**ФОНД ОЦЕНОЧНЫХ СРЕДСТВ**

по дисциплине

**Базовые методы обработки данных с использованием Python**

**ПАСПОРТ**

**фонда оценочных средств**

**по дисциплине «Базовые методы обработки данных с использованием Python»**

(наименование дисциплины)

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Семестр** | **Интерактив** | **Трудоемкость, кред.** | **Общий объем курса, час.** | **Лекции, час.** | **Практич. занятия, час.** | **Лаборат. работы, час.** | **СРС, час.** | **КСР, час.** | **Форма(ы) контроля, экз./зач./КР/КП** |
| 2 |  | 2 | 72 | 16 | 0 | 16 | 40 | 0 | З |
| ИТОГО | 0 | 2 | 72 | 16 | 0 | 16 | 40 | 0 |  |

Группа: М20-201, М20-204

**Модели контролируемых компетенций**

В результате освоения дисциплины у выпускника формируются следующие компетенции:

ОПК-1 - способен решать актуальные задачи фундаментальной и прикладной математики

ОПК-2 - способен совершенствовать и реализовывать новые математические методы решения прикладных задач

ОПК-3 - способен разрабатывать математические модели и проводить их анализ при решении задач в области профессиональной деятельности

ПК-2 - способен к разработке и внедрению наукоемкого программного обеспечения, способствующего решению передовых задач науки и техники на основе современных математических методов и алгоритмов

ПК-4.1 - способен проводить обработку и интеллектуальный анализ данных с использованием математического аппарата и современных цифровых технологий.

**В результате освоения дисциплины студент должен:**

**Знать:**

|  |  |
| --- | --- |
| **З-1** | Основные языковые конструкции в Python |
| **З-2** | Основные структуры данных в Python |

**Уметь:**

|  |  |
| --- | --- |
| **У-1** | Применять основные структуры данных и функции |
| **У-2** | Работать с текстом, структурированными файлами включая файлы в формате json |

**Владеть:**

|  |  |
| --- | --- |
| **В-1** | Навыками работы с библиотеками pandas, scipy, matplotlib, bs4 |

|  |  |  |  |
| --- | --- | --- | --- |
| **Компетенции** | **Знания (знать)** | **Умения (уметь)** | **Навыки (владеть)** |
| ОПК-1, | З-1, З-2 | У-1, У-2 |  |
| ОПК-2 | З-1, З-2 | У-1, У-2 |  |
| ОПК-3 | З-1, З-2 | У-1, У-2 |  |
| ПК-2 | З-1, З-2 | У-1, У-2 | В-1 |
| ПК-4.1 | З-1, З-2 | У-1, У-2 | В-1 |

**Программа оценивания контролируемых компетенций**

Формирование у студентов компетенций контролируется в течение всего времени освоения дисциплины в рамках:

* текущего контроля;
* рубежного контроля;
* промежуточного контроля.

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **№ п.п.** | **Наименование раздела учебной дисциплины** | **Неде-ли** | **Лек-ции, час.** | **Практ. зан./ семи-нары, час.** | **Лаб. рабо-ты, час.** | **Обязат. текущий контроль (форма\*, неделя)** | **Аттеста-ция раздела (форма\*, неделя)** | **Макси-мальный балл за раздел \*\*** | **Компетенции по разделам, проверяемые при текущем и рубежном контроле** | **Компетенции, проверяемые на зач. /экз.** |
| 1 семестр | | | | | | | | | | |
| 1 | Структуры данных и языковые конструкции в Python | 1-8 | 8 |  | 8 | Т | КИ, 8 | 25 | ОПК-1, ОПК-2, ОПК-3,  ПК-2  ПК-4.1 |  |
| 2 | Обработка и визуализация данных в Python | 9-16 | 8 |  | 8 | Т | КИ, 16 | 25 | ОПК-1, ОПК-2, ОПК-3,  ПК-2  ПК-4.1 |  |
|  | Зачет |  |  |  |  |  | З | 50 |  | ОПК-1, ОПК-2, ОПК-3,  ПК-2  ПК-4.1 |
|  | Итого за 1 семестр |  |  |  |  |  |  | 100 |  |  |

**Соответствие оценочных средств видам контроля**

|  |  |
| --- | --- |
| **Вид контроля** | **Наименование оценочного средства (способ оценки: устно/ письменно /комп. технолог.)** |
| КИ | Контроль по итогам выполнения (интегральная оценка без проведения дополнительного контроля) |
| Т | Тестирование |
| З | Вопросы к зачету |

**Методика оценки результатов сдачи зачеа**

по курсу ««Базовые методы обработки данных с использованием Python»»

Критерии оценки знаний устанавливаются в соответствии с требованиями к профессиональной подготовке, исходя из действующих учебных планов и программ, с учётом характера будущей практической деятельности выпускника.

**«ОТЛИЧНО»** (45-50 баллов) - студент владеет знаниями предмета в соответствии с рабочей программой, достаточно глубоко осмысливает дисциплину; самостоятельно, в логической последовательности и исчерпывающе отвечает на вопросы билета.

**«ХОРОШО»** (35-45 баллов) - студент владеет знаниями дисциплины почти в полном объеме программы (имеются пробелы знаний только в некоторых, особенно сложных разделах); самостоятельно и отчасти при наводящих вопросах дает полноценный ответы на вопросы билета.

**«УДОВЛЕТВОРИТЕЛЬНО»** (30-34 баллов) - студент владеет основным объемом знаний по дисциплине; проявляет затруднения в самостоятельных ответах, оперирует неточными формулировками; в процессе ответов допускаются ошибки по существу вопросов.

**«НЕУДОВЛЕТВОРИТЕЛЬНО»** (ниже 30 баллов) - студент не освоил обязательного минимума знаний предмета; не способен ответить на вопрос билета даже при дополнительных наводящих вопросах экзаменатора.

**Итоговая оценка по курсу выставляется в соответствии**

**со следующей таблицей:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Сумма баллов по дисциплине** | **Оценка по 4-х бальной шкале** | **Зачет** | **Оценка (ECTS)** | **Градация** |
| 90 - 100 | 5 (отлично) | Зачтено | А | Отлично |
| 85 - 89 | 4 (хорошо) | В | Очень хорошо |
| 75 - 84 | С | Хорошо |
| 70 - 74 | D | Удовлетворительно |
| 65 - 69 | 3 (удовлетворительно) |
| 60 - 64 | E | Посредственно |
| Ниже 60 | 2 (неудовлетворительно) | Не зачтено | F | Неудовлетворительно |

|  |
| --- |
| ИНСТИТУТ ИНТЕЛЛЕКТУАЛЬНЫХ КИБЕРНЕТИЧЕСКИХ СИСТЕМ |
| КАФЕДРА КИБЕРНЕТИКИ |

**Вариант контрольной работы по дисциплине**

**«Базовые методы обработки данных с использованием Python»**

1. Какая встроенная функция Python лучше всего подходит для цепочечных вычислений (в частности, вычислений значения многочлена по схеме Горнера):

а) chain()

б) map()

в) reduce() +

2. Какой модуль стандартной библиотеки Python позволяет работать с WWW на более низком уровне:

а) httplib +

б) urlparse

в) urllib2

3. Сокрытие информации о внутреннем устройстве объекта, при котором вся работа с объектом ведется только через общедоступный интерфейс называется:

а) абстракцией

б) инкапсуляцией +

в) агрегацией

4. Что делает метод normalize() DOM-объекта:

а) готовит XML для красивого вывода

б) сводит воедино все идущие подряд текстовые узлы +

в) исправляет XML, добавляя пропущенные теги

5. Дан массив >>> c = array([[1,2], [2,3], [4,5]])

Чему равен срез c[:,1]:

а) array([1, 2, 4])

б) array([2, 3])

в) array([2, 3, 5]) +

6. Какие новые имена появятся в текущем модуле после выполнения следующего кода:

import sre as re

from re import compile

а) имена sre, re и compile

б) только имена re и compile +

в) только имена sre и compile

7. Чему будет равен результат выполнения:

urlparse.urlsplit(“http://google.com/search?q=Python#1″):

а) (‘http’, ‘google.com’, ‘/search’, ”, ‘q=Python’, ‘1’)

б) (‘http://’, ‘google.com/’, ‘search?’, ‘q=Python#’, ‘1’)

в) (‘http’, ‘google.com’, ‘/search’, ‘q=Python’, ‘1’) +

8. Для чего применяется метод nextset() объекта-курсора:

а) для перехода к следующему набору записей результата запроса +

б) для перехода к следующей записи результата запроса

в) для получения следующего набора записей результата запроса

9. В каком модуле нужно искать функции, помогающие тестировать программу:

а) dictutils

б) profile

в) unittest +

10. Каким образом в модуле poplib представлен сеанс работы с POP3-сервером:

а) набор функций

б) экземпляр класса РОРЗ +

в) список кортежей

11. Какой из перечисленных обработчиков mod\_python выполняется раньше других:

а) PythonPostReadRequestHandler +

б) PythonHandler

в) PythonFixupHandler

12. Что будет получено в результате вычисления следующего выражения:

(0 < 5 <= 3) and (0 / 0):

а) True (или 1)

б) False (или 0) +

в) синтаксическая ошибка

13. Какое из приведенных ниже регулярных выражений некорректно:

а) a+b++ +

б) (?P(ac))

в) (a+b+)+

14. Зачем в XML пространства имен:

а) пространства имен позволяют указывать опции для приложения, обрабатывающего XML

б) пространства имен позволяют включать однотипные XML-документы друг в друга

в) для сочетания в одном документе XML с различными DTD +

15. Что такое регулярное выражение:

а) шаблон, описывающий множество строк +

б) синтаксически правильное выражение на языке Python

в) шаблон для поиска файлов в каталоге

16. Для чего нужны функции модуля gettext:

а) для получения текста от пользователя

б) для обеспечения интернационализации программы +

в) для чтения строки со стандартного ввода

17. Какое значение threadsafety соответствует ситуации, когда потоки могут одновременно использовать как DB-API 2.0 совместимый модуль, так и соединения, получаемые на основе этого модуля:

а) 1

б) 0

в) 2 +

18. Можно ли в XML использовать собственные теги:

а) нельзя

б) можно +

в) можно, если указаны пространства имен

19. Экземпляры какого класса сочетают замок и средство коммуникации между потоками:

а) Event

б) Lock

в) Condition +

20. Как средствами самого Python определить имена формальных аргументов функции func(), если известно, что функция написана на Python:

а) func.func\_globals

б) inspect.getargspec(func) +

в) func.func\_locals

21. Какие кодировки исходного текста программы поддерживает интерпретатор Python:

а) большинство кодировок, распространенных сегодня +

б) ASCII, Latin-1, UTF-8

в) ASCII

22. Какой метод позволяет узнать, имеет ли данное сообщение несколько частей:

а) items()

б) get\_main\_type() +

в) get\_payload()

23. Какой метод позволяет узнать, имеет ли данное сообщение несколько частей:

а) items()

б) get\_payload()

в) is\_multipart() +

24. Что включает в себя Zope:

а) поддержку CGI-сценариев

б) поддержку сценариев DTML +

в) СУБД общего назначения

25. Что включает в себя Zope:

а) СУБД общего назначения

б) поддержку CGI-сценариев

в) собственный web-сервер +

26. Как перевести Unicode-строку u в кодировку koi8-r:

а) u.decode(‘koi8-r’)

б) u.encode(‘koi8-r’) +

в) string.decode(u, ‘koi8-r’)

27. К какому уровню модели взаимодействия открытых систем относится протокол FTP:

а) приложений +

б) представления

в) сетевому

**Вопросы к зачету по дисциплине**

**«Базовые методы обработки данных с использованием Python»**

Билет включает два вопроса.

**ВОПРОСЫ К ЗАЧЕТУ**

Вопросы по синтаксису Python:

Литералы чисел и строк в Python

Условный оператор и каскадная условная конструкция в Python

Строки в Python. Методы find, count, replace, strip, upper, lower и другие.

Строки в Python. Срезы с двумя и тремя параметрами

Кортежи в Python. Создание и использование. Кортежи переменных

Списки в Python. Методы списков и операции со списками

Срезы списков. Присваивание в срез. Проблема копирования списка

Списки в Python. List comprehensions: генерация списков.

Цикл for и его особенности в Python. Функция range()

Двумерные массивы (списки списков). Вложенная генерация.

Классы в Python. Перегрузка операторов.

Классы в Python. Конструктор класса.

Классы в Python. Наследование.

Оператор yield и функции-генераторы.

Алгоритмические и теоретические вопросы:

Позиционные системы счисления

Связь двоичной и шестнадцатеричной систем счисления

Основы алгебры логики. Операции И, ИЛИ, НЕ, исключающее или, импликация, эквиваленция. Таблицы истинности функций

Основы алгебры логики. Свойства операций И и ИЛИ

Основы алгебры логики. Операция НЕ. Законы де Моргана

Основы алгебры логики. Табличное задание логической функции. Дизъюнктивная нормальная форма

Структурное программирование. Декомпозиция задачи и проектирование «сверху-вниз».

Кодировки текста ASCII и Unicode.

Рекурсия. Прямой и обратный ход рекурсии. Стек вызовов

Однопроходные алгоритмы: подсчёт, сумма, произведение.

Среднеквадратическое отклонение: однопроходный алгоритм.

Однопроходные алгоритмы: поиск числа в потоке, максимум.

Однопроходные алгоритмы: местоположение максимума, три максимальных элемента.

Тест простоты числа.

Разложение числа на множители.

Наивный поиск подстроки в строке.

Алгоритм обращения массива.

Алгоритм циклического сдвига в массиве.

Сортировка обезьяны.

Сортировка вставками.

Сортировка выбором.

Сортировка методом пузырька.

Сортировка дурака

Сортировка подсчётом.

Поразрядная сортировка.

Рекурсивные сортировки. Быстрая сортировка.

Рекурсивные сортировки. Сортировка слиянием.

Пирамида (куча). Пирамидальная сортировка.

Виды сортировок. Устойчивость сортировок.

Алгоритм Евклида

Вычисление чисел Фибоначчи.

Ханойские башни.

Генерация всех перестановок (рекурсивная)

Использование списка как стека. Метод pop()

Динамическое программирование. Задачи о Кузнечике на прямой.

Динамическое программирование. Задачи о Короле на плоскости.

Списки: односвязный, двусвязный, кольцо.

Стек.

Очередь.

Хеш-функция. Открытая хеш-таблица.

Хеш-функция. Закрытая хеш-таблица