуется разработать программу, предназначенную для администратора гостиницы. Такая программа должна обеспечивать хранение сведений об имеющихся в гостинице номерах, о проживающих в гостинице клиентах и о служащих. Количество номеров в гостинице известно, и имеются номера трех типов: одноместный, двухместный и трехместный, отличающиеся стоимостью проживания за сутки. Клиенту предоставляется место в номере. Номер оплачивается по прибытии. Служащий гостиницы убирает все номера на одном этаже в определенные дни недели, при этом в разные дни он может убирать разные этажи. В программе необходимо реализовать работу со следующими данными:

1) Номер (наименование, этаж, тип номера).
2) Клиент (фамилия, имя, отчество, № паспорта, откуда прибыл, номер, оплата (суток), когда прибыл, когда убыл).
3) Служащий (фамилия, имя, отчество, список этажей, список дней недели)

При работе с приложением должны быть реализованы следующие запросы:

- Список клиентов, прибывших из заданного города.
- Сколько в гостинице свободных номеров.
- Сколько служащих работали в определенный день на определенном этаже.

По результатам запросов необходимо сгенерировать форму, представленную в виде таблицы. Полученную таблицу сохранить в текстовый файл.

5 Учебно-методическое обеспечение дисциплины

5.1 Основная литература

1. Лаптев В.В. С ++. Объектно-ориентированное программирование: Учебное пособие.- СПб.: Питер, 2008. – 464 с.: ил.
2. Павловская Т.А. С/С ++. Структурное программирование: Практикум / Т.А. Павловская, Ю.А. Щупак. СПб.: Питер, 2007. – 239 с.: ил.
3. Павловская Т.А. С#. Программирование на языке высокого уровня. Учебник для вузов. – СПб.: Питер, 2013. – 432 с. : ил.

5.2 Дополнительная литература

1. Черносвитов, А. Visual C++ 7: руководство по практическому изучению / А. Черносвитов . - СПб. : Питер, 2002. - 528 с. : ил.
2. Сэвитч, У. C++ в примерах = Problem Solving with C++ : пер. с англ. / У. Сэвитч. - М.: ЭКОМ, 1997. - 736 с. : ил.
3. Груздев, Д. В. Программирование C++ (1 курс) : учебное пособие / Д. В. Груздев. — Воронеж : ВГУ, 2017. — 80 с. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/154781 (дата обращения: 19.03.2026). — Режим доступа: для авториз. пользователей.
4. Юрина, Т. А. Программирование и алгоритмизация : учебно-методическое пособие / Т. А. Юрина. — Омск : СибАДИ, 2021. — 88 с. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/179228 (дата обращения: 19.03.2026). — Режим доступа: для авториз. пользователей.

5.3 Периодические издания

1. Автоматизация в промышленности: журнал. - М.:Изд. дом "Инфоавтоматизация", 2026.
2. Программирование: журнал. – М.: Изд. «Национальная платформа научных периодических изданий», 2026.
3. Информационные технологии в проектировании и производстве: журнал. – М.: Изд. «ФГУП НТЦ оборонного комплекса «Компас», 2026.

5.4 Интернет-ресурсы

1 http://www.rsl.ru/ - Российская государственная библиотека.
2 http://window.edu.ru/ - ИС «Единое окно доступа к образовательным ресурсам.
3 http://e.lanbook.com/– Электронная библиотечная система «Издательство «Лань».
4 http://www.biblio-online.ru – Электронная библиотека издательства «Юрайт».
5 http://www.iprbookshop.ru - Электронная библиотечная система IPRbooks.
6 https://www.coursera.org/ - «Coursera»;
7 https://openedu.ru/ - «Открытое образование»;
8 https://universarium.org/ - «Универсариум»;
9 https://www.edx.org/ - «EdX»;
10 https://www.coursera.org/learn/python - «Coursera», МООК: «Programming for Everybody (Getting Started with Python)».

5.5 Программное обеспечение, профессиональные базы данных и информационные справочные системы современных информационных технологий

1. Операционная система РЕД ОС.
2. Пакет офисных приложений «МойОфис Образование»
3. Для работы с ресурсами Интернет - веб-браузер Яндекс https://yandex.ru/.
4. ГАРАНТ Платформа F1 [Электронный ресурс]: справочно-правовая система. / Разработчик ООО НПП «ГАРАНТ-Сервис», 119992, Москва, Воробьевы горы, МГУ, [1990–2026]. – Режим доступа в сети ОГУ http://garant.net.osu.ru
5. Автоматизированная интерактивная система сетевого тестирования - АИССТ (зарегистрирована в РОСПАТЕНТ, Свидетельство о государственной регистрации программы для ЭВМ №2011610456, правообладатель – Оренбургский государственный университет), режим доступа - http://aist.osu.ru.

6 Материально-техническое обеспечение дисциплины

Учебные аудитории для проведения занятий лекционного типа, семинарского типа, курсового проектирования, для проведения групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации.

Аудитории оснащены комплектами ученической мебели, техническими средствами обучения, служащими для представления учебной информации большой аудитории.

Для проведения практических и лабораторных занятий используется компьютерный класс, оснащенный компьютерной техникой, удовлетворяющей требованиям к конфигурации аппаратного обеспечения используемых программ.

Помещение для самостоятельной работы обучающихся оснащены компьютерной техникой, подключенной к сети "Интернет", и обеспечением доступа в электронную информационно-образовательную среду ОГУ.