

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«КУБАНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»
Факультет физико-технический



УТВЕРЖДАЮ:

Проректор по учебной работе,
качеству образования – первый
проректор

Т.А. Хагуров

«30» мая 2025 г.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

Б1.В.ДВ.01.01.01 ОСНОВЫ ПАРАЛЛЕЛЬНОГО ПРОГРАММИРОВАНИЯ

Направление подготовки **09.03.02 Информационные системы и технологии**

Направленность (профиль) **Аналитические информационные системы**

Форма обучения **заочная**

Квалификация **бакалавр**

Краснодар 2025

Рабочая программа дисциплины Б1.В.ДВ.01.01.01 Основы параллельного программирования составлена в соответствии с федеральным государственным образовательным стандартом высшего образования (ФГОС ВО) по направлению подготовки/ специальности 09.03.02 Информационные системы и технологии

Программу составил:

В.Н. Значко, ст. преподаватель. каф. теор. физики и комп. технологий

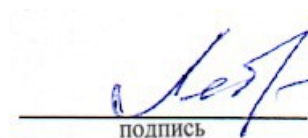


подпись

Рабочая программа дисциплины утверждена на заседании кафедры теоретической физики и компьютерных технологий

протокол № 9 «08» апреля 2025 г.

Заведующий кафедрой, доктор физико-математических наук, профессор Лебедев К.А

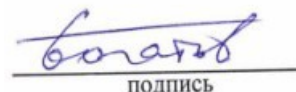


подпись

Утверждена на заседании учебно-методической комиссии физико-технического факультета

протокол № 11 от «21» апреля 2025 г.

Председатель УМК факультета Богатов Н.М.



подпись

Рецензенты:

М.С. Коваленко, кандидат физико-математических наук, доцент кафедры физики и информационных систем

Л.Р. Григорян, генеральный директор ООО НПФ «Мезон» кандидат физико-математических наук

1 Цели и задачи изучения дисциплины

1.1 Цель освоения дисциплины

Целью дисциплины является формирование у студентов знаний и навыков в области параллельного программирования, включая освоение методов разработки, отладки и оптимизации параллельных алгоритмов и программ для современных вычислительных систем.

1.2 Задачи дисциплины:

- Изучение теоретических основ параллельного программирования.
- Освоение инструментов и технологий для создания параллельных программ.
- Развитие навыков анализа и оптимизации параллельных алгоритмов.
- Применение полученных знаний для решения практических задач.

1.3 Место дисциплины в структуре образовательной программы

Данная дисциплина входит в часть по выбору блока Б1 "Дисциплины (модули)" части учебного плана, формируемой участниками образовательных отношений.

Изучение дисциплины базируется на знаниях, полученных при изучении дисциплин:

- Информатика и теория алгоритмов (Б1.О.15).
- Программирование на Python и анализ данных (Б1.О.16).
- Технологии программирования на C/C++ (Б1.О.18).

Дисциплина изучается в тесной взаимосвязи с учебным материалом других дисциплин и обеспечивает все виды практик, а также следующие дисциплины направления подготовки:

- Архитектура ЭВМ (Б1.В.01).
- Операционные системы (Б1.В.05).
- Теория систем и системная интеграция (Б1.В.ДВ.01.02.04).

1.4 Перечень планируемых результатов обучения по дисциплине, соотнесенных с планируемыми результатами освоения образовательной программы

Изучение данной учебной дисциплины направлено на формирование у обучающихся следующих компетенций:

Код и наименование индикатора*	Результаты обучения по дисциплине
ПК-6 Способность создания (модификации) и сопровождения информационных систем (ИС), автоматизирующих задачи организационного управления и бизнес-процессы в организациях различных форм собственности с целью повышения эффективности деятельности организаций - пользователей ИС	
ПК-4.1. зЗнать определение первоначальных требований заказчика к ИС и возможности их реализации в ИС на этапе предконтрактных работ	Студент знает принципы параллельного программирования, языки и инструменты для разработки параллельных приложений (MPI, OpenMP, CUDA, TBB).
ПК-4.2. Уметь осуществлять документирование существующих бизнеспроцессов организации заказчика (реверс-инжиниринг бизнес-процессов организации), выявлять и анализировать требования к ИС	Студент умеет разрабатывать, отлаживать и оптимизировать параллельные программы для решения практических задач (распараллеливание циклов, обработка данных, моделирование).
ПК-4.3. Иметь навыки адаптации бизнес-процессов заказчика к возможностям ИС	Студент владеет навыками работы с инструментами параллельного программирования, анализа производительности и устранения ошибок .
ПК-1 Способность проводить научные исследования при разработке, внедрении и сопровождении информационных технологий и систем на всех этапах жизненного цикла	
ПК-1.1 Знать информационные процессы, технологии, системы и сети, их инструментальное (программное, техническое, организационное) обеспечение, способы и методы проектирования.	Студент знает архитектуру параллельных систем (многопроцессорные системы, GPU, кластеры), методы проектирования параллельных алгоритмов.
ПК-1.2 Уметь проводить работы по обработке и анализу научно-технической информации и результатов исследований в области ИТиС.	Студент умеет анализировать эффективность параллельных алгоритмов, интерпретировать результаты профилирования и оптимизации.
ПК-1.3 Иметь навыки по эксплуатации информационных технологий и систем в различных областях и сферах цифровой экономики.	Студент владеет навыками применения параллельных вычислений в научных исследованиях, анализе данных и других прикладных областях.

2 Структура и содержание дисциплины

2.1 Распределение трудоёмкости дисциплины по видам работ

Общая трудоёмкость дисциплины составляет 3 зачетные единицы (108 часов), их распределение по видам работ представлено в таблице

Виды работ		Всего часов	Форма обучения		
			очная	очно-заочная	заочная
			3 семестр		
Контактная работа, в том числе:		8	8		
Аудиторные занятия (всего):		8	8		
Занятия лекционного типа		4	4		
Лабораторные занятия					
Практические занятия		4	4		
Семинарские занятия					
Иная контактная работа:					
Контроль самостоятельной работы (КСР)					
Промежуточная аттестация (ИКР)					
Самостоятельная работа, в том числе:		126	126		
<i>Самостоятельное изучение разделов, самоподготовка (проработка и повторение лекционного материала и материала учебников и учебных пособий, подготовка к коллоквиуму)</i>		122	122		
Подготовка к текущему контролю		4	4		
Контроль:		4	4		
Общая трудоёмкость	часы	144	144		
	в т.ч. контактная работа	8	8		
	зач. ед.	4	4		

2.2 Содержание дисциплины

Распределение видов учебной работы и их трудоемкости по разделам дисциплины. Разделы (темы) дисциплины, изучаемые в 3 семестре (на 2 курсе) (очная форма обучения).

№	Наименование раздела	Количество часов				
		Всего	Аудиторная работа			Внеаудиторная работа
			Л	ПЗ	ЛР	СРС
1	Введение в параллельное программирование	12	1	1		10
2	Модели параллельных вычислений	38	1	1		36
3	Инструменты и технологии параллельного программирования	38	1	1		36
4	Оптимизация параллельных алгоритмов	42	1	1		40
	Итого по разделам дисциплины	130	4	4		122

2.3 Содержание разделов (тем) дисциплины

2.3.1 Занятия лекционного типа

№	Наименование раздела	Содержание раздела (темы)	Форма текущего контроля
1	Введение в параллельное программирование	Основные понятия, преимущества и области применения параллельного программирования.	опрос
2	Модели параллельных вычислений	Классификация моделей (MPI, OpenMP, CUDA), их особенности и применение.	опрос
3	Инструменты и технологии параллельного программирования	Обзор инструментов (библиотеки, фреймворки) и их использование.	опрос
4	Оптимизация параллельных алгоритмов	Методы анализа и повышения эффективности параллельных программ.	опрос

2.3.2 Занятия лабораторного типа

№	Наименование раздела	Тематика занятий	Форма текущего контроля
1	Введение в параллельное программирование	Лабораторная работа №1. "Основы работы с потоками": создание и управление потоками в Python/C++. Лабораторная работа №2. "Анализ	отчёт

		производительности": сравнение последовательного и параллельного выполнения задач.	
2	Модели параллельных вычислений	Лабораторная работа №3. "Реализация алгоритма с использованием MPI": разработка простой параллельной программы. Лабораторная работа №4. "Использование OpenMP для распараллеливания циклов": оптимизация вычислений.	отчёт
3	Инструменты и технологии параллельного программирования	Лабораторная работа №5. "Работа с CUDA": написание программы для GPU. Лабораторная работа №6. "Применение библиотеки TBW": решение задачи с использованием Intel Threading Building Blocks.	отчёт
4	Оптимизация параллельных алгоритмов	Лабораторная работа №7. "Профилирование параллельных программ": выявление узких мест. Лабораторная работа №8. "Оптимизация обмена данными между процессами": уменьшение накладных расходов.	отчёт

Курсовые проекты не предусмотрены

2.4 Перечень учебно-методического обеспечения для самостоятельной работы обучающихся по дисциплине

№	Вид СРС	Перечень учебно-методического обеспечения дисциплины по выполнению самостоятельной работы
1	Самостоятельное изучение раздела, самоподготовка (проработка и повторение лекционного материала и материала учебников и учебных пособий, подготовка к лабораторным занятиям, коллоквиумам)	Воеводин В.В., Воеводин Вл.В. "Параллельные вычисления" (2-е изд.). – СПб.: БХВ-Петербург, 2022. Тормасов А.Г. "Лабораторный практикум по параллельным вычислениям" – Казань: КФУ, 2023.
2	Подготовка к текущему контролю	Богачёв К.Ю. "Основы параллельного программирования" – М.: ДМК Пресс, 2018. Гергель В.П., Стронгин Р.Г. "Основы параллельного программирования для многопроцессорных вычислительных систем" – М.: НИУ ВШЭ, 2015.

Учебно-методические материалы для самостоятельной работы обучающихся из числа инвалидов и лиц с ограниченными возможностями здоровья предоставляются в формах, адаптированных к ограничениям их здоровья и восприятия информации:

Для лиц с нарушениями зрения:

- в печатной форме увеличенным шрифтом,
- в форме электронного документа,
- в форме аудиофайла,
- в печатной форме на языке Брайля.

Для лиц с нарушениями слуха:

- в печатной форме,
- в форме электронного документа.

Для лиц с нарушениями опорно-двигательного аппарата:

- в печатной форме,
- в форме электронного документа,
- в форме аудиофайла.

Данный перечень может быть конкретизирован в зависимости от контингента обучающихся.

3. Образовательные технологии, применяемые при освоении дисциплины

Активные и интерактивные формы проведения занятий

- лекция-визуализация,
- лекция – пресс-конференция,
- метод малых групп.

Для лиц с ограниченными возможностями здоровья предусмотрена организация

консультаций с использованием электронной почты.

4 Оценочные средства для текущего контроля успеваемости и промежуточной аттестации.

4.1 Фонд оценочных средств для проведения текущего контроля

1. Что такое параллельное программирование?

- a) Программирование, при котором задачи выполняются строго последовательно.
- b) Программирование, при котором несколько задач выполняются одновременно на разных процессорах или ядрах.
- c) Программирование, использующее только один поток выполнения.
- d) Программирование, ориентированное исключительно на однопроцессорные системы.

**2. Какая модель параллельных вычислений использует общую память?

- a) MPI
- b) OpenMP
- c) CUDA
- d) MapReduce

**3. Что означает аббревиатура MPI?

- a) Multi-Process Integration
- b) Message Passing Interface
- c) Memory Parallel Interconnect
- d) Multi-Threading Protocol

**4. Какой инструмент используется для программирования GPU?

- a) OpenMP
- b) MPI
- c) CUDA
- d) TBB

**5. Что такое race condition?

- a) Ошибка, возникающая при неправильном распределении памяти.
- b) Конфликт, когда несколько потоков одновременно изменяют общие данные без синхронизации.

- c) Проблема, связанная с нехваткой вычислительных ресурсов.
- d) Ошибка компиляции в параллельных программах.

6. Какой директивой OpenMP можно распараллелить цикл?

- a) `#pragma omp parallel``
- b) `#pragma omp for``
- c) `#pragma omp single``
- d) `#pragma omp barrier``

**7. Что такое deadlock (взаимоблокировка)?

- a) Ошибка, при которой программа завершается аварийно.
- b) Ситуация, когда два или более потока бесконечно ждут друг друга.
- c) Утечка памяти в параллельной программе.
- d) Ошибка при компиляции кода.

**8. Какая библиотека Intel предназначена для параллельного программирования?

- a) OpenCL
- b) MPI
- c) TBB (Threading Building Blocks)
- d) Pthreads

**9. Какой метод помогает избежать ложного разделения кэша (false sharing)?

- a) Уменьшение количества потоков.
- b) Выравнивание данных или добавление padding.
- c) Увеличение размера кэша.
- d) Использование только одного ядра.

**10. Какой параметр OpenMP ограничивает количество потоков?

- a) `omp_set_num_cores``
- b) `omp_set_thread_limit``
- c) `omp_set_num_threads``
- d) `omp_set_max_parallel``

**11. Какой алгоритм часто используется в MapReduce?

- a) Быстрая сортировка (QuickSort)

- b) Сортировка слиянием (MergeSort)
- c) Разделяй и властвуй (Divide and Conquer)
- d) Жадные алгоритмы (Greedy)

12. Что такое granularity (гранулярность) в параллельном программировании?

- a) Размер блока данных, обрабатываемого за одну операцию.
- b) Количество потоков в программе.
- c) Скорость выполнения параллельного кода.
- d) Объем памяти, выделяемой под потоки.

13. Какой механизм синхронизации предотвращает одновременный доступ к общим данным?

- a) `#pragma omp parallel`
- b) `#pragma omp critical`
- c) `#pragma omp single`
- d) `#pragma omp master`

**14. Какой инструмент используется для профилирования параллельных программ?

- a) Valgrind
- b) GDB
- c) Intel VTune Profiler
- d) Clang Static Analyzer

**15. Какой подход используется в CUDA для организации параллельных вычислений?

- a) Разделение на процессы (MPI-подход).
- b) Использование блоков (blocks) и потоков (threads).
- c) Только глобальная синхронизация.
- d) Последовательное выполнение ядер (kernels).

4.2 Фонд оценочных средств для проведения промежуточной аттестации

Зачетно-экзаменационные материалы для промежуточной аттестации (экзамен)

1. Что такое параллельное программирование?
2. Назовите основные преимущества параллельных вычислений.
3. Какие модели параллельных вычислений вы знаете?
4. В чём разница между MPI и OpenMP?
5. Как работает механизм разделяемой памяти в OpenMP?
6. Какие инструменты используются для разработки параллельных программ?
7. Что такое race condition и как его избежать?
8. Как измеряется производительность параллельных программ?
9. Какие методы оптимизации параллельных алгоритмов существуют?
10. Приведите пример задачи, где параллельные вычисления дают значительный выигрыш.
11. Как организовать обмен данными между процессами в MPI?
12. Что такое deadlock и как его предотвратить?
13. Какие библиотеки для параллельного программирования вы знаете?
14. Как распараллелить цикл с помощью OpenMP?
15. Что такое асинхронное программирование?
16. Какие проблемы могут возникнуть при использовании GPU для вычислений?
17. Как провести профилирование параллельной программы?
18. Какие существуют стратегии распределения нагрузки между потоками?
19. Как избежать ложного разделения кэша (false sharing)?
20. Что такое granularity в параллельном программировании?
21. Как организовать синхронизацию потоков?
22. Какие существуют способы уменьшения накладных расходов в MPI?
23. Как использовать CUDA для ускорения вычислений?
24. Какие существуют паттерны параллельного программирования?
25. Как оценить эффективность параллельного алгоритма?
26. Какие инструменты используются для отладки параллельных программ?
27. Как избежать взаимоблокировок (deadlocks) в многопоточных программах?
28. Какие существуют методы балансировки нагрузки?
29. Как организовать параллельную обработку данных в реальном времени?
30. Какие существуют ограничения у параллельных вычислений?

31. Как использовать ТВВ для параллельного программирования?
32. Какие существуют подходы к распараллеливанию рекурсивных алгоритмов?
33. Как организовать параллельную обработку больших данных?
34. Какие существуют методы уменьшения задержек в распределённых системах?
35. Как использовать MapReduce для параллельной обработки данных?
36. Какие существуют стандарты параллельного программирования?
37. Как организовать параллельное выполнение задач с зависимостями?
38. Какие существуют методы оптимизации работы с памятью в GPU?
39. Как использовать параллельные вычисления в машинном обучении?
40. Какие перспективы развития параллельного программирования вы видите?

Критерии оценивания результатов обучения

Оценочные средства для инвалидов и лиц с ограниченными возможностями здоровья выбираются с учетом их индивидуальных психофизических особенностей.

- при необходимости инвалидам и лицам с ограниченными возможностями здоровья предоставляется дополнительное время для подготовки ответа на экзамене;

- при проведении процедуры оценивания результатов обучения инвалидов и лиц с ограниченными возможностями здоровья предусматривается использование технических средств, необходимых им в связи с их индивидуальными особенностями;

- при необходимости для обучающихся с ограниченными возможностями здоровья и инвалидов процедура оценивания результатов обучения по дисциплине может проводиться в несколько этапов.

Процедура оценивания результатов обучения инвалидов и лиц с ограниченными возможностями здоровья по дисциплине (модулю) предусматривает предоставление информации в формах, адаптированных к ограничениям их здоровья и восприятия информации:

Для лиц с нарушениями зрения:

- в печатной форме увеличенным шрифтом,
- в форме электронного документа.

Для лиц с нарушениями слуха:

- в печатной форме,
- в форме электронного документа.

Для лиц с нарушениями опорно-двигательного аппарата:

- в печатной форме,
- в форме электронного документа.

Данный перечень может быть конкретизирован в зависимости от контингента обучающихся.

5 Перечень основной и дополнительной учебной литературы, необходимой для освоения дисциплины (модуля).

5.1 Основная литература:

1. Воеводин В.В., Воеводин Вл.В. "Параллельные вычисления" (2-е изд.). – СПб.: БХВ-Петербург, 2022.
2. Богачёв К.Ю. "Основы параллельного программирования" – М.: ДМК Пресс, 2018.
3. Гергель В.П., Стронгин Р.Г. "Основы параллельного программирования для многопроцессорных вычислительных систем" – М.: НИУ ВШЭ, 2015.

5.2 Дополнительная литература:

1. Антонов А.С. "Параллельное программирование с использованием OpenMP" – М.: МГУ, 2016.
2. Корнеев В.Д. "Параллельное программирование в MPI" – Новосибирск: Изд-во СО РАН, 2013.
3. Бахтин В.А., Мееров И.Б. "Программирование графических процессоров с CUDA" – М.: МГТУ им. Баумана, 2020.
4. Сысоев А.В. "Высокопроизводительные вычисления на GPU" – СПб.: Университет ИТМО, 2019.
5. Лупиков С.Г. "Методы распараллеливания алгоритмов" – М.: Физматлит, 2017.
6. Немнюгин С.А., Стесик О.Л. "Параллельное программирование для многоядерных процессоров" – СПб.: Питер, 2014.
7. Тормасов А.Г. "Лабораторный практикум по параллельным вычислениям" – Казань: КФУ, 2023

5.3 Интернет-ресурсы

1. БД Web of Science - главный ресурс для исследователей по поиску и анализу научной литературы, охватывающей около 18000 научных журналов со всего мира. База данных международных индексов научного цитирования <http://webofscience.com/>
2. zbMATH - полная математическая база данных. Охватывает материалы с конца 19 века. zbMATH содержит около 4000000 документов из более

- 3000 журналов и 170000 книг по математике, статистике, информатике.
<https://zbmath.org/>
3. БД Kaggle - это платформа для сбора и обработки данных. Является онлайн площадкой для научного моделирования. <https://www.kaggle.com>.
 4. База данных Научной электронной библиотеки eLIBRARY.RU
<https://elibrary.ru/>
 5. База данных Всероссийского института научной и технической информации (ВИНИТИ) РАН <http://www2.viniti.ru/>
 6. «ЭЛЕКТРОННАЯ БИБЛИОТЕКА ДИССЕРТАЦИЙ» Российской Государственной Библиотеки (РГБ) – в настоящее время ЭБД содержит более 800 000 полных текстов диссертаций. <https://dvs.rsl.ru>
 7. Портал открытых данных Российской Федерации <https://data.gov.ru>
 8. База открытых данных Министерства защиты РФ
<https://rosmintrud.ru/opendata>
 9. Федеральный портал единое окно доступа к информационным ресурсам -
<http://window.edu.ru>.
 10. Российский фонд фундаментальных исследований предоставляет доступ к информационным наукометрическим базам данных и полнотекстовым научным ресурсам издательств Springer Nature и Elsevier -
<http://www.rfbr.ru/rffi/ru>
 11. Федеральный портал "Информационно-коммуникационные технологии в образовании" - <http://www.ict.edu.ru/>
 12. «Лекториум ТВ» – видеолекции ведущих лекторов России. Лекториум – on-line – библиотека, где ВУЗы и известные лектории России презентуют своих лучших лекторов. Доступ к материалам свободный и бесплатный -
<http://www.lektorium.tv>.

6. Методические указания для обучающихся по освоению дисциплины

Общие рекомендации

1. Начинать изучение дисциплины следует с проработки рабочей программы (цели, задачи, структура курса).
2. Подготовка к лекциям:
 - Повторение конспекта предыдущей лекции.
 - Предварительный просмотр материала новой темы.
3. Конспектирование:
 - Фиксация ключевых положений, выводов, формулировок (не дословно).

Семинарские занятия

Цель: закрепление материала, развитие навыков публичных выступлений, дискуссий.

Этапы:

1. Вступительное слово преподавателя.
2. Доклады студентов.
3. Обсуждение и альтернативные мнения.
4. Итоги и оценки.

Текущий контроль: устные опросы, письменные работы.

Индивидуальные консультации

- Особенно важны для инвалидов и лиц с ОВЗ.
- Формы: разъяснение материала, помощь в адаптации заданий.

Методические указания к написанию рефератов и докладов

Цель реферата: навыки поиска литературы, анализа информации, грамотного оформления.

Требования:

Объем: 15–25 страниц.

Структура: титульный лист, содержание, введение, основная часть, заключение, список литературы (не менее 5 источников).

Оформление: шрифт 12–14 pt, интервал 1.5, поля 20–25 мм.

Защита реферата:

- Устный доклад (8–10 минут) с презентацией (8–10 слайдов).

- Критерии оценки: соответствие теме, глубина проработки, культура изложения.

Требования к докладу

- Грамотность, ссылки на источники, мультимедийная презентация.
- Время выступления: 8–10 минут.

Процедура проверки

1. Предварительная проверка реферата (антиплагиат) за 3 дня до защиты.
2. Защита на семинаре (оценка: «отлично» – «неудовлетворительно»).

7. Материально-техническое обеспечение по дисциплине (модулю)

Для проведения лекционных и семинарских занятий используется мультимедийный проектор и ноутбук.

Наименование специальных помещений	Оснащенность специальных помещений	Перечень лицензионного программного обеспечения
Учебные аудитории для проведения занятий лекционного типа	Мебель: учебная мебель. Технические средства обучения: экран, проектор, компьютер.	Microsoft Windows; Microsoft Office
Учебные аудитории для проведения групповых и индивидуальных консультаций, семинарских занятий, текущего контроля и промежуточной аттестации	Мебель: учебная мебель. Технические средства обучения: экран, проектор, компьютер/ноутбук.	Microsoft Windows; Microsoft Office

Для самостоятельной работы обучающихся предусмотрены помещения, укомплектованные специализированной мебелью, оснащенные компьютерной техникой с возможностью подключения к сети «Интернет» и обеспечением доступа в электронную информационно-образовательную среду университета.

Наименование помещений для самостоятельной работы обучающихся	Оснащенность помещений для самостоятельной работы обучающихся	Перечень лицензионного программного обеспечения
Помещение для самостоятельной работы обучающихся (читальный зал Научной библиотеки)	Мебель: учебная мебель. Комплект специализированной мебели: компьютерные столы. Оборудование: компьютерная техника с подключением к информационно-коммуникационной сети «Интернет» и доступом в электронную информационно-образовательную среду образовательной организации, веб-камеры, коммуникационное оборудование, обеспечивающее доступ к сети интернет (проводное	Microsoft Windows; Microsoft Office.

	соединение и беспроводное соединение по технологии Wi-Fi).	
Помещение для самостоятельной работы обучающихся	Мебель: учебная мебель. Комплект специализированной мебели: компьютерные столы. Оборудование: компьютерная техника с подключением к информационно-коммуникационной сети «Интернет» и доступом в электронную информационно-образовательную среду образовательной организации, веб-камеры, коммуникационное оборудование, обеспечивающее доступ к сети интернет (проводное соединение и беспроводное соединение по технологии Wi-Fi).	Microsoft Windows; Microsoft Office.