МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное бюджетное образовательное учреждение
высшего образования
МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ГЕОДЕЗИИ И КАРТОГРАФИИ
(МИИГАиК)

Факультет геоинформатики и информационной безопасности
Кафедра информатики и геоинформационных технологий

**ДОКУМЕНТ ПОДПИСАН
ЭЛЕКТРОННОЙ ПОДПИСЬЮ**

Сертификат: 5BFEE1D286E9C95A0C0D2C611E434BD5
Владелец: Степанченко Алексей Леонидович
Действителен: с 23.09.2026 по 17.12.2027

«УТВЕРЖДАЮ»
Проректор
А. Л. Степанченко
18 апреля 2025 г.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ (МОДУЛЯ) «Б1.В.04 ЯЗЫКИ ПРОГРАММИРОВАНИЯ ДЛЯ РАБОТЫ С БОЛЬШИМИ ДАННЫМИ»

Уровень высшего образования: магистратура

Направление подготовки: 09.04.03 Прикладная информатика

Направленность (профиль) подготовки: Анализ больших пространственных данных

Квалификация (степень) выпускника: магистр

Форма обучения: очная

Год набора (приема на обучение): 2025

Срок получения образования: 2 года

Объем: в зачетных единицах: 3 з.е.
в академических часах: 108 ак.ч.

Москва, 2025

Разработчики:
Старший преподаватель кафедры Информатики и геоинформационных технологий Стрельцов А. С.

Рабочая программа дисциплины (модуля) составлена в соответствии с требованиями ФГОС ВО по направлению подготовки 09.04.03 Прикладная информатика, утвержденного приказом Минобрнауки от 19.09.2017 № 916

Согласование и утверждение

№	Подразделение	Ответственное лицо	ФИО	Виза	Дата, протокол (при наличии)
1	факультет Геоинформатики и информационной безопасности	Председатель ученого совета факультета	Матерухин А. В.	Рассмотрено	14.10.2024, № 10-24

Содержание

1. Цель и задачи освоения дисциплины (модуля)

2. Планируемые результаты обучения по дисциплине (модулю), соотнесенные с планируемыми результатами освоения образовательной программы

3. Место дисциплины в структуре ОП

4. Объем дисциплины и виды учебной работы

5. Содержание дисциплины
   5.1. Разделы, темы дисциплины и виды занятий
   5.2. Содержание разделов, тем дисциплины и формы текущего контроля

6. Оценочные материалы текущего контроля

7. Оценочные материалы промежуточной аттестации

8. Порядок проведения промежуточной аттестации

9. Материально-техническое и учебно-методическое обеспечение дисциплины
   9.1. Перечень основной и дополнительной учебной литературы
   9.2. Профессиональные базы данных и ресурсы «Интернет», к которым обеспечивается доступ обучающихся
   9.3. Программное обеспечение и информационно-справочные системы, используемые при осуществлении образовательного процесса по дисциплине
   9.4. Оборудование и технические средства обучения

10. Методические указания по освоению дисциплины (модуля)

1. Цель и задачи освоения дисциплины (модуля)

Цель освоения дисциплины - формирование у обучающихся комплексных компетенций в области применения современных языков программирования и технологий распределенных вычислений для эффективной обработки, анализа и визуализации больших объемов структурированных и пространственных данных в профессиональной деятельности.

Задачи изучения дисциплины:

- Освоение основ программирования на языке Python и специализированных библиотек для работы с большими данными, включая технологии параллельных и распределенных вычислений.;
- Изучение методов и инструментов обработки больших объемов табличных, векторных и растровых пространственных данных с использованием современных фреймворков и облачных платформ.;
- Формирование навыков оптимизации вычислительных процессов, масштабирования решений и интеграции различных технологий для построения эффективных систем обработки больших данных..

2. Планируемые результаты обучения по дисциплине (модулю), соотнесенные с планируемыми результатами освоения образовательной программы

Компетенции, индикаторы и результаты обучения

ПК-1 Способен выполнять работы по сбору, обработке и анализу больших пространственных данных

ПК-1.1 Собирает большие пространственные данные

Знать:

ПК-1.1/Зн1 Методы и средства поиска и получения, систематизации и обработки аэрокосмической информации и снимков местности

Уметь:

ПК-1.1/Ум1 Планировать процессы и этапы обработки(получения) данных фотографической, оптико-электронной и других видов аэрокосмических съемок для решения определенного круга производственных и научных задач

Владеть:

ПК-1.1/Нв1 Методами планирования аэросъемочного процесса для получения изображений с заданными параметрами

ПК-1.2 Обрабатывает большие пространственные данные

Знать:

ПК-1.2/Зн1 Основы и физические принципы регистрации аэрокосмической информации о местности

ПК-1.2/Зн2 Технологии и алгоритмы сбора, обработки и анализа больших данных о Земле

Уметь:

ПК-1.2/Ум1 Осуществлять выбор аэрокосмических снимков, получаемых различными системами и комплексами дистанционного зондирования, в зависимости от решаемых практических или исследовательских задач

ПК-1.2/Ум2 Применять типовые технологии и алгоритмы сбора, обработки и анализа больших данных о Земле для решения практических задач

Владеть:

ПК-1.2/Нв1 Методами сбора и оценки качества пространственной информации, имеющей значения для подготовки и реализации проектов в соответствующих сферах профессиональной деятельности

ПК-1.2/Нв2 Навыками использования типовых технологий и алгоритмов сбора, обработки и анализа больших данных о Земле для решения практических задач

ПК-1.3 Анализирует большие пространственные данные

Знать:

ПК-1.3/Зн1 Основные виды и способы дистанционного зондирования и характеристики формируемых с их помощью снимков и изображений

ПК-1.3/Зн2 Технологии и алгоритмы анализа больших данных

ПК-1.3/Зн3 Программные среды и языки, применяемые для обработки больших объемов данных

Уметь:

ПК-1.3/Ум1 Ум1 проводить оценку фотографического качества аэрокосмических изображений и анализ соответствия его заданным в ТЗ на съемку параметрам

ПК-1.3/Ум2 Использовать научную и научно-техническую литературу в области обработки больших данных при программной реализации алгоритмов анализа больших данных

ПК-1.3/Ум3 Использовать применяемые в обработке больших данных программное обеспечение для решения аналитических и исследовательских задач профессиональной деятельности

Владеть:

ПК-1.3/Нв1 Навыками оценки качественных характеристик используемой при аэросъемке фотографической аппаратуры и соответствия их решаемой задаче

ПК-1.3/Нв2 Навыками использования типовых алгоритмов анализа больших данных для решения практических задач профессиональной деятельности

ПК-1.3/Нв3 Навыками использования специализированного программного обеспечения для анализа больших данных

3. Место дисциплины в структуре ОП

Дисциплина (модуль) Б1.В.04 «Языки программирования для работы с большими данными» относится к формируемой участниками образовательных отношений части образовательной программы и изучается в семестре(ах): 1.

В процессе изучения дисциплины студент готовится к решению типов задач профессиональной деятельности, предусмотренных ФГОС ВО и образовательной программой.

4. Объем дисциплины (модуля) и виды учебной работы

Период обучения	Общая трудоемкость (часы)	Общая трудоемкость (ЗЕТ)	Практические занятия (часы)	Самостоятельная работа (часы)	Промежуточная аттестация (часы)
1 семестр	108	3	68	40	Зачет
Всего	108	3	68	40	

5. Содержание дисциплины (модуля)

5.1. Разделы, темы дисциплины и виды занятий

(часы промежуточной аттестации не указываются)

Наименование раздела, темы	Всего	Практические занятия	Самостоятельная работа	Планируемые результаты обучения, соотнесенные с результатами освоения программы
Раздел 1. Основы Python для обработки больших данных	**14**	**8**	**6**	ПК-1.2
Тема 1.1. Введение в язык Python и особенности работы с большими данными	4	2	2	
Тема 1.2. Управляющие конструкции и обработка исключений	4	2	2	
Тема 1.3. Модули, пакеты и генераторы	6	4	2	
Раздел 2. Объектно-ориентированное программирование для масштабируемых решений	**8**	**4**	**4**	ПК-1.2
Тема 2.1. ООП-подход к обработке больших данных	4	2	2	
Тема 2.2. ООП-рефакторинг и оптимизация	4	2	2	
Раздел 3. NumPy и Pandas для работы с большими табличными данными	**10**	**6**	**4**	ПК-1.1 ПК-1.2 ПК-1.3
Тема 3.1. NumPy для эффективных численных вычислений	4	2	2	
Тема 3.2. Pandas для анализа больших датасетов	6	4	2	
Раздел 4. Распределенные вычисления с Dask	**18**	**12**	**6**	ПК-1.2
Тема 4.1. Введение в Dask и параллельные вычисления	6	4	2	
Тема 4.2. Dask DataFrame для табличных данных	6	4	2	
Тема 4.3. Dask Delayed для параллелизации функций	6	4	2	
Раздел 5. Обработка больших пространственных векторных данных	**18**	**12**	**6**	ПК-1.1 ПК-1.2 ПК-1.3
Тема 5.1. GeoPandas для векторного анализа	6	4	2	
Тема 5.2. Dask-GeoPandas для распределенной обработки	6	4	2	

Тема 5.3. API и потоковая обработка геоданных	6	4	2	
Раздел 6. Обработка больших растровых данных	**16**	**10**	**6**	ПК-1.1 ПК-1.2 ПК-1.3
Тема 6.1. RasterIO для работы с большими растрами	6	4	2	
Тема 6.2. Xarray и rioxarray для многомерных данных	6	4	2	
Тема 6.3. Распределенная обработка растров	4	2	2	
Раздел 7. Облачные платформы для больших геоданных	**15**	**10**	**5**	ПК-1.1 ПК-1.2 ПК-1.3
Тема 7.1. Google Earth Engine для обработки данных планетарного масштаба	6	4	2	
Тема 7.2. Работа с большими коллекциями в GEE	6	4	2	
Тема 7.3. Интеграция облачных и локальных вычислений	3	2	1	
Раздел 8. Оптимизация и профилирование	**9**	**6**	**3**	ПК-1.2
Тема 8.1. Многопоточность и многопроцессорность	3	2	1	
Тема 8.2. Асинхронное программирование	3	2	1	
Тема 8.3. Профилирование и оптимизация памяти	3	2	1	
Итого	**108**	**68**	**40**	

5.2. Содержание разделов, тем дисциплин и формы текущего контроля

Раздел 1. Основы Python для обработки больших данных

Тема 1.1. Введение в язык Python и особенности работы с большими данными

Установка и настройка среды разработки для работы с большими данными . Основные типы и структуры данных Python. Функции ввода-вывода для больших файлов.

Тема 1.2. Управляющие конструкции и обработка исключений

Условные операторы и циклы для обработки больших массивов данных . Создание эффективных функций. Обработка исключений при работе с большими файлами. Контекстные менеджеры для управления ресурсами. Работа с текстовыми файлами по частям.

Текущий контроль

Вид (форма) контроля
Просмотр выполненных работ

Тема 1.3. Модули, пакеты и генераторы

Организация кода в модули и пакеты . Генераторы и итераторы для обработки больших данных. Модуль itertools для эффективной итерации. Ленивые вычисления.

Текущий контроль

Вид (форма) контроля
Просмотр выполненных работ

Раздел 2. Объектно-ориентированное программирование для масштабируемых решений

Тема 2.1. ООП-подход к обработке больших данных

Классы и объекты для представления больших датасетов .Инкапсуляция логики обработки данных. Создание классов для потоковой обработки.

Тема 2.2. ООП-рефакторинг и оптимизация

Наследование и полиморфизм для расширяемых систем. Абстрактные классы для унификации интерфейсов обработки данных.

Текущий контроль

Вид (форма) контроля
Просмотр выполненных работ

Раздел 3. NumPy и Pandas для работы с большими табличными данными

Тема 3.1. NumPy для эффективных численных вычислений

Векторизованные операции в NumPy . Работа с большими многомерными массивами. Оптимизация использования памяти. Chunking стратегии для обработки массивов, не помещающихся в память.

Тема 3.2. Pandas для анализа больших датасетов

Эффективная работа с DataFrame для больших данных . Чтение больших CSV файлов по частям (chunksize). Оптимизация типов данных для экономии памяти. Использование категориальных типов. Параллельная обработка с pandas.

Текущий контроль

Вид (форма) контроля
Просмотр выполненных работ

Раздел 4. Распределенные вычисления с Dask

Тема 4.1. Введение в Dask и параллельные вычисления

Архитектура Dask и принципы работы. Dask Arrays для больших численных массивов. Планировщики задач: синхронный, многопоточный, распределенный. Визуализация графа вычислений.

Тема 4.2. Dask DataFrame для табличных данных

Dask DataFrame как масштабируемая альтернатива Pandas. Ленивые вычисления и оптимизация запросов. Партиционирование данных. Операции shuffle и их оптимизация. Интеграция с Pandas workflows.

Тема 4.3. Dask Delayed для параллелизации функций

Декоратор @delayed для параллелизации пользовательских функций. Применение Dask Delayed к функциям пакета geometry. Построение графов вычислений. Оптимизация CPU-интенсивных задач.

Текущий контроль

Вид (форма) контроля
Просмотр выполненных работ

Раздел 5. Обработка больших пространственных векторных данных

Тема 5.1. GeoPandas для векторного анализа

Работа с большими векторными датасетами в GeoPandas . Пространственные индексы для ускорения операций. Оптимизация пространственных соединений. Буферизация и overlay операции для больших данных.

Тема 5.2. Dask-GeoPandas для распределенной обработки

Масштабирование GeoPandas с помощью dask-geopandas. Партиционирование пространственных данных. Распределенные пространственные операции. Обработка данных OpenStreetMap для больших территорий .

Текущий контроль

Вид (форма) контроля
Просмотр выполненных работ

Тема 5.3. API и потоковая обработка геоданных

Работа с API геосервисов для получения больших объемов данных . Пагинация и батчинг запросов. Кеширование результатов. Асинхронные запросы с aiohttp.

Текущий контроль

Вид (форма) контроля
Просмотр выполненных работ

Раздел 6. Обработка больших растровых данных

Тема 6.1. RasterIO для работы с большими растрами

Чтение больших растров по окнам (windows) . Пирамиды изображений и overview. Виртуальные растры (VRT). Оптимизация операций чтения/записи.

Тема 6.2. Xarray и rioxarray для многомерных данных

Xarray для работы с многомерными научными данными. Интеграция с Dask для чанкованных массивов. Rioxarray для геопространственных растров. Ленивая загрузка и обработка временных серий спутниковых снимков.

Тема 6.3. Распределенная обработка растров

Dask-чанкинг для xarray/rioxarray. Параллельная обработка больших растровых мозаик. Map-reduce операции над растрами. Оптимизация I/O операций.

Текущий контроль

Вид (форма) контроля
Просмотр выполненных работ

Раздел 7. Облачные платформы для больших геоданных

Тема 7.1. Google Earth Engine для обработки данных планетарного масштаба

Архитектура Google Earth Engine . Серверная обработка петабайтов данных. Каталоги данных и их эффективное использование. Оптимизация запросов и reduceRegion операций.

Тема 7.2. Работа с большими коллекциями в GEE

Фильтрация и обработка ImageCollection . Временной анализ больших серий. Экспорт результатов в Google Cloud Storage. Батч-обработка и оптимизация квот.

Тема 7.3. Интеграция облачных и локальных вычислений

Гибридные подходы: GEE + локальная обработка. Использование Google Colab для масштабирования . Интеграция с BigQuery для табличных данных.

Текущий контроль

Вид (форма) контроля
Просмотр выполненных работ

Раздел 8. Оптимизация и профилирование

Тема 8.1. Многопоточность и многопроцессорность

Модули threading и multiprocessing в Python. GIL и его обход. Пулы процессов для CPU-bound задач. Concurrent.futures для высокоуровневого параллелизма.

Тема 8.2. Асинхронное программирование

Asyncio для I/O-bound операций. Асинхронная обработка API запросов. Aiofiles для асинхронной работы с файлами. Интеграция с синхронным кодом.

Тема 8.3. Профилирование и оптимизация памяти

Профилирование с cProfile и line_profiler. Анализ использования памяти с memory_profiler. Оптимизация структур данных. Использование Numba и Cython для критических участков.

Текущий контроль

Вид (форма) контроля
Просмотр выполненных работ

6. Порядок проведения промежуточной аттестации

Порядок проведения промежуточной аттестации регламентируется локальными нормативными актами Университета:
- Положением о формах, периодичности, порядке текущего контроля успеваемости и промежуточной аттестации обучающихся по основным образовательным программам высшего образования – программам бакалавриата, программам специалитета, программам магистратуры в федеральном государственном бюджетном образовательном учреждении высшего образования «Московский государственный университет геодезии и картографии», п.6 «Организация и проведение промежуточной аттестации».
- Положением о порядке осуществления образовательного процесса, проведения текущего контроля успеваемости и промежуточной аттестации обучающихся с применением электронного обучения и дистанционных образовательных технологий в федеральном государственном бюджетном образовательном учреждении высшего образования «Московский государственный университет геодезии и картографии», п.8 «Проведение промежуточной аттестации с применением ЭО и ДОТ».

Промежуточная аттестация: Зачет, 1 семестр.
Зачет по дисциплине является формой промежуточной аттестации и проводится в форме тестирования.

Студентам будет предложен тест, содержащий 20 вопросов. На выполнение работы отводится 60 минут, в течение которых использование вспомогательных материалов и электронных устройств запрещено.

Оценивание проводится по принципу «зачтено» или «не зачтено» Для успешной сдачи зачета и получения отметки «зачтено» студенту необходимо дать правильные ответы не менее чем на 60% вопросов, что составляет 12 правильных ответов из 20. В случае, если количество правильных ответов составит менее 12, зачет считается не сданным и выставляется отметка «не зачтено»

7. Оценочные материалы текущего контроля

Раздел 1. Основы Python для обработки больших данных
Контролируемые ИДК: ПК-1.2

Тема 1.2. Управляющие конструкции и обработка исключений
Форма контроля/оценочное средство: Просмотр выполненных работ

Тема 1.3. Модули, пакеты и генераторы
Форма контроля/оценочное средство: Просмотр выполненных работ

Раздел 2. Объектно-ориентированное программирование для масштабируемых решений
Контролируемые ИДК: ПК-1.2

Тема 2.2. ООП-рефакторинг и оптимизация
Форма контроля/оценочное средство: Просмотр выполненных работ

Раздел 3. NumPy и Pandas для работы с большими табличными данными
Контролируемые ИДК: ПК-1.1 ПК-1.2 ПК-1.3

Тема 3.2. Pandas для анализа больших датасетов
Форма контроля/оценочное средство: Просмотр выполненных работ

Раздел 4. Распределенные вычисления с Dask
Контролируемые ИДК: ПК-1.2

Тема 4.3. Dask Delayed для параллелизации функций
Форма контроля/оценочное средство: Просмотр выполненных работ

Раздел 5. Обработка больших пространственных векторных данных
Контролируемые ИДК: ПК-1.1 ПК-1.2 ПК-1.3

Тема 5.2. Dask-GeoPandas для распределенной обработки
Форма контроля/оценочное средство: Просмотр выполненных работ

Тема 5.3. API и потоковая обработка геоданных
Форма контроля/оценочное средство: Просмотр выполненных работ

Раздел 6. Обработка больших растровых данных
Контролируемые ИДК: ПК-1.1 ПК-1.2 ПК-1.3

Тема 6.3. Распределенная обработка растров
Форма контроля/оценочное средство: Просмотр выполненных работ

Раздел 7. Облачные платформы для больших геоданных
Контролируемые ИДК: ПК-1.1 ПК-1.2 ПК-1.3

Тема 7.3. Интеграция облачных и локальных вычислений
Форма контроля/оценочное средство: Просмотр выполненных работ

Раздел 8. Оптимизация и профилирование
Контролируемые ИДК: ПК-1.2

Тема 8.3. Профилирование и оптимизация памяти
Форма контроля/оценочное средство: Просмотр выполненных работ

8. Оценочные материалы промежуточной аттестации

1 семестр, Зачет
Контролируемые ИДК: ПК-1.1 ПК-1.2 ПК-1.3
Вопросы/Задания:

1. Какой модуль стандартной библиотеки Python предоставляет функции для создания эффективных итераторов?

2. Как называется библиотека, позволяющая масштабировать операции GeoPandas на наборы данных, превышающие объём оперативной памяти, за счёт разбиения на партиции и распределённой обработки?

3. Как называется тип данных Pandas, рекомендуемый для столбцов с небольшим количеством уникальных строковых значений с целью экономии памяти?

4. Дополните предложение термином:

Параметр _____ при чтении больших CSV-файлов средствами Pandas позволяет читать и обрабатывать файл по частям (чанкам), а не загружать его целиком в память.

5. Дополните предложение термином:

Формат хранения данных _____ является колоночным и хорошо подходит для аналитических запросов к большим датасетам, в отличие от строкового формата CSV.

6. Чем Dask DataFrame принципиально отличается от Pandas DataFrame? Раскройте содержание партиционирования и поясните его роль в этом отличии.

7. Что такое GIL (Global Interpreter Lock)? Опишите его влияние на многопоточное выполнение CPU-интенсивных задач в Python и назовите способы его обхода, упомянутые в программе дисциплины.

8. Раскройте содержание понятия ленивых вычислений (lazy evaluation) применительно к библиотеке Dask: дайте определение, поясните роль графа задач и объясните, чем ленивое выполнение отличается от немедленного (eager) выполнения, характерного для обычного Pandas.

9. Раскройте содержание оптимизации пространственных соединений (spatial join) больших векторных данных в GeoPandas: объясните, зачем нужен пространственный индекс, и опишите, как его отсутствие сказывается на вычислительной сложности операции соединения.

10. Раскройте содержание архитектуры Google Earth Engine (GEE): объясните, в чём заключается принципиальное отличие серверной обработки данных GEE от традиционного подхода локальной обработки данных, и укажите, какое практическое следствие это имеет для работы с данными планетарного масштаба.

11. Расположите этапы чтения большого CSV-файла по частям (chunks) средствами Pandas и агрегации результата в правильной последовательности.

1. Накопление промежуточного результата обработки чанка в общем аккумуляторе.
2. Получение очередного чанка данных в виде отдельного DataFrame.
3. Повторение получения и обработки чанков до достижения конца файла, за которым следует получение итогового результата.
4. Открытие CSV-файла с указанием параметра chunksize.
5. Выполнение необходимой обработки (фильтрации, агрегации) над текущим чанком.

12. Расположите этапы построения пространственного индекса и его использования для пространственного соединения в GeoPandas в правильной последовательности.

1. Формирование итогового результата соединения на основе объектов, прошедших точную геометрическую проверку.
2. Точная геометрическая проверка пересечения только для найденных кандидатов, а не для всех объектов индексированного слоя.
3. Построение пространственного индекса (например, R-дерева) для одного из соединяемых слоёв.
4. Для каждого объекта другого слоя – быстрый поиск по индексу кандидатов, потенциально пересекающихся с этим объектом.

13. Расположите этапы обработки временной серии спутниковых снимков в Google Earth Engine в правильной последовательности.

1. Экспорт итогового результата в облачное хранилище.
2. Определение интересующей коллекции изображений (ImageCollection).
3. Агрегация результатов по всей отфильтрованной коллекции (например, вычисление среднего значения индекса за период).
4. Применение операции reduceRegion для получения значения по заданному региону.
5. Применение функции обработки к каждому изображению коллекции (например,

вычисление индекса).
6. Фильтрация коллекции по временному диапазону и территории интереса.

14. Расположите этапы асинхронной обработки запросов к API геосервиса с использованием aiohttp в правильной последовательности.

1. Формирование списка асинхронных задач-запросов к API.
2. Сбор результатов всех завершившихся запросов после их одновременного выполнения.
3. Одновременный запуск всех задач с ожиданием их параллельного выполнения.

15. Расположите этапы профилирования и оптимизации памяти Python-скрипта, обрабатывающего большие данные, в правильной последовательности.

1. Анализ отчёта профилировщика и определение конкретных участков кода или структур данных, требующих оптимизации.
2. Повторный запуск профилировщика для проверки эффекта внесённых изменений.
3. Запуск скрипта с профилировщиком памяти (например, memory_profiler) для выявления мест наибольшего потребления памяти.
4. Внесение изменений в код (например, использование более компактных типов данных или обработка данных по частям).

16. Установите соответствие между библиотекой экосистемы Python для работы с большими геопространственными данными и типом данных, с которым она работает.

GeoPandas - Распределённые и параллельные вычисления над данными любого из перечисленных типов
RasterIO - Многомерные размеченные массивы (например, временные серии растровых снимков)
Xarray/rioxarray - Большие векторные (точечные, линейные, полигональные) геопространственные данные
Dask - Большие растровые файлы, чтение по окнам

17. Установите соответствие между форматом хранения данных и его характерной особенностью.

Parquet - Формат, разработанный для быстрой записи и чтения DataFrame между R и Python
Feather - Колоночный формат, хорошо подходящий для аналитических запросов к большим датасетам
Zarr - Простой текстовый строковый формат без колоночной оптимизации для аналитических запросов
CSV - Иерархический формат для хранения больших объёмов научных данных
HDF5 - Формат для хранения чанкованных, сжатых N-мерных массивов, облачно-оптимизированная альтернатива других форматов

18. Установите соответствие между инструментом профилирования Python-кода и тем, что он измеряет.

Line_profiler - Потребление оперативной памяти кодом
memory_profiler - Число вызовов и суммарное время выполнения каждой функции программы
cProfile - Время выполнения кода построчно

19. Установите соответствие между технологией (модулем) параллелизма Python и типом задач, для которого она подходит.

Multiprocessing - Задачи, ограниченные ожиданием ввода-вывода, с явным асинхронным управлением переключением
threading - CPU-интенсивные задачи, ускоряемые компиляцией критичных участков кода в машинный код
asyncio - Задачи, ограниченные ожиданием ввода-вывода, в пределах одного процесса
Numba/Cython - CPU-интенсивные задачи, требующие обхода GIL через независимые процессы

20. Установите соответствие между понятием Google Earth Engine и его определением.

ImageCollection - Предварительно загруженный и оптимизированный для обработки на серверах Google набор общедоступных геопространственных данных
reduceRegion - Стек или временная серия изображений, объединённых в единую коллекцию

9. Материально-техническое и учебно-методическое обеспечение дисциплины

9.1. Перечень основной и дополнительной учебной литературы

Основная литература

1. Железнов М. М. Методы и технологии обработки больших данных: учебно-методическое пособие / М. М. Железнов. - Москва: МИСИ – МГСУ, 2020. - 46 с. - 978-5-7264-2193-3. - Текст: электронный // Издательство Лань: [сайт]. - URL: https://e.lanbook.com/book/145102 - Режим доступа: по подписке

2. Воронов В. И. Data Mining - технологии обработки больших данных: учебное пособие / В. И. Воронов, Л. И. Воронова, В. А. Усачев. - Data Mining - технологии обработки больших данных - Москва: Московский технический университет связи и информатики, 2018. - 47 с. - 2227-8397. - Текст: электронный // IPR SMART: [сайт]. - URL: https://www.iprbookshop.ru/81324.html - Режим доступа: по подписке

Дополнительная литература

1. Макшанов А. В. Большие данные. Big Data: учебник для вузов / А. В. Макшанов, А. Е. Журавлев, Л. Н. Тындыкарь. - 4-е изд., стер. - Санкт-Петербург: Лань, 2024. - 188 с. - 978-5-507-47346-5. - Текст: электронный // Издательство Лань: [сайт]. - URL: https://e.lanbook.com/book/362318 - Режим доступа: по подписке

9.2. Профессиональные базы данных и ресурсы «Интернет», к которым обеспечивается доступ обучающихся

Профессиональные базы данных

Не используются.

Ресурсы «Интернет»

1. https://docs.python.org/3/index.html - Техническая документация по языку программирования Python
2. https://sdo.miigaik.ru/ - Система дистанционного обучения МИИГАиК

9.3. Программное обеспечение и информационно-справочные системы, используемые при осуществлении образовательного процесса по дисциплине

Перечень программного обеспечения
(обновление производится по мере появления новых версий программы)

1. Anaconda;
2. Microsoft Visual Studio;

Перечень информационно-справочных систем
(обновление выполняется еженедельно)

1. 1С Предприятие;

9.4. Оборудование и технические средства обучения

Компьютерный класс

Маркерная доска - 1 шт.
Моноблок - 96 шт.
Проектор - 8 шт.

Автоматизированное рабочее место (АРМ) - 40 шт.
Плоттер - 1 шт.
Сканер - 1 шт.
Интерактивная панель - 1 шт.

10. Методические указания по освоению дисциплины (модуля)

С целью успешного освоения данного предмета студенту необходимо выполнять домашние работы по указанным преподавателям темам и представленным на занятиях материалам. Для более успешного выполнения работы студенту предлагается использование учебно-методических материалов, соответствующих указанной теме.

Самостоятельная работа студентов описывается и регулируется методическими рекомендациями по дисциплине.

Перечень вопросов для промежуточной аттестации:

1. Для чего в Python используется контекстный менеджер (конструкция `with`)?
2. Какой модуль стандартной библиотеки Python предоставляет функции для создания эффективных итераторов?
3. Какой подход из перечисленных НЕ является способом обхода Global Interpreter Lock (GIL) для CPU-bound задач?
4. Какой инструмент используется для по-строчного профилирования кода с целью выявления узких мест?
5. Какой принцип ООП используется для сокрытия сложной логики обработки данных внутри класса?
6. Какова основная цель использования абстрактных классов при проектировании систем обработки данных?
7. Что такое векторизация в NumPy?
8. Для чего предназначен параметр `chunksize` при чтении больших CSV-файлов с помощью Pandas?
9. Какой тип данных в Pandas рекомендуется использовать для столбцов с небольшим количеством уникальных строковых значений для экономии памяти?
10. На каком принципе основана работа библиотеки Dask?
11. Какой декоратор Dask используется для распараллеливания выполнения обычных Python-функций?
12. Чем Dask DataFrame принципиально отличается от Pandas DataFrame?
13. Какой планировщик задач Dask следует использовать для выполнения вычислений на кластере из нескольких машин?
14. Какова основная цель пространственного индекса в GeoPandas?
15. Для какой задачи предназначена библиотека `dask-geopandas`?
16. Что такое "пагинация" при работе с API геосервисов?
17. Какая библиотека используется для чтения больших растровых файлов по частям (окнам)?
18. Что такое виртуальный растр (VRT)?
19. Какая библиотека предназначена для работы с многомерными размеченными массивами данных и хорошо интегрируется с Dask?
20. В чем заключается основная особенность архитектуры Google Earth Engine (GEE)?
21. Что делает функция `reduceRegion` в Google Earth Engine?
22. Что представляет собой `ImageCollection` в Google Earth Engine?
23. Для какого типа задач наиболее эффективно применение `asyncio`?
24. Какая технология используется для ускорения Python-кода путем его JIT-компиляции в машинный код?
25. Какой формат хранения данных является колоночным и хорошо подходит для аналитических запросов к большим датасетам?
26. Что такое "ленивые вычисления"?
27. Какая библиотека является основной для работы с геопространственными векторными данными в экосистеме Python?

28. Какой инструмент используется для анализа потребления памяти Python-скриптом?
29. Какой формат данных разработан для быстрой записи и чтения DataFrame между R и Python?
30. Какая библиотека позволяет использовать `async/await` для асинхронной работы с файлами?
31. Что такое "чанкинг" (chunking) применительно к массивам NumPy или Xarray?
32. Какая библиотека является зависимостью `rioxarray` и обеспечивает ему геопространственные функции?
33. Какой гибридный подход к вычислениям упоминается в программе курса?
34. Что такое GIL (Global Interpreter Lock)?
35. Какой формат данных предназначен для хранения чанкованных, сжатых N-мерных массивов и является облачно-оптимизированной альтернативой HDF5?
36. Какая библиотека используется для асинхронных HTTP-запросов?
37. Какое основное преимущество использования генераторов при обработке больших файлов?

Описание возможностей изучения дисциплины лицами с ОВЗ и инвалидами

При необходимости обучения студентов-инвалидов и лиц с ограниченными возможностями здоровья аудиторные занятия могут быть заменены или дополнены изучением полнотекстовых лекций, презентаций, видео- и аудиоматериалов в электронной информационно-образовательной среде (ЭИОС) университета. Индивидуальные задания подбираются в адаптированных к ограничениям здоровья формах (письменно или устно, в форме презентаций). Выбор методов обучения зависит от их доступности для инвалидов и лиц с ограниченными возможностями здоровья.

Рабочая программа дисциплины для инвалидов и лиц с ограниченными возможностями здоровья разрабатывается индивидуально с учетом особенностей психофизического развития, индивидуальных возможностей и состояния здоровья таких обучающихся.