

МИНПРОСВЕЩЕНИЯ РОССИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования
"Тульский государственный педагогический университет им. Л.Н. Толстого"
(ФГБОУ ВО "ТГПУ им. Л.Н. Толстого")

Параллельное программирование

рабочая программа дисциплины (модуля)

Закреплена за кафедрой	информатики и информационных технологий
ОПОП	09.03.03 Прикладная информатика направленность (профиль) Прикладная информатика в здравоохранении
Квалификация	Бакалавр
Год начала подготовки	2019
Форма обучения	очная
Общая трудоемкость	4 з.е.

Виды контроля по семестрам:
экзамен 6

Семестр(Курс.Номер семестра на курсе)	6(3.2)		Итого	
	УП	РПД	УП	РПД
Лекции	18	18	18	18
Лабораторные	22	22	22	22
Итого ауд.	40	40	40	40
КСР	2	2	2	2
Контактная работа	42	42	42	42
Сам. работа	66	66	66	66
Часы на контроль	36	36	36	36
Итого трудоемкость в часах	144	144	144	144

Программу составил(и):

к.т.н., доцент, Клепиков А.К.; ассистент, Гладких И.Ю.

Рабочая программа дисциплины

Параллельное программирование

разработана в соответствии с ФГОС:

Федеральный государственный образовательный стандарт высшего образования по направлению подготовки 09.03.03 Прикладная информатика (уровень бакалавриата) (приказ Минобрнауки России от 19.09.2017 г. № 922)

составлена на основании учебного плана:

09.03.03 Прикладная информатика
направленность (профиль) Прикладная информатика в здравоохранении
утвержденного Учёным советом вуза от 30.05.2019 протокол № 6.

Рабочая программа одобрена на заседании кафедры

информатики и информационных технологий

Зав. кафедрой Богатырева Ю.И.

РПД утверждена Учёным советом университета
протокол от 30.5.2019 г. № 6

1. ЦЕЛИ ОСВОЕНИЯ ДИСЦИПЛИНЫ

Цель преподавания дисциплины - изучение и практическое освоение общих принципов и современных методов параллельного программирования программирования.

2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ООП

Цикл (раздел) ООП:	Б1.В
2.1	Требования к предварительной подготовке обучающегося:
1.	Объектно-ориентированное программирование
2.	Технологии веб-программирования
3.	Методы программирования
4.	Структуры и алгоритмы компьютерной обработки данных
5.	Алгоритмизация и программирование
6.	технологическая практика
2.2	Дисциплины и практики, для которых освоение данной дисциплины (модуля) необходимо как предшествующее:
1.	Теория автоматов и формальных языков
2.	Основы веб-дизайна
3.	Разработка программных приложений для здравоохранения

3. СООТНЕСЕНИЕ РЕЗУЛЬТАТОВ ОБУЧЕНИЯ ПО ДИСЦИПЛИНЕ (МОДУЛЮ) С ИНДИКАТОРАМИ ДОСТИЖЕНИЯ КОМПЕТЕНЦИЙ

3.1 Компетенции обучающегося и индикаторы их достижения:

ОПК-7: Способен разрабатывать алгоритмы и программы, пригодные для практического применения;	
ОПК-7.1	Знает основные языки программирования и работы с базами данных, операционные системы и оболочки, современные программные среды разработки информационных систем и технологий
	критерии оценки эффективности параллельных программ и их ограничения параллельные вычислительные методы архитектурные принципы реализации параллельной обработки в вычислительных машинах
ОПК-7.2	Умеет применять языки программирования и работы с базами данных, современные программные среды разработки информационных систем и технологий для автоматизации бизнес-процессов, решения прикладных задач различных классов, ведения баз данных и информационных хранилищ
	разрабатывать параллельные программы с использованием библиотеки MPI
ОПК-7.3	Владеет навыками программирования, отладки и тестирования прототипов программно-технических комплексов задач
	настройки высокопроизводительных вычислительных систем оценки качества настройки высокопроизводительных вычислительных систем
ПК-2: Способность разрабатывать и адаптировать прикладное программное обеспечение	
ПК-2.1	Знает принципы оформления программного кода в соответствии с установленными требованиями
	методы и языковые механизмы конструирования параллельных программ
ПК-2.2	Имеет навык использования современных языков программирования и методик разработки и внедрения прикладного программного обеспечения
	конструирования высокопроизводительных вычислительных систем
ПК-2.3	Умеет внедрять и адаптировать прикладное программное обеспечение
	осуществлять перенос реализованных программных средств на параллельные платформы проводить сравнительный анализ последовательных и параллельных программных средств и оценивать их эффективность

3.2 Результаты обучения по дисциплине:

В результате освоения дисциплины обучающийся должен:

	Знать:
3.1	критерии оценки эффективности параллельных программ и их ограничения
3.2	параллельные вычислительные методы
3.3	архитектурные принципы реализации параллельной обработки в вычислительных машинах
3.4	методы и языковые механизмы конструирования параллельных программ
	Уметь:
У.1	разрабатывать параллельные программы с использованием библиотеки MPI
У.2	осуществлять перенос реализованных программных средств на параллельные платформы

У.3	проводить сравнительный анализ последовательных и параллельных программных средств и оценивать их эффективность
	Владеть:
В.1	конструирования высокопроизводительных вычислительных систем
В.2	настройки высокопроизводительных вычислительных систем
В.3	оценки качества настройки высокопроизводительных вычислительных систем

4. СТРУКТУРА И СОДЕРЖАНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)

Код занятия	Наименование разделов и тем /вид занятия/	Семестр / Курс	Часов	Литература	Содержание
	Введение в предмет				
1.1	Модели параллельного программирования /Лек/	6	2	Л1.1Л2.1 Л2.2 Л2.3 Л2.4	Формальные модели параллельного программирования. Параллельные схемы программ. Информационный базис и схема управления. Асинхронные вычислительные процессы над информационным базисом. Неуправляемые вычислительные процессы.
1.2	Формальные модели параллельного программирования /Ср/	6	4	Л1.1Л2.1 Л2.2 Л2.3 Л2.4	Формальные модели параллельного программирования. Параллельные схемы программ. Информационный базис и схема управления. Асинхронные вычислительные процессы над информационным базисом. Неуправляемые вычислительные процессы.
1.3	Схемы управления и эквивалентность вычислительных процессов /Лек/	6	2	Л1.1Л2.1 Л2.2 Л2.3 Л2.4	Неуправляемые вычислительные процессы. Схема управления параллельной схемы программы. Управляющий автомат и особенности автоматных функций. Эквивалентность вычислительных процессов по произвольному отношению. Детерминизм и отношение "большей параллельности" схем управления.
1.4	Управление параллельными схемами программы /Ср/	6	4	Л1.1 Л1.2Л2.1 Л2.2 Л2.3 Л2.4	Неуправляемые вычислительные процессы. Схема управления параллельной схемы программы. Управляющий автомат и особенности автоматных функций. Эквивалентность вычислительных процессов по произвольному отношению. Детерминизм и отношение "большей параллельности" схем управления.
1.5	Лабораторная работа №1 /Лаб/	6	4	Л1.1 Л1.2Л2.1 Л2.2 Л2.3 Л2.4	Параллельные схемы программ. Асинхронные вычислительные процессы над информационным базисом. Неуправляемые вычислительные процессы. Схема управления параллельной схемы программы. Управляющий автомат и особенности автоматных функций.
1.6	Работа над индивидуальным проектом /Ср/	6	6	Л1.1 Л1.2Л2.1 Л2.2 Л2.3 Л2.4	Параллельные схемы программ. Асинхронные вычислительные процессы над информационным базисом. Неуправляемые вычислительные процессы. Схема управления параллельной схемы программы. Управляющий автомат и особенности автоматных функций.
	Вычислительные системы с массовым параллелизмом				
2.1	Моделирующие вычислительные системы и системы реального времени: особенности применения /Лек/	6	2	Л1.1Л2.1 Л2.2 Л2.3 Л2.4	Определение и особенности применения: моделирующие вычислительные системы и системы реального времени. Примеры. Характерные особенности таких систем.

2.2	Характеристики и особенности моделирующих вычислительных систем и систем реального времени /Ср/	6	4	Л1.1Л2.1 Л2.2 Л2.3 Л2.4	Определение и примеры, особенности применения: моделирующие вычислительные системы и системы реального времени. Характерные особенности таких систем.
2.3	Классификация вычислительных систем /Лек/	6	2	Л1.1Л2.1 Л2.2 Л2.3 Л2.4	Классификация ВС по Флинну. Реконфигурируемые системы MIMD архитектуры. Статическая и динамическая реконфигурация. Оценка производительности массивно параллельных ВС. Причины снижения производительности.
2.4	Оценка производительности вычислительных систем /Ср/	6	4	Л1.1Л2.1 Л2.2 Л2.3 Л2.4	Классификация ВС по Флинну. Реконфигурируемые системы MIMD архитектуры. Статическая и динамическая реконфигурация. Оценка производительности массивно параллельных ВС. Причины снижения производительности.
2.5	Лабораторная работа №2 /Лаб/	6	4	Л1.1 Л1.2Л2.1 Л2.2 Л2.3 Л2.4	Определение и примеры. Особенности применения: моделирующие вычислительные системы и системы реального времени. Примеры. Характерные особенности таких систем. Классификация ВС по Флинну. Реконфигурируемые системы MIMD архитектуры. Статическая и динамическая реконфигурация. Оценка производительности массивно параллельных ВС. Причины снижения производительности.
2.6	Работа над индивидуальным проектом /Ср/	6	6	Л1.1 Л1.2Л2.1 Л2.2 Л2.3 Л2.4	
	Параллельное программирование в MPI				
3.1	Структура библиотеки MPI /Лек/	6	2	Л1.1 Л1.2Л2.1 Л2.2 Л2.3 Л2.4	Краткая характеристика библиотеки MPI. Структура библиотеки MPI. Интеграция библиотеки MPI в код на языке C++ и в проекты. Общие процедуры MPI.
3.2	Лабораторная работа № 3 /Лаб/	6	2	Л1.1 Л1.2Л2.1 Л2.2 Л2.3 Л2.4	Структура библиотеки MPI. Интеграция библиотеки MPI в код на языке C++ и в проекты. Общие процедуры MPI.
3.3	Интеграция элементов из библиотеки MPI в проекты на языке C++ /Ср/	6	4	Л1.1 Л1.2Л2.1 Л2.2 Л2.3 Л2.4	Описание библиотеки MPI: различные версии и модификации для разных языков программирования и платформ выполнения программ. Структура библиотеки MPI. Интеграция библиотеки MPI в код на языке C++ и в проекты. Общие процедуры MPI.
3.4	Организация обмена сообщениями /Лек/	6	2	Л1.1 Л1.2Л2.1 Л2.2 Л2.3 Л2.4	Модели вычислений. Организация обмена сообщениями. Коллективные функции.
3.5	Кластерные технологии /Ср/	6	4	Л1.1 Л1.2Л2.1 Л2.2 Л2.3 Л2.4	Модели вычислений. Кластерные технологии. Организация кластера на базе локальной сети.
3.6	Лабораторная работа №4 /Лаб/	6	4	Л1.1 Л1.2Л2.1 Л2.2 Л2.3 Л2.4	Интеграция библиотеки MPI в код на языке C++ и в проекты. Организация приема/передачи данных между отдельными процессами. Коллективные функции. Коммуникаторы, группы и области связи
3.7	Работа над индивидуальным проектом /Ср/	6	6	Л1.1 Л1.2Л2.1 Л2.2 Л2.3 Л2.4	
	Реализация параллельных алгоритмов				

4.1	Параллельные алгоритмы /Лек/	6	2	Л1.1 Л1.2Л2.1 Л2.2 Л2.3 Л2.4	Параллельные алгоритмы. Общие способы распараллеливания алгоритмов. Организация параллельного исполнения рекурсивных вычислений. Параллельные численные алгоритмы линейной алгебры.
4.2	Этапы разработки параллельного алгоритма и рекурсии в параллельном программировании /Ср/	6	4	Л1.1 Л1.2Л2.1 Л2.2 Л2.3 Л2.4	Декомпозиция, проектирование коммуникаций, укрупнение, планирование вычислений. Распараллеливание рекурсии: представление в виде простой и сложной рекурсии или в виде цикла.
4.3	Вычисления над матричными данными в параллельном программировании /Лек/	6	2	Л1.1 Л1.2Л2.1 Л2.2 Л2.3 Л2.4	Параллельные численные алгоритмы линейной алгебры. Способы разбиения матриц (горизонтальная, вертикальная, блочные схемы). Методы вычисления произведения матриц с использованием разных схем разбиения матриц. Обеспечение предельно допустимого параллелизма. Обращение матриц.
4.4	Лабораторная работа №5 /Лаб/	6	4	Л1.1 Л1.2Л2.1 Л2.2 Л2.3 Л2.4	Практическая реализация параллельных алгоритмов. Параллельные программы рекурсивных вычислений. Параллельные численные алгоритмы линейной алгебры. Практическая реализация алгоритмов разбиения матриц (горизонтальная, вертикальная, блочные схемы).
4.5	Практические приемы параллельного программирования. Параллельные вычисления над матричными данными. /Ср/	6	6	Л1.1 Л1.2Л2.1 Л2.2 Л2.3 Л2.4	Реализация параллельных алгоритмов. Практическая организация параллельного исполнения рекурсивных вычислений. Параллельные численные алгоритмы линейной алгебры. Способы разбиения матриц (горизонтальная, вертикальная, блочные схемы). Практические реализации методов вычисления произведения матриц с использованием разных схем разбиения матриц. Обеспечение и доказательство предельно допустимого параллелизма. Практические реализации параллельных алгоритмов обращение матриц
4.6	Параллелизация решения уравнений /Лек/	6	2	Л1.1 Л1.2Л2.1 Л2.2 Л2.3 Л2.4	Параллельные методы решения систем линейных уравнений. Параллельные численные алгоритмы решения дифференциальных уравнений в частных производных.
4.7	Лабораторная работа №6 /Лаб/	6	4	Л1.1 Л1.2Л2.1 Л2.2 Л2.3 Л2.4	Практическая реализация параллельных методов решения систем линейных уравнений. Практическая реализация параллельных численных алгоритмов решения дифференциальных уравнений в частных производных.
4.8	Практические приемы параллельного решения уравнений /Ср/	6	4	Л1.1 Л1.2Л2.1 Л2.2 Л2.3 Л2.4	Параллельные методы решения систем линейных уравнений. Параллельные численные алгоритмы решения дифференциальных уравнений в частных производных.
4.9	Работа над индивидуальным проектом /Ср/	6	4	Л1.1 Л1.2Л2.1 Л2.2 Л2.3 Л2.4	
4.10	Подготовка к зачету /Ср/	6	6	Л1.1 Л1.2Л2.1 Л2.2 Л2.3 Л2.4	Вопросы к зачету

5. ОЦЕНОЧНЫЕ МАТЕРИАЛЫ

5.1. Типовые задания для проведения текущего контроля

Образцы заданий к лабораторным работам:

1. Написать программу, используя коммуникационные функции (MPI_Ssend, MPI_Bsend, MPI_Rsend, MPI_Isend, MPI_Irecv), передающие одномерные и двумерные массивы (вектора и матрицы) между двумя процессорами
2. Провести сравнение по скорости передачи данных в зависимости от применяемых функций и размера передаваемых данных
3. Написать программу, используя коммуникационную функцию (MPI_Bcast), реализующую алгоритм передачи данных от 0 процесса всем остальным
4. Написать программу, используя коммуникационную функцию (MPI_Gather), реализующую алгоритм передачи частей массива от всех процессоров на 0.
5. Написать программу, используя коммуникационную функцию (MPI_Allgather), реализующую алгоритм передачи частей массива от всех процессоров на все процессора.

Индивидуальное проектное задание заключается в разработке параллельной программы:

Каждый проект независимо от темы и задания должен предоставлять пользователю выбор количества узлов, на которых будет запущен процесс вычислений, а также выводить, время, за которое они произошли.

Индивидуальное проектное задание заключается в разработке параллельной программы:

Каждый проект независимо от темы и задания должен предоставлять пользователю выбор количества узлов, на которых будет запущен процесс вычислений, а также выводить, время, за которое они произошли.

1. Нахождение кратчайших путей для ориентированного графа

Исходной информацией для задачи является взвешенный граф, $G=(V,R)$ содержащий n вершин ($|V|=n$), в котором каждому ребру графа приписан неотрицательный вес. Граф будем полагать ориентированным, т.е., если из вершины i есть ребро в вершину j , то из этого не следует наличие ребра из j в i . В случае, если вершины все же соединены взаимнообратными ребрами, то веса, приписываемые им, могут не совпадать. Для имеющегося графа G требуется найти минимальные длины путей между каждой парой вершин графа.

Входные данные начинаются N количеством вершин от 0 до 10000 и M количеством ребер в графе от 0 до 10000 через пробел, затем в M строках записаны номера вершин соединенных i -ым ребром и его вес от 0 до 10000, где i – номер строки, начиная счет со второй строки.

Выходные данные должны содержать все кратчайшие пути для всех пар вершин, если такой путь существует для пары. Каждый путь должен быть записан на отдельной строке в виде последовательности номеров ребер через пробел в том порядке, в котором они включены в путь, и к ним добавить последнее после знака «=» число, показывающее вес пути, в начале строки должны быть помещены номера вершин, соединенных этим путем («I J : »). В случае отсутствия пути между вершинами поставить вместо последовательности вершин прочерк.

Пример входных(input.txt) и выходных(output.txt) данных:

input.txt output.txt

```

5 4
1 2 1
1 3 8
1 4 3
2 3 2
3 1 3
4 2 1
1 2 : 1 = 1
1 3 : 1 4 = 4
1 4 : 3 = 3
1 5 : -
2 1 : -
2 3 : 4 = 3
2 4 : -
2 5 : -
3 1 : 5 = 3
3 2 : -
3 4 : -
3 5 : -
4 1 : -
4 2 : 6 = 1
4 3 : -
4 5 : -
5 1 : -
5 2 : -
5 3 : -
5 4 : -

```

Примечание: в случае если наименьший путь между парой вершин не является единственным, считать кратчайшим тот, который содержит наименьшее количество вершин, если же данное условие не дает единственное решение, вывести любой из них.

2. Решение систем линейных уравнений методом алгоритма Гаусса, рекурсивное удвоение Стоуна

Задача: дана расширенная (добавлен вектор-столбец свободных членов) матрица коэффициентов системы линейных

уравнений, найти решение системы.

Входной файл в первой строке содержит N количество строк и M количество столбцов матрицы через пробел, далее в N строках содержится M элементов (числа от -10000 до 10000) матрицы через пробел. Порядок расположения элементов матрицы внутри файла соответствует порядку расположения элементов в самой матрице.

$-10000 \leq N, M \leq 10000$

Выходной файл должен содержать значение переменных, совокупность которых является решением данной системы линейных уравнений, через пробел. Порядок расположения значений переменных должен соответствовать порядку номеров их переменных. В случае невозможности нахождения единственного решения вывести в файл сообщение «Error Data». В случае решения системы равного пустому множеству вывести сообщение «No Solutions»

Пример входных(input.txt) и выходных(output.txt) данных:

input.txt output.txt

1 2

-7 21

-3

input.txt output.txt

2 3

1 1 5

1 -2 -4

5.2. Типовые задания для проведения промежуточной аттестации

Вопросы к зачету

1. Что такое параллельное программирование и суперкомпьютеры. Области в которых может возникать потребность в параллельных вычислениях. Особенности параллельных вычислений.
2. Увеличение производительности при параллельных вычислениях. К каким областям задач может быть применено распараллеливание к каким нет. Закон Амдала.
3. Ускорение параллельного алгоритма.
4. Эффективность параллельного алгоритма.
5. Распараллеливание арифметических выражений. Алгоритм Винограда.
6. Разделение на подзадачи. Установление связей между отдельными подзадачами. Объединение мелких подзадач в большие, законченные счетные единицы (агломерация).
7. Средства поддержки параллельной работы: параллелизм процессов, механизм синхронизации и механизм разделения ресурсов, семафоры, обмен сообщениями.
8. Библиотека MPI. Инициализация, завершение. Базовые функции.
9. Точечные обмены. Синхронный и асинхронный режим.
10. Буферизация сообщений.
11. Отложенные операции. Функции инициализации и завершения операции.
12. Коллективные взаимодействия. Передача сообщений
13. Управление группами и коммуникатором.
14. Коллективные взаимодействия. Прием сообщений
15. Создание и освобождение коммуникатора.
16. Типы данных в MPI.
17. Пакет mpich. Установка и настройка кластера под управлением Linux.
18. Простейшие параллельные алгоритмы и их степень параллелизма.
19. Параллельный алгоритм скалярного умножения векторов и его ускорение по сравнению с последовательным алгоритмом.
20. Параллельный алгоритм умножения матрицы на вектор и его ускорение по сравнению с последовательным алгоритмом.
21. Параллельный алгоритм умножения матрицы на матрицу и его ускорение по сравнению с последовательным алгоритмом.
22. Параллельный алгоритм решения СЛАУ прямым методом Гаусса и его ускорение по сравнению с последовательным алгоритмом.
23. Технологии параллельного программирования.
24. Кластерные технологии.

5.3. Перечень видов оценочных средств

1. Лабораторные работы.
2. Индивидуальное проектное задание.
3. Вопросы к зачету.

5.4. Процедура применения оценочных материалов

Промежуточная аттестация может проводиться с применением электронного обучения и (или) дистанционных образовательных технологий в соответствии с "Порядком проведения промежуточной аттестации с применением электронного обучения и /или дистанционных образовательных технологий".

Описание балльно-рейтинговой системы по дисциплине.

Итоговая рейтинговая оценка по дисциплине складывается из следующих составляющих:

- 1) В течение семестра за выполнение заданий по курсу студент может максимально получить 70 баллов.;
- 2) Обязательной формой текущей аттестации знаний является выполнение индивидуального проектного задания 20 баллов.
- 3) На зачёте ответ студента может быть максимально оценен в 30 баллов.

При этом, для получения положительной итоговой оценки на зачете необходимо получить не менее 60% по каждой составляющей и выполнить все задания для лабораторных работ. Шкала перевода баллов в оценку: до 40 - «не зачтено»; 41 - 100 - «зачтено».

Выполнение заданий: 70
Лабораторные работы 40
Индивидуальное проектное задание 20
Зачет 30
ИТОГО: 100

6. УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)

6.1. Рекомендуемая литература

6.1.1. Основная литература

	Авторы, составители	Заглавие	Издательство, год (кол-во экземпляров для печатных изданий)	Ссылка на электронное издание
Л1.1	Антонов А. С.	Параллельное программирование с использованием технологии MPI: курс	, 2008	http://biblioclub.ru/index.php?page=book&id=233577
Л1.2	Алексеев А. А.	Основы параллельного программирования с использованием Visual Studio 2010	, 2016	http://biblioclub.ru/index.php?page=book&id=428829

6.1.2. Дополнительная литература

	Авторы, составители	Заглавие	Издательство, год (кол-во экземпляров для печатных изданий)	Ссылка на электронное издание
Л2.1	Куперман В. Г., Торина Е. Г.	Основы информатики и начала программирования: Учебное пособие	, 1997 (1 шт.)	
Л2.2	Лаптев В. В., Морозов А. В., Бокова А. В.	C++. Объектно-ориентированное программирование: Учебное пособие для студентов вузов	СПб.: Питер, 2008 (20 шт.)	
Л2.3	Лупин С. А., Посыпкин М. А.	Технологии параллельного программирования: Учебное пособие для студентов вузов	, 2008 (10 шт.)	
Л2.4	Мартынюк Ю. М., Гербут С. С., Ванькова В. С., Торина Е. Г., Снижко Е. А.	Методы программирования: Учебное пособие	, 2013 (10 шт.)	

6.2. Перечень ресурсов информационно-телекоммуникационной сети "Интернет"

Э1	СITForum.ru [Электронный ресурс] : образовательный портал / "ЦИТ Форум". - [Б. м. : б. и.], 2019. - Загл. с титул. экрана.			
Э2	Math-Net.Ru [Электронный ресурс] : общероссийский математический портал / Математический институт им. В. А. Стеклова РАН ; Российская академия наук, Отделение математических наук. - М. : [б. и.], 2019. - Загл. с титул. экрана. - Б. ц.			
Э3	ИКТ [Электронный ресурс] : федеральный образовательный портал / ФГАУ ГНИИ ИТТ "Информика". - М. : [б. и.], 2003. - Загл. с титул. экрана. - Б. ц.			
Э4	Университетская библиотека Online [Электронный ресурс] : электронная библиотечная система / ООО "Директ-Медиа". - М. : [б. и.], 2019. - Загл. с титул. экрана. - Б. ц.			
Э5	Научная электронная библиотека eLIBRARY.RU [Электронный ресурс] : информационный портал / ООО "РУНЭБ" ; Санкт-Петербургский государственный университет. - М. : [б. и.], 2019. - Загл. с титул. экрана. - Б. ц.			

6.3. Информационные технологии

6.3.1 Перечень лицензионного и свободно распространяемого программного обеспечения

1.	Операционная система ROSA Enterprise Linux Desktop № RL00450-1-110518-01. RL00450-1-110518-17 от 11 мая 2018 г.			
2.	Операционная система Microsoft Windows Professional 7 Russian. Лицензия №48497058 от 13.05.2011 г., договор № Пр/16/6 от 05 апреля 2016 г.			
3.	Операционная система Microsoft Windows 10 Professional Russian. Контракт № ПР/ФЕН/15/18 от 23.10.2015 г., договор № Пр/16/6 от 05 апреля 2016 г.			

4.	Файловый архиватор 7z. Свободно распространяемое ПО
5.	Текстовый редактор NotePad++. Свободно распространяемое ПО
6.	Файловый менеджер Far manager. Свободно распространяемое ПО
7.	Оболочка программирования Code: Blocks 17.12. Свободно распространяемое ПО
8.	Среда программирования и набор инструментов для программирования. MinGW 0.6.3 Свободно распространяемое ПО

6.3.2 Перечень информационных справочных систем и профессиональных баз данных

1.	Полнотекстовый архив ведущих западных научных журналов на российской платформе Национального электронно-информационного консорциума (НЭИКОН) (http://neicon.ru)
2.	Web of Science Core Collection – политематическая реферативно-библиографическая и наукометрическая (библиометрическая) база данных (http://webofscience.com)
3.	Портал «Информационно-коммуникационные технологии в образовании» (http://www.ict.edu.ru)
4.	Официальный интернет-портал базы данных правовой информации (http://pravo.gov.ru)
5.	Портал Федеральных государственных образовательных стандартов высшего образования (http://fgosvo.ru)

7. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)

Ауд.	Назначение	Оборудование и технические средства обучения	Вид
2-15	Компьютерный класс	компьютеры, рулонный экран, стол преподавателя, столы компьютерные, переносной проектор	Экзамен
3-108	Компьютерный класс	доска учебная, компьютеры	КСР
4-303	Помещение для самостоятельной работы	аудиоколонки, кондиционер, маркерная доска, столы компьютерные, столы учебные, компьютерная техника с возможностью подключения сети "Интернет" и обеспечением доступа в электронную информационно-образовательную среду университета	Ср
4-305	Компьютерный класс	аудиоколонки для проектора и интерактивной доски, аудиоколонки учебные, интерактивная доска, компьютеры, кондиционер, маркерная доска, проектор, столы компьютерные, столы учебные	Лаб
4-306	Компьютерный класс	аудиоколонки для проектора и интерактивной доски, интерактивная доска, компьютеры, кондиционер, маркерная доска, проектор, столы компьютерные, столы учебные	Лек
4-318	Компьютерный класс	компьютеры, маркерная доска, серверная стойка лаборатории МТС, стол преподавателя, столы компьютерные, столы учебный большой	Лаб

8. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ОБУЧАЮЩИХСЯ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ (МОДУЛЯ)

Приступая к изучению новой учебной дисциплины, студенты должны ознакомиться с учебной программой, учебной, научной и методической литературой, имеющейся в библиотеке университета, встретиться с преподавателем, ведущим дисциплину, получить в библиотеке рекомендованные учебники и учебно-методические пособия, осуществить запись на соответствующий курс в среде электронного обучения университета.

Глубина усвоения дисциплины зависит от активной и систематической работы студента на лекциях и лабораторных работ, а также в ходе самостоятельной работы, по изучению рекомендованной литературы.

На лекциях важно сосредоточить внимание на ее содержании. Это поможет лучше воспринимать учебный материал и уяснить взаимосвязь проблем по всей дисциплине. Основное содержание лекции целесообразнее записывать в тетради в виде ключевых фраз, понятий, тезисов, обобщений, схем, опорных выводов. Необходимо обращать внимание на термины, формулировки, раскрывающие содержание тех или иных явлений и процессов, научные выводы и практические рекомендации. Желательно оставлять в конспектах поля, на которых делать пометки из рекомендованной литературы, дополняющей материал прослушанной лекции, а также подчеркивающие особую важность тех или иных теоретических положений. С целью уяснения теоретических положений, разрешения спорных ситуаций необходимо задавать преподавателю уточняющие вопросы. Для закрепления содержания лекции в памяти, необходимо во время самостоятельной работы внимательно прочесть свой конспект и дополнить его записями из учебников и рекомендованной литературы. Конспектирование читаемых лекций и их последующая доработка способствует более глубокому усвоению знаний, и поэтому являются важной формой учебной деятельности студентов.

Прочное усвоение и долговременное закрепление учебного материала невозможно без продуманной самостоятельной работы. Такая работа требует от студента значительных усилий, творчества и высокой организованности. В ходе самостоятельной работы студенты выполняют следующие задачи: дорабатывают лекции, изучают рекомендованную литературу, готовятся к лабораторным работам, к коллоквиуму, контрольным работам по отдельным темам дисциплины. При этом эффективность учебной деятельности студента во многом зависит от того, как он распорядился выделенным для самостоятельной работы бюджетом времени.

Результатом самостоятельной работы является прочное усвоение материалов по предмету согласно программе дисциплины. В итоге этой работы формируются профессиональные умения и компетенции, развивается творческий подход к решению возникших в ходе учебной деятельности проблемных задач, появляется самостоятельности мышления.

Целью лабораторных работ по данной дисциплине является закрепление теоретических знаний, полученных при изучении

дисциплины.

При подготовке к лабораторной работе целесообразно выполнить следующие рекомендации: изучить основную литературу; ознакомиться с дополнительной литературой, новыми публикациями в периодических изданиях: журналах, газетах и т. д.; при необходимости доработать конспект лекций. При этом учесть рекомендации преподавателя и требования учебной программы.

При выполнении лабораторных работ основным методом обучения является самостоятельная работа студента под управлением преподавателя. На них пополняются теоретические знания студентов, их умение творчески мыслить, анализировать, обобщать изученный материал, проверяется отношение студентов к будущей профессиональной деятельности.

Оценка выполненной работы осуществляется преподавателем комплексно: по результатам выполнения заданий, устному сообщению и оформлению работы. После подведения итогов занятия студент обязан устранить недостатки, отмеченные преподавателем при оценке его работы.