

МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное учреждение науки
«Институт ядерной физики им. Г.И. Будкера СО РАН»

УТВЕРЖДАЮ
директор ИЯФ СО РАН,

академик _____ П.В.Логачев

«_____» _____ 2018 г.

ЯЗЫК PYTHON ДЛЯ НАУЧНЫХ ВЫЧИСЛЕНИЙ

Рабочая программа дисциплины

Направление подготовки

03.06.01 Физика и астрономия (уровень подготовки кадров высшей квалификации)

Общая трудоемкость дисциплины: 3 зачетные единицы – **108** часа.

Форма промежуточной аттестации: зачет

Виды деятельности:

Лекции	0	контактная работа обучающихся с преподавателем	34
Практические занятия	34		
Самостоятельная работа	70	занятия в активной и интерактивной форме	34
Консультации			
Зачеты	4	Экзамены	

Новосибирск- 2018

Программа курса «Язык PYTHON для научных вычислений» составлена в соответствии с требованиями федерального государственного стандарта высшего образования по направлению подготовки 03.06.01 Физика и астрономия (уровень подготовки кадров высшей квалификации).

Цель курса – приобретение аспирантами навыков программирования на современном высокоуровневом языке Python, который всё шире используется в научных вычислениях и автоматизации научного эксперимента.

Преподавание дисциплины предусматривает следующие формы организации учебного процесса: практические занятия в терминальном классе, самостоятельная работа студента, зачет.

Общая трудоемкость дисциплины составляет 3 зачетных единицы, 108 часов, в том числе 34 часа практических занятий, 70 часов самостоятельной работы.

Составитель:

к.т.н., Г.А. Фатькин

Рабочая программа

Содержание

1. Цели освоения дисциплины	4
2. Место дисциплины в структуре программы обучения	4
3. Компетенции обучающегося, формируемые в результате освоения дисциплины	4
4. Структура и содержание дисциплины	5
5. Образовательные технологии	7
6. Учебно-методическое обеспечение самостоятельной работы студентов и аспирантов	7
7. Фонд оценочных средств для проведения аттестации по итогам освоения дисциплины:	7
8. Учебно-методическое и информационное обеспечение дисциплины	7
9. Материально-техническое обеспечение дисциплины	8

«Современные технологии разработки электроники»

Рабочая программа дисциплины

1. Цели освоения дисциплины

В последнее время язык Python приобретает всё большую и большую популярность для научных вычислений, в силу большого удобства, динамической природы, наличия большого количества библиотек для вычислений, импорта и экспорта данных. Одним из основных преимуществ языка является его открытость, и открытость большинства программных пакетов, таким образом исследователь может разобраться в применяемых алгоритмах, и даже модифицировать их при необходимости. В силу этой открытости, и простоты языка, многие вновь разрабатываемые алгоритмы быстро реализуются в виде библиотек для Python, в частности существуют библиотеки для работы с: линейной алгеброй, символьными расчётами, графами, нейронными сетями, статистической обработкой данных, задачами оптимизации, и проч.

Целью освоения дисциплины «Язык PYTHON для научных вычислений» является ознакомление с языком Python, средой разработки, а также библиотеками для математической обработки, построения графиков и анализа данных для этого языка.

2. Место дисциплины в структуре программы обучения

Дисциплина «Язык PYTHON для научных вычислений» относится к вариативной части блока 1 «Дисциплины (модули)». Дисциплина является элективной (дисциплиной по выбору аспиранта). Студенты, приступающие к изучению этой дисциплины, должны иметь общую базовую подготовку в рамках программы 5-6 лет обучения в ВУЗе, в том числе:

- знать основы математического анализа и линейной алгебры,
- владеть английским языком на уровне, достаточном для чтения технической документации.

3. Компетенции обучающегося, формируемые в результате освоения дисциплины

Дисциплина «Язык PYTHON для научных вычислений» нацелена на формирование у выпускника следующих компетенций:

- готовность участвовать в работе российских и международных исследовательских коллективов по решению научных и научно-образовательных задач (УК-3);
- готовность использовать современные методы и технологии научной коммуникации на государственном и иностранном языках (УК-4);
- способность самостоятельно осуществлять научно-исследовательскую деятельность в соответствующей профессиональной области с использованием современных методов исследования и информационно-коммуникационных технологий (ОПК-1);

В результате освоения дисциплины обучающийся должен:

- Знать основные управляющие конструкции Python, ориентироваться в стандартных библиотеках.
- Уметь применять язык Python для задач научных вычислений, обработки результатов экспериментов, визуализации и преобразования данных.
- Уметь находить и устанавливать необходимые библиотеки при помощи пакетного менеджера pip, ориентироваться в документации пакетов.

4. Структура и содержание дисциплины

Дисциплина «Язык PYTHON для научных вычислений» представляет собой полугодовой курс, читаемый в аспирантуре ИЯФ СО РАН. Общая трудоемкость дисциплины составляет 3 зачетных единицы, 108 часов.

№ п/п	Раздел дисциплины	Неделя семестра	Виды учебной работы, включая самостоятельную работу студентов и трудоемкость (в часах)				
			Всего	Аудиторные часы		Сам. работа в течение семестра (не включая период сессии)	Промежуточная аттестация (подготовка к зачету, зачет)
				Лекции (кол-во часов)	Практические занятия (кол-во часов)		
1	Введение	1	6		2	4	
2	Типы, математика, функции, управляющие конструкции	2	6		2	4	
3	Строки, списки, кортежи	3	6		2	4	
4	Словари, множества	4	6		2	4	
5	Файлы	5	6		2	4	
6	Классы и объекты	6-7	10		4	6	
7	Исключения, декораторы	8	6		2	4	
8	Библиотека matplotlib	9	6		2	4	
9	Библиотека numpy	10	6		2	4	
10	Библиотека scipy, линейная алгебра	11	6		2	4	
11	Библиотека scipy, интегрирование, дифф. уравнения	12	6		2	4	
12	Символьные расчёты в sympy	13-14	10		4	6	
13	Статистическая обработка pandas	15	8		2	6	
14	Работа над проектом	16-17	16		4	12	
16	Зачеты		4				4
	Итого по курсу:		108	0	34	70	4

Содержание дисциплины:

- 1.** Введение. Организация занятий, подготовка рабочей среды для курса. Работа с виртуальной машиной, запуск интерпретатора, среда Jupyter. Пакетный менеджер pip и справка python.
- 2.** Типы, математика, функции, управляющие конструкции. Типы в python: int, float, string, типизация duck-typing и типы как объекты. Целочисленная арифметика, арифметика с плавающей точкой, арифметика произвольной точности. Управляющие конструкции языка: ветвление, циклы. Функции как объект первого класса: объявление, передача и возврат параметров.
- 3.** Строки, списки, кортежи. Изменяемые и неизменяемые объекты. Основные операции: конкатенация, выборка, добавление элементов. Дополнительные методы для работы со строками: поиск в строке, разделение строки, замена в строке. Итерация по элементам. Отличия строки и массива байтов.
- 4.** Словари и множества. Основное отличие между словарём и множеством в Python. Добавление и удаление элементов, ключи, итерация по элементам, объединение. Функции zip, map, reduce.
- 5.** Файлы. Работа с файлами: открытие, модификация, закрытие. Оператор with. Бинарные и текстовые файлы. Модули для работы с файлами в различных форматах: CSV, парсеры.
- 6.** Классы и объекты. Создание классов, поля и методы класса. Создание объектов класса, поля объекта. Специальные операторы для классов и объектов. Наследование.
- 7.** Исключения, декораторы. Механизм обработки исключений, исключения стандартных объектов и операторов. Создание собственных исключений. Применение декораторов, декорирование методов.
- 8.** Библиотека matplotlib. Построение различных графиков и диаграмм при помощи библиотеки matplotlib. Экспорт полученных графиков. Различие между графическими «движками».
- 9.** Библиотека numpy. Массивы, создание и операции с массивами в numpy. Многомерные массивы в numpy, манипуляции с формой массивов, копирование массивов. Матрицы в numpy. Способы экспорта и импорта данных в numpy.
- 10.** Библиотека scipy, линейная алгебра. Scipy как надстройка над numpy. Матрицы в scipy, разреженные матрицы. LU-факторизация, решение систем линейных уравнений, задачи на собственные значения.
- 11.** Библиотека scipy, интегрирование, дифференциальные уравнения. Численное интегрирование и дифференцирование. Численное решение дифференциальных уравнений. Преобразования Фурье, Лапласа, вэйвлет-преобразования. Статистические и специальные функции.
- 12.** Символьные расчёты в sympy. Пакет sympy для символьных расчётов. Символы и операции, упрощение символьных выражений, подстановки. Тригонометрические функции, многочлены. Решение уравнений, комбинаторика, дискретная математика. Матрицы в scipy.
- 13.** Статистическая обработка pandas. Наборы данных: одномерные и таблицы, основная информация о них. Индексирование, выборки, импорт и экспорт данных. Работа с временными сериями. Статистические функции.
- 14.** Работа над проектом на python.

5. Образовательные технологии

Практические занятия проводятся в терминальном классе, и разделены на две части. В ходе первой части излагается материал предстоящего занятия с практической демонстрацией на проекторе в интерактивном режиме. В процессе демонстрации, обучающийся самостоятельно на своём рабочем месте повторяет действия преподавателя, варьируя их по собственному желанию, и задаёт вопросы. После чего формулируется задача, использующая только что изложенный материал, и обучающиеся приступают к её решению, советуясь с преподавателем. Окончательное решение задачи производится самостоятельно, в ходе на самоподготовки.

В конце курса, слушателю предлагается сделать проект на Python, связанный с его научной работой или в близкой области. Часть времени практических занятий отведена под доклад результатов работы над проектом и его критического разбора группой.

Текущий контроль по дисциплине осуществляется по посещаемости, результатам работы над задачами. Итоговый контроль знаний по дисциплине осуществляется по результатам сдачи задач и проекта.

6. Учебно-методическое обеспечение самостоятельной работы студентов и аспирантов

Самостоятельная работа студентов поддерживается обсуждением заданий на практических занятиях, использованием литературы из основного и дополнительного списков (см. ниже).

Система контроля включает текущий (по ходу семестра) контроль освоения практического материала.

В качестве текущего контроля успеваемости используется контроль посещаемости, результаты работы над задачами и представления программного проекта.

7. Фонд оценочных средств для проведения аттестации по итогам освоения дисциплины:

Программные проекты подбираются индивидуально в зависимости от научной работы слушателя. Это могут быть компоненты ПО, реализуемого непосредственно в рамках его работы или вспомогательные расчетные или другие программные компоненты. В случае, если такую задачу сформулировать трудно, подбирается проект из близкой области (моделирование, обработка данных, взаимодействие процессов и т. д.)

Примеры задач, предлагаемых обучающимся:

1. Написать программу, осуществляющую деление 1 на 998001 с точностью до 3000 знака после запятой.
2. Написать программу, разбивающую входной текстовый файл на буквы и слова, подсчитывающую количество отдельно встречающихся слов и букв, а также выводящую график получившегося распределения в сравнении с распределением, подсчитанным на основе анализа текстов цифровой библиотеки Google.
3. Реализовать класс кольцевого буфера с методами добавления и удаления элемента.
4. Оценить отличие распределения, выдаваемого генератором псевдослучайных чисел в Python от равномерного на большой выборке. Построить график распределения.
5. Написать программу для матричного анализа движения частицы в канале линейного ускорителя.
6. Реализовать фильтр низких частот в фурье-области с окном Хэмминга.

8. Учебно-методическое и информационное обеспечение дисциплины

Список основной и дополнительной литературы:

1. Грозин А. Г. Питон в научных вычислениях <http://star.inp.nsk.su/~grozin/python/>
2. Руководство по языку Python:
<https://docs.python.org/3/tutorial/index.html>
3. Mark Pilgrim Dive into python
<http://www.diveintopython.net/>
4. Программирование на Python, 4е издание в 2 томах, Лутц М., изд. Символ-Плюс, 2011

в) программное обеспечение и Интернет-ресурсы:

9. Страница Python Foundation: <http://python.org>
10. Страница проектов SciPy/NumPy: <http://scipy.org/>.
11. Страница pandas: <http://pandas.pydata.org>
12. Страница matplotlib: <http://matplotlib.org>

9. Материально-техническое обеспечение дисциплины

Практические занятия проводятся в терминальном классе, оборудованном персональными компьютерами. Для занятий подготовлена виртуальная машина с ОС LINUX и всеми необходимыми пакетами для разработки в среде Python. Обучающийся может использовать виртуальную машину на своём личном компьютере при необходимости.