ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ БЮДЖЕТНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

ФИНАНСОВЫЙ УНИВЕРСИТЕТ ПРИ ПРАВИТЕЛЬСТВЕ РОССИЙСКОЙ ФЕДЕРАЦИИ

АЛТАЙСКИЙ ФИЛИАЛ

УТВЕРЖДАЮ
Директор филиала

_____В.А. Иванова

«26» апреля 2022 г.

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ ПО ДИСЦИПЛИНЕ

Технологии обработки данных

(наименование дисциплины)

Направление подготовки	**09.03.03 «Прикладная информатика»**
Профиль(и)/направленность	**Прикладные информационные системы в экономике и финансах**
Ответственный за актуализацию РПД	**Жданов Е.П.**
Год утверждения РПД	**2022 г.**

(основной РПД)

СОДЕРЖАНИЕ

1. Перечень планируемых результатов освоения образовательной программы (перечень компетенций) с указанием индикаторов их достижения и планируемых результатов обучения по дисциплине 3
2. Перечень вопросов, заданий, тем для подготовки к текущему контролю 4

1. Перечень планируемых результатов освоения образовательной программы (перечень компетенций) с указанием индикаторов их достижения и планируемых результатов обучения по дисциплине

Код компе-тен-ции	Наименование компетенции	Индикаторы достижения компетенции	Результаты обучения (умения и знания), соотнесенные с индикаторами достижения компетенции
ПКН-6	Способность организовывать поиск и сбор информации, ее хранение в структурированном виде, проектировать и реализовывать реляционные и нереляционные базы и хранилища данных	Демонстрирует знание основ реляционных баз данных, нормализации данных, ACID, CRUD, ORM, использует транзакции.	**Знать:** способы хранения данных с отношениями «один ко многим» и «один к одному». **Уметь:** с помощью языка Python сохранять и читать данные с от- ношениями «один ко многим» и «один к одному».
		Демонстрирует знание различных технологий хранения данных: реляционные и нереляционные базы данных, документарные хранилища, извлекает данные из разных источников и в разных форматах, в том числе программно.	**Знать:** различные технологии сериализации, хранения и чтения данных. **Уметь:** с помощью языка Python сохранять и читать данные в по- пулярных универсальных фор- матах.
		Проектирует хранилища данных исходя из их назначения и характера данных, выбирает инструментальное и архитектурное решение, физическую и логическую схему данных и обосновывает свой выбор.	**Знать:** подходы к организации структуры данных в форматах CSV, XML и JSON. **Уметь:** с помощью языка Python сохранять и читать данные в форматах CSV, XML и JSON.

2. Перечень вопросов, заданий, тем для подготовки к текущему контролю

1. Какие структуры данных используются в Numpy? В чем их отличие от списков Python?
2. Какие функции для генерации массивов использует Numpy?
3. Какие способы предлагает Numpy для извлечения данных из массивов?
4. Что такое векторизация кода и почему это ускоряет работу программ?
5. Какие виды матричных операций реализованы в Numpy?
6. Назовите основные функции чтения данных из файлов и записи в файл в Numpy?
7. Какие есть функции в Numpy для статистического анализа данных?
8. Какие функции используются для преобразования формы, размера и соединения массивов?
9. Объясните, в чём отличие функций arange() и linspace()?
10. Что такое срез массива?
11. Допустим, m — это двумерный массив. Что означают конструкции m[:, j] и m[i, :]?
12. Что означают отрицательные индексы в срезах массивов?
13. Как быстро определить номера минимальных и максимальных элементов в данном массиве?
14. Что такое универсальные функции?

Ответы на вопросы

1. Какие структуры данных используются в Numpy? В чем их отличие от списков Python?

Главный объект NumPy - это однородный многомерный массив. Чаще всего это одномерная последовательность или двумерная таблица, заполненные элементами одного типа, как правило числами, которые проиндексированы кортежем положительных целых чисел. В NumPy, элементы этого кортежа называются осями, а число осей рангом. Данные могут иметь сложную структуру. Для работы с такими данными NumPy предоставляет возможность создавать структурированные массивы. Структурированный массив — это массив, который организован как последовательность именованных полей (столбцов), причем каждое поле может содержать данные любого типа.

Списки в Python - упорядоченные изменяемые коллекции объектов произвольных типов (почти как массив, но типы могут отличаться). Список может

содержать любое количество любых объектов (в том числе и вложенные списки), или не содержать ничего.

2. Какие функции для генерации массивов использует Numpy?

Массив может быть создан из обычного списка или кортежа Python с использованием функции array(). Функция zeros() заполняет массив нулями, функция ones() - единицами, а функция empty() - случайными числами, которые зависят от состояния памяти. По умолчанию, тип создаваемого массива - *float64*. NumPy предоставляет функции arange() и linspace(), которые возвращают одномерные массивы.

3. Какие способы предлагает Numpy для извлечения данных из массивов?

Индексация: обратиться к элементу можно по его индексу. a[7]

Срезы: a[4:8]. Извлечение элементов с определенным шагом. a[0:9:2]

К массивам можно применять оператор in:

```
a=np.array([ -1, 1, -1, 9, -1, 25, -1, 49, -1, 81, -1, 121])
for i in a:
print(i**(1/2), end = ', ')
```

nan, 1.0, nan, 3.0, nan, 5.0, nan, 7.0, nan, 9.0, nan, 11.0,

4. Что такое векторизация кода и почему это ускоряет работу программ?

Массивы NumPy могут быть обычными операндами в математических выражениях. Для выполнения таких операций на Python, были вынуждены писать циклы.

Писать такие циклы в NumPy, нет никакой необходимости, потому что все операции и так выполняются поэлементно, то есть массивы являются векторами, а операции — векторными.

5. Какие виды матричных операций реализованы в Numpy?

Все математические операции Python: +, -, /, **, %, // , +=, -=, *=, /= и тому подобные, преобразование типа, логические.

6. Назовите основные функции чтения данных из файлов и записи в файл в Numpy?

Если массивы, которые необходимо сохранить являются небольшими, то можно воспользоваться функцией numpy.save(). Файлы *.npy* удобны для хранения одного массива, если в одном файле нужно сохранить несколько массивов, то необходимо воспользоваться функцией numpy.savez(), которая сохранит их в несжатом виде в файле NumPy с форматом *.npz*. Что бы вместе с массивами сохранялись их оригинальные имена, необходимо в функции numpy.savez() указывать их как ключи словарей Python. В случае очень больших массивов можно воспользоваться функцией numpy.savez_compressed().

Массив можно загрузить из файла с помощью функции numpy.load().

7. Какие есть функции в Numpy для статистического анализа данных?

Элементарные статистические функции: np.amin(a) Минимальный элемент массива, np.amax(a) максимальный элемент, np.median(a) медиана элементов массива, np.mean(a) среднее арифметическое, np.var(a) дисперсия, np.std(a) стандартное отклонение и другие, еще не изученные.

8. Какие функции используются для преобразования формы, размера и соединения массивов?

Изменить форму массива можно с помощью функции reshape.

Разделить массив вдоль горизонтальной оси можно с помощью функции hsplit, а вдоль вертикальной vsplit. Если размерность массива больше 2 и делить его нужно вдоль определенной оси, то можно воспользоваться функцией array_split, указав необходимую ось.

Функция vstack объединяет массивы по вертикали, а hstack по горизонтали.

Функция column_stack представляет одномерные массивы в виде столбцов после чего объединяет их. Функция row_stack эквивалентна vstack.

9. Объясните, в чём отличие функций arange() и linspace()?

Если функция arange используется с аргументами типа *float*, то предсказать количество элементов в возвращаемом массиве не так-то просто. Гораздо чаще возникает необходимость указания не шага изменения чисел в диапазоне, а количества чисел в заданном диапазоне. Функция linspace, так же как и arange принимает три аргумента, но третий аргумент, как раз и указывает количество чисел в диапазоне.

10. Что такое срез массива?

Срезы позволяют извлекать части массива для создания новых массивов. Когда вы используете срезы для списков Python, результирующие массивы — это копии, но в NumPy они являются представлениями одного и того же лежащего в основе буфера. В зависимости от части массива, которую необходимо извлечь, нужно использовать синтаксис среза; это последовательность числовых значений, разделенная двоеточием (:) в квадратных скобках.

Причем индексы массивов могут содержать любое количество осей и элементов по этим осям.

11. Допустим, m — это двумерный массив. Что означают конструкции m[:, j] и m[i, :]?

Извлекаем столбец j.

Извлекаем строку i.

12. Что означают отрицательные индексы в срезах массивов?

Индексы могут принимать отрицательные целые значения. В этом случае отсчет ведется от конца массива.

13. Как быстро определить номера минимальных и максимальных элементов в данном массиве?

Функция argmin() возвращает индекс минимального значения вдоль указанной оси.

В случаях, когда минимальному значению соответствует несколько элементов, то возвращается индекс элемента с первым вхождением. Данная функция обладает эквивалентным методом класса *ndarray*, т.е. np.argmin(a) равносильно вызову метода a.argmin().

Функция nanargmin() возвращает индекс минимального значения вдоль указанной оси с игнорированием значений *nan*.

Аналогично, argmax().

15.Что такое универсальные функции?

Универсальная функция - это оболочка над обычной функцией, благодаря которой возможно выполнять определенные действия над целыми массивами. Данные функции выполняют действия над массивами поэлементно, поддерживают механизм транслирования, автоматически преобразуют типы данных и обеспечивают доступ к более тонким настройкам своей работы.

Например, np.sin(x, out = y).

Критерии балльной оценки различных форм текущего контроля успеваемости

Вид учебной деятельности	*Баллы*	*Примечание*
Выступление по вопросу на семинарском занятии	До 5	20 баллов до 15 ноября
Аудиторная практическая работа	До 11	
Посещаемость	До 4	
Аудиторная практическая работа	До 11	20 баллов до 30 декабря
Письменная работа	До 5	

Посещаемость	До 4	
Итого	**40**	За семестр

Порядок перевода 100-балльной оценки в пятибалльную

100-балльная система	**5-балльная система**
50-100	зачтено
менее 50	не зачтено
86-100	отлично
70-85	хорошо
50-69	удовлетворительно
Менее 50	неудовлетворительно

Студент считается аттестованным при условии, что сумма баллов за работу в семестре (модуле) и ответ на зачете/экзамене составляет не менее 51 балла.