

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ

Федеральное государственное бюджетное образовательное учреждение высшего  
образования  
«ДАГЕСТАНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

Факультет математики и компьютерных наук

**РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ**  
Параллельные вычисления

Кафедра дискретной математики и информатики

Образовательная программа  
**02.03.02 - Фундаментальная информатика и информационные технологии**

Направленность (профиль) программы:  
**Информатика и компьютерные науки**

Уровень высшего образования  
**бакалавриат**

Форма обучения  
**очная**

Статус дисциплины: входит в часть ОПОП, формируемую  
участниками образовательных отношений

Махачкала, 2021

Рабочая программа дисциплины «Параллельные вычисления» составлена в 2021 году в соответствии с требованиями ФГОС ВО по направлению подготовки 02.03.02- Фундаментальная информатика и информационные технологии (уровень бакалавриата) № 808 от 23.08.2017.

Разработчик: кафедра дискретной математики и информатики, Ханикалов Х.Б.-ст. преподаватель кафедры дискретной математики и информатики

Рабочая программа дисциплины одобрена:

на заседании кафедры дискретной математики и информатики от 30 мая 2021 г., протокол № 9.

Зав. кафедрой М.М.В. Магомедов А.М.  
(подпись)

и  
на заседании Методической комиссии ФМИКН от 23 июня 2021 г., протокол № 6.

Председатель В.Д.Б. Бейбалаев В.Д.  
(подпись)

Рабочая программа дисциплины согласована с учебно-методическим управлением  
« 09 » 07 2021 г. А.Г.Г. Гасангаджиева А.Г.  
(подпись)

## Аннотация рабочей программы дисциплины

Дисциплина “Параллельные вычисления” входит в часть ОПОП, формируемую участниками образовательных отношений обязательных дисциплин бакалавриата по направлению 02.03.02 - Фундаментальная информатика и информационные технологии.

Дисциплина реализуется на факультете математики и компьютерных наук кафедрой дискретной математики и информатики.

Содержание дисциплины охватывает круг вопросов, связанных с современными знаниями в области параллельного и распределенного программирования.

Дисциплина нацелена на формирование следующих компетенций выпускника: общепрофессиональных – ОПК-2, профессиональных – ПК-4.

Преподавание дисциплины предусматривает проведение следующих видов учебных занятий: лекции.

Рабочая программа дисциплины предусматривает проведение следующих видов контроля успеваемости: в форме контрольной работы и итогового зачета в конце семестра.

Объем дисциплины - 2 зачетные единицы, в том числе в академических часах по видам учебных занятий:

Се- местр	Учебные занятия						СРС, в том числе экза- мен	Форма проме- жуточной атте- стации (зачет, дифференциро- ванный зачет, экзамен
	в том числе							
	Контактная работа обучающихся с преподавателем							
	Все го	из них						
		Лек- ции	Лабора- торные занятия	Практи- ческие занятия	КСР	Кон- сульта- ции		
8	72	14				58	зачет	

### 1. Цели освоения дисциплины

Цель изучения дисциплины «Параллельные вычисления» - приобретение студентами теоретических знаний и устойчивых навыков использования параллельного программирования. Основная задача дисциплины: подготовка студентов к работе по созданию и поддержке современных распределенных информационных систем.

## 2. Место дисциплины в структуре ОПОП бакалавриата

Дисциплина «Параллельные вычисления» входит в часть образовательной программы формируемую участниками образовательных отношений бакалавриата по направлению 02.03.02 и изучается в соответствии с графиком учебного процесса в 8 семестре (14 ч. лек., 58 ч. СР, зачет).

Дисциплина логически и содержательно взаимосвязана с такими дисциплинами, как «Основы программирования», «Языки программирования», «Языки программирования».

## 3. Компетенции обучающегося, формируемые в результате освоения дисциплины (перечень планируемых результатов обучения).

### 1. Компетенции обучающегося, формируемые в результате освоения дисциплины

Процесс изучения дисциплины направлен на формирование элементов следующих компетенций в соответствии с ФГОС ВО по данному направлению подготовки: ОПК - 2, ПК - 4.

Код и наименование компетенции из ОПОП	Код и наименование индикатора достижения компетенций	Планируемые результаты обучения	Процедура освоения
ОПК-2. Способен применять компьютерные/суперкомпьютерные методы, современное программное обеспечение, в том числе отечественного происхождения, для решения задач профессиональной деятельности	ОПК-2.1. Знает основные положения и концепции в области программирования, архитектуру языков программирования, теории коммуникации, знает основную терминологию, знаком с содержанием Единого Реестра Российских программ.	Знать: основы параллельной обработки данных; Уметь: разрабатывать, отлаживать и запускать на исполнение параллельные программы с использованием функций MPI; Владеть: навыками программирования параллельных вычислений.	Участие в лабораторных занятиях. Самостоятельная работа с программой MPI. Изучение лекционного материала.

	<p>ОПК-2.2.</p> <p>Умеет анализировать типовые языки программирования, составлять программы.</p>	<p>Знать: основные алгоритмы параллельной обработки данных;          Уметь: создавать программные продукты с применением параллельных алгоритмов;          Владеть: навыками использования технологии MPI для решения прикладных задач.</p>	
	<p>ОПК-2.3.</p> <p>Имеет практический опыт решения задач анализа, интеграции различных типов программного обеспечения, анализа типов коммуникаций.</p>	<p>Знать: методы параллельного программирования;          Уметь: написать программы на языках программирования, поддерживающих распараллеливание;          Владеть: навыками использования технологии OpenMP для решения прикладных задач.</p>	

<p>ПК-4. Способность применять в профессиональной деятельности современные языки программирования и методы параллельной обработки данных, операционные системы, электронные библиотеки и пакеты программ, сетевые технологии.</p>	<p>ПК-4.1.</p> <p>Знает современные языки программирования и методы параллельной обработки данных. Знаком с содержанием Единого Реестра Российских программ для электронных вычислительных машин и баз данных.</p>	<p>Знать: теорию параллельных вычислений;</p> <p>Уметь: использовать теорию параллельных вычислений при практической реализации;</p> <p>Владеть: навыками проектирования эффективных алгоритмов параллельных вычислений для решения прикладных задач.</p>	<p>Участие в лабораторных занятиях. Самостоятельная работа с программой. Изучение лекционного материала.</p>
	<p>ПК-4.2.</p> <p>Умеет реализовывать численные методы решения прикладных задач в профессиональной сфере деятельности, пакеты программного обеспечения, операционные системы, электронные библиотеки, сетевые технологии.</p>	<p>Знать: принципы разработки параллельных алгоритмов и программ;</p> <p>Уметь: применять к объектам стандартные модификаторы</p> <p>Владеть: навыками проектирования эффективных алгоритмов параллельных вычислений для решения прикладных задач.</p>	

	ПК-4.3. Имеет практический опыт разработки интеграции информационных систем.	Знать: основные алгоритмы параллельной обработки данных; Уметь: написать программы на языках программирования, поддерживающих распараллеливание; Владеть: навыками	
--	---	--	--

#### 4. Объем, структура и содержание дисциплины.

4.1. Объем дисциплины составляет 2 зачетные единицы, 72 академических часа.

4.2. Структура дисциплины.

№ п/п	Раздел дисциплины	Семестр	Неделя семестра	Виды учебной работы, включая самостоятельную работу студентов и трудоемкость (в часах)						Формы текущего контроля успеваемости (по неделям семестра) Форма промежуточной аттестации (по семестрам)
				Всего	Лекции	Практические занятия	Лабораторные занятия	Сам. работа	Контроль самост. работ.	
<b>Модуль 1. Принципы построения параллельных вычислительных систем</b>										
1	Тема 1. Введение в предмет. Основные понятия и определения.	8	1	10	2			12		Опрос
2	Тема 2. Принципы построения параллельных вычислительных систем.	8	2	10	2			12		Опрос
3	Тема 3. Классификация параллельных компьютеров и систем.	8	3	10	2			12		Прием самостоятельных работ
	<i>Итого по модулю 1:</i>			30	6			36		Контрольная работа
<b>Модуль 2. Технология OpenMP</b>										
1	Тема 1. Технология OpenMP	8	4	12	2			14		Опрос
2	Тема 2. Параллельные и последовательные области	8	5	10	2			12		Прием самостоя-

									ятельных работ
3	Тема 3. Распределение работы	8	6	19	2			21	Прием самостоятельных работ
4	Тема 4. Синхронизация	8	7	10	2			12	Прием самостоятельных работ
	<i>Итого по модулю 2:</i>			51	8			59	Зачет
	<b>ИТОГО</b>			81	14			95	

### 4.3. Содержание дисциплины, структурированное по темам (разделам).

#### 4.3.1. Содержание лекционных занятий по дисциплине

#### **Модуль 1. Принципы построения параллельных вычислительных систем**

##### **Тема 1.** Введение в предмет. Основные понятия и определения.

Понятие параллельных вычислений. Необходимость параллельных вычислений. Сдерживающие факторы. Характеристика необходимых знаний и умений.

##### **Тема 2.** Принципы построения параллельных вычислительных систем.

Основные виды ЭВМ. Принципы фон-Неймана. Структура традиционных ЭВМ. Способы повышения производительности компьютеров.

##### **Тема 3.** Классификация параллельных компьютеров и систем

Классификация Флинна, Хокни, Шнайдера. Взаимосвязь классификаций. Параллельные компьютеры с общей и разделенной памятью. Параллельные компьютеры с сетевой структурой. Характеристика типовых схем коммуникации в многопроцессорных вычислительных системах.

#### **Модуль 2. Технология OpenMP**

##### **Тема 1.** Технология OpenMP

Компиляция программы. Модель параллельной программы. Директивы и функции. Выполнение программы. Замер времени.

##### **Тема 2.** Параллельные и последовательные области.

Директива parallel. Переменные среды и вспомогательные функции. Директива single. Директива master.

##### **Тема 3.** Распределение работы

Параллельные циклы. Параллельные секции. Директива workshare. Задачи (tasks).

##### **Тема 4.** Синхронизация

Барьер. Директива ordered. Критические секции. Директива atomic.

4.3.2. Практические и лабораторные занятия по плану отсутствуют.

## 5. Образовательные технологии

Процесс изложения учебного материала сопровождается презентациями и демонстрацией решения задач в интерактивном режиме с использованием мультимедийного проектора.

Систематически проводится сравнение реализации потоков и нитей средствами различных языков Delphi, C# и др. Задания, предлагаемые студентам по различным темам, индивидуализированы, рекомендации по их выполнению, а также по исправлению ошибок, даются как непосредственно на лабораторных занятиях, так и по электронной почте.

Часть рекомендаций и указаний к решению заданий, которая относится ко всей учебной группе, размещается на сайте кафедры. Предусмотрено ведение образовательного блога.

Общение студентов по электронной почте с преподавателями кафедры носит регулярный характер, практикуется (в эпизодической форме) и общение с выпускниками кафедры – представителями российских и зарубежных компаний.

## 6. Учебно-методическое обеспечение самостоятельной работы студентов.

### 6.1 Виды самостоятельной работы и её контроля

Самостоятельная работа студентов складывается из:

- проработки лекционного материала (настоятельно рекомендуется самостоятельное практическое решение всех разобранных на лекциях упражнений);
- изучения рекомендованной литературы и материалов соответствующих форумов интернет;
- подготовки к сдаче промежуточных форм контроля (контрольных работ).
- 

№	Вид самостоятельной работы	Вид контроля	Учебно-методическое обеспечение
1.	Проработка лекционного материала	Контрольный фронтальный опрос	См. разделы 7.3, 8, 9 данного документа
2.	Изучение реко-	Контрольный фронталь-	См. разделы 7.3, 8, 9

	мендованной литературы и материалов соответствующих форумов интернет	ный опрос, прием и представление рефератов.	данного документа
3.	Подготовка к сдаче промежуточных форм контроля	Контрольные работы по каждому модулю.	См. разделы 7.3, 8, 9 данного документа

## **7. Фонд оценочных средств для проведения текущего контроля успеваемости, промежуточной аттестации по итогам освоения дисциплины.**

### **7.1 Типовые контрольные задания**

#### **Примерная тематика проверочных заданий к лекциям.**

##### Проверочные вопросы к модулю 1

1. Архитектура компьютеров с общей памятью.
2. Архитектура многоядерных процессоров.
3. Архитектура компьютеров с общей памятью. Вычислительные кластеры.
4. Метакомпьютинг, грид-технологии, облачные вычисления
5. Пиковая и реальная производительность компьютеров.
6. Формирования классов параллельных вычислительных систем.

##### Проверочные вопросы к модулю 2

1. Технологии параллельного программирования: эффективность, продуктивность, переносимость.
2. Технологии параллельного программирования MPI
3. Асинхронных вычислений
4. Синхронные вычисления
5. Рекуррентные формулы

##### Практическое задание

1. Создание простейших параллельных программ, компиляция и запуск параллельных программ
2. Создание параллельных программ с использованием виртуальных топологий.
3. Создание параллельных программ алгоритма умножения матрицы на вектор
4. Создание параллельных программ для решения систем линейных алгебраических уравнений методом простой итерации в двух вариантах

### 7.1.1 Вопросы для самостоятельной работы

1. Основные проблемы использования параллельной обработки данных.
2. Закон Мура о росте производительности последовательных компьютеров.
3. Модели многопроцессорных систем с общей и распределенной памятью. Модель конвейерной системы.
4. Основные проблемы параллельных вычислений: синхронизация, взаимное исключение, блокировка (тупики).
5. Распараллеливание алгоритмов: параллелизм данных и параллелизм кода;
6. Параллельные алгоритмы умножения вектора на матрицу.
7. Алгоритм исключения элементов матрицы (исключение Гаусса).
8. Параллельная реализация алгоритмов умножения двух матриц.
9. Алгоритмы маршрутизации.
- 10.. Основные возможности системы DVM.

### 7.1.2. Типовые контрольные задания

#### Задание 1.

Напишите программу, в которой создается  $k$  нитей, и каждая нить выводит на экран свой номер и общее количество нитей в параллельной области в формате:

```
I am <Номер нити> thread from <Количество нитей> threads!
```

#### Задание 2.

Изучите конструкции для управления работой с данными `shared` и `private`. Напишите программу, в которой создается  $k$  нитей, и каждая нить выводит на экран свой номер через переменную `rank` следующим образом:

```
rank = omp_get_thread_num();  
printf("I am %d thread.\n", rank);
```

Экспериментами определите, общей или частной должна быть переменная `rank`.

#### Задание 3.

Напишите программу, в которой две нити параллельно вычисляют сумму чисел от 1 до  $N$ . Распределите работу по нитям с помощью оператора `if` языка C. Для сложения результатов вычисления нитей воспользуйтесь OpenMP-параметром `reduction`.

#### Задание 4.

Изучите OpenMP-директиву параллельного выполнения цикла for. Напишите программу, в которой k нитей параллельно вычисляют сумму чисел от 1 до N. Распределите работу по нитям с помощью OpenMP-директивы for.

### Задание 5.

Изучите параметр schedule директивы for. Модифицируйте программу «Сумма чисел» из задания 4 таким образом, чтобы дополнительно выводилось на экран сообщение о том, какая нить, какую итерацию цикла выполняет:

[<Номер нити>]: calculation of the iteration number <Номер итерации>.

Задайте  $k = 4$ ,  $N = 10$ .

### Задание 6.

Напишите OpenMP-программу, которая вычисляет число  $\pi$  с точностью до N знаков после запятой. Используйте следующую формулу:

$$\pi = \left( \frac{4}{1+x_0^2} + \frac{4}{1+x_1^2} + \dots + \frac{4}{1+x_{N-1}^2} \right) \times \frac{1}{N}, \text{ где } x_i = (i+0.5) \times \frac{1}{N}, i = \overline{0, N-1}$$

Распределите работу по нитям с помощью OpenMP-директивы for.

### Задание 7.

Напишите OpenMP-программу, которая вычисляет произведение двух квадратных матриц  $A \times B = C$  размера  $n \times n$ . Используйте следующую формулу:

$$A = \begin{pmatrix} a_{11} & a_{12} & a_{13} & \dots & a_{1n} \\ a_{21} & a_{22} & a_{23} & \dots & a_{2n} \\ \cdot & \cdot & \cdot & \dots & \cdot \\ a_{n1} & a_{n2} & a_{n3} & \dots & a_{nn} \end{pmatrix} \quad B = \begin{pmatrix} b_{11} & b_{12} & b_{13} & \dots & b_{1n} \\ b_{21} & b_{22} & b_{23} & \dots & b_{2n} \\ \cdot & \cdot & \cdot & \dots & \cdot \\ b_{n1} & b_{n2} & b_{n3} & \dots & b_{nn} \end{pmatrix}$$

$$c_{im} = \sum_{j=1}^n a_{ij} \cdot b_{jm}; i = 1, 2, \dots, n; m = 1, 2, \dots, n$$

$$C = \begin{pmatrix} \sum_{j=1}^n a_{1j} \cdot b_{j1} & \sum_{j=1}^n a_{1j} \cdot b_{j2} & \sum_{j=1}^n a_{1j} \cdot b_{j3} & \dots & \sum_{j=1}^n a_{1j} \cdot b_{jn} \\ \sum_{j=1}^n a_{2j} \cdot b_{j1} & \sum_{j=1}^n a_{2j} \cdot b_{j2} & \sum_{j=1}^n a_{2j} \cdot b_{j3} & \dots & \sum_{j=1}^n a_{2j} \cdot b_{jn} \\ \dots & \dots & \dots & \dots & \dots \\ \sum_{j=1}^n a_{nj} \cdot b_{j1} & \sum_{j=1}^n a_{nj} \cdot b_{j2} & \sum_{j=1}^n a_{nj} \cdot b_{j3} & \dots & \sum_{j=1}^n a_{nj} \cdot b_{jn} \end{pmatrix}$$

### Задание 8.

Изучите OpenMP-директивы создания параллельных секций `sections` и `section`. Напишите программу, содержащую 3 параллельные секции, внутри каждой из которых должно выводиться сообщение:

```
[<Номер нити>]: came in section <Номер секции>
```

Вне секций внутри параллельной области должно выводиться следующее сообщение:

```
[<Номер нити>]: parallel region
```

Запустите приложение на 2-х, 3-х, 4-х нитях. Проследите, как нити распределяются по параллельным секциям.

### Задание 9.

Перепишите программу, в которой параллельно вычисляется сумма чисел от 1

до N (см. задание 4), без использования параметра `reduction`. Вместо параметра `reduction` используйте директиву `atomic`.

### Задание 10.

Перепишите параллельную программу вычисления числа  $\pi$  (см. задание 6) без использования параметра `reduction`. Вместо параметра `reduction` используйте директиву `critical`.

#### 7.1.3. Темы для рефератов

1. Сравнение архитектуры CPU и GPU
2. Эволюция GPU
3. Использование нескольких GPU

4. Современные направления развития параллельных вычислительных систем
5. Создание потоков и нитей в Delphi
6. Средства распараллеливания в современных языках программирования
7. Вытесняющая мультизадачность
8. Алгоритмы исключения тупиковых ситуаций
9. Мультизадачность в Windows 10.0
10. Отечественные супер-эвм

#### **7.1.4. Контрольные вопросы к лекциям**

1. Архитектура компьютеров с общей памятью.
2. Архитектура многоядерных процессоров.
3. Архитектура компьютеров с общей памятью. Вычислительные кластеры.
4. Метакомпьютинг, грид-технологии, облачные вычисления
5. Пиковая и реальная производительность компьютеров.
6. Формирования классов параллельных вычислительных систем.
7. Технологии параллельного программирования: эффективность, продуктивность, переносимость.
8. Технологии параллельного программирования MPI
9. Асинхронные вычисления
10. Синхронные вычисления
11. Рекуррентные формулы
12. Создание простейших параллельных программ, компиляция и запуск параллельных программ
13. Создание параллельных программ с использованием виртуальных топологий.
14. Создание параллельных программ алгоритма умножения матрицы на вектор
15. Создание параллельных программ для решения систем линейных алгебраических уравнений методом простой итерации в двух вариантах

#### **7.2. Методические материалы, определяющие процедуру оценивания знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций.**

Общий результат выводится как интегральная оценка, складывающаяся из текущего контроля - 50% и промежуточного контроля - 50%.

Текущий контроль по дисциплине включает:

- посещение занятий - 30 баллов,
- выполнение лабораторных заданий – 20 баллов,
- выполнение домашних (аудиторных) контрольных работ - 50 баллов.

Промежуточный контроль по дисциплине включает:

- устный опрос - 30 баллов,
- письменная контрольная работа - 50 баллов,
- тестирование - 20 баллов.

#### **8. Перечень основной и дополнительной учебной литературы, необходимой для освоения дисциплины.**

а) основная литература:

1. Биллиг В.А. Параллельные вычисления и многопоточное программирование [Электронный ресурс] / В.А. Биллиг. — 2-е изд. — Электрон. текстовые данные. — М.: Интернет Университет Информационных Технологий (ИНТУИТ), 2016. — 310 с. — 2227-8397. — Режим доступа: <http://www.iprbookshop.ru/73705.html>
2. Боресков А.В. и др. Параллельные вычисления на GPU. Архитектура и программная модель CUDA. Учеб. пособие. 2-е изд. М.: Издательство Московского университета, 2015. — 336с.
3. Антонов А.С. Технологии параллельного программирования MPI и OpenMP: Учебное пособие. М.: Изд-во МГУ, 2012. — 344с.

б) дополнительная литература:

1. Косяков М.С. Введение в распределенные вычисления. — СПб: НИУ ИТМО, 2014. — 155 с.
2. Барский А.Б. Параллельные информационные технологии [Электронный ресурс] : учебное пособие / А.Б. Барский. — Электрон. текстовые данные. — Москва, Саратов: Интернет-Университет Информационных Технологий (ИНТУИТ), Вузовское образование, 2017. — 503 с. — 978-5-4487-0087-3. — Режим доступа: <http://www.iprbookshop.ru/67379.html>
3. Левин М.П. Параллельное программирование с использованием OpenMP [Электронный ресурс] / М.П. Левин. — Электрон. текстовые данные. — М.: Интернет Университет Информационных Технологий (ИНТУИТ), 2016. — 133 с. — 978-5-94774-857-4. — Режим доступа: <http://www.iprbookshop.ru/52216.html>

## **9. Перечень ресурсов информационно-телекоммуникационной сети «Интернет», полезных для освоения дисциплины**

1. Поток выполнения [Электронный ресурс] // URL = <https://ru.wikipedia.org/wiki/> /
2. Операционная платформа. Определения и классификация [Электронный ресурс] // URL = <http://gigabaza.ru/doc/33138.html> /
3. Процессы и потоки in-depth. Обзор различных потоковых моделей [Электронный ресурс] // URL = <http://habrahabr.ru/post/40227/>

4. Автоматическое распараллеливание программ для распределенных систем. Статическое построение расширенного графа управления [Электронный ресурс] // URL = <http://xreferat.com/33/4673-1/>.

5. <http://www.intuit.ru> [Электронный ресурс]

6. <http://www.parallel.ru> Материалы на сайте Лаборатории параллельных информационных технологий МГУ.

#### **10. Методические указания для обучающихся по освоению дисциплины.**

1. При решении лабораторных заданий программистский подход непременно должен присутствовать (без него решение не будет полноценным), однако, он не должен заслонять сугубо математические (доказательство и др.) и алгоритмические (построение, оптимизация, верификация и др.) аспекты.

2. Необходимо обратить внимание на распознавание ситуаций, когда распараллеливание: а) допустимо, б) целесообразно, в) необходимо. Нельзя игнорировать «накладные расходы» ресурсов, выделяемых собственно распараллеливанию вычислений.

3. Важно различать архитектурные и теоретические проблемы распараллеливания.

4. При решении проблемы автоматического распараллеливания особое внимание следует уделить созданию внутреннего представления программы, органично соответствующего проблематике решаемой задачи.

#### **11. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине, включая перечень программного обеспечения и информационных справочных систем.**

При осуществлении образовательного процесса студентами и профессорско-преподавательским составом используются следующее программное обеспечение: Microsoft Visual Studio Express, Microsoft Windows, Ubuntu Linux, Skype. Также студентам предоставляется доступ к российским и международным электронным библиотекам через компьютеры университета.

#### **12. Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине.**

Имеется необходимая литература в библиотеке, медиа проектор и компьютер для проведения лекций-презентаций.

Лабораторные занятия проводятся в компьютерных классах с необходимым программным обеспечением.

Вся основная литература предоставляется студенту в электронном формате.