

Федеральное государственное бюджетное образовательное учреждение высшего образования Московский государственный университет имени М.В. Ломоносова
ФИЗИЧЕСКИЙ ФАКУЛЬТЕТ
КАФЕДРА АСТРОФИЗИКИ И ЗВЕЗДНОЙ АСТРОНОМИИ

УТВЕРЖДАЮ
Декан физического факультета
МГУ
_____ / Н.Н. Сысоев /
«__» _____ 20__ г.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

Наименование дисциплины:

НАУЧНОЕ ПРОГРАММИРОВАНИЕ НА ЯЗЫКЕ PYTHON

Уровень высшего образования:

Специалитет

Направление подготовки:

03.05.01 Астрономия

Направленность (профиль) ОПОП:

Общая специальность

Квалификация «Специалист»

Форма обучения: Очная форма обучения

Рабочая программа рассмотрена и одобрена
Ученым советом физического факультета МГУ

(протокол № _____, _____)

Москва 20__

Рабочая программа дисциплины разработана в соответствии с самостоятельно установленным МГУ образовательным стандартом (ОС МГУ) для реализуемых основных профессиональных образовательных программ высшего образования по направлению подготовки 03.05.01 Физика.

Год (годы) приема на обучение _____

Авторы–составители рабочей программы дисциплины:

к.ф.-м.н. старший научный сотрудник Маланчев Константин Леонидович, отдел релятивистской астрофизики ГАИШ МГУ

Заведующий кафедрой астрофизики и звездной астрономии
Д.ф.-м.н., профессор Постнов Константин Александрович

Аннотация к рабочей программе дисциплины «Научное программирование на языке Python»

Настоящий курс предлагает слушателю познакомиться или развить имеющиеся навыки программирования на языке Python. В последнее десятилетие этот язык стал де-факто стандартом быстрого создания программ во многих областях науки, в том числе астрономии. В курсе будут рассмотрены базовый синтаксис языка, его стандартная библиотека, а также ряд научных библиотек таких как `numpy`, `scipy` и `astropy`. Больше внимание будет уделено часто совершаемым ошибкам и наиболее продуктивным подходам к написанию поддерживаемого кода. Также будут рассмотрены вопросы, посвященные публикации кода и его совместному использованию.

Разделы рабочей программы

1. Место дисциплины в структуре основной профессиональной образовательной программы высшего образования (ОПОП ВО).
2. Входные требования для освоения дисциплины (модуля), предварительные условия (при наличии)
3. Результаты обучения по дисциплине (модулю), соотнесенные с формируемыми компетенциями
4. Форма обучения.
5. Язык обучения.
6. Содержание дисциплины.
7. Объем дисциплины
8. Структурированное по темам (разделам) содержание дисциплины (модуля) с указанием отведенного на них количества академических часов и виды учебных занятий
9. Текущий контроль и промежуточная аттестация.
10. Фонд оценочных средств (ФОС) для оценивания результатов обучения по дисциплине (модулю).
11. Шкала оценивания.
12. Типовые контрольные задания или иные материалы для проведения текущего контроля успеваемости.
13. Типовые контрольные задания или иные материалы для проведения промежуточной аттестации.
14. Материально-техническая база, информационные технологии, программное обеспечение и информационные справочные системы

1. Место дисциплины в структуре ОПОП ВО

Дисциплина «Научное программирование на языке Python» реализуется на 5-м курсе во 10-м семестре и является составной частью профессионального блока вариативной части.

2. Входные требования для освоения дисциплины (модуля), предварительные условия

Отсутствуют.

3. Результаты обучения по дисциплине (модулю), соотнесенные с формируемыми компетенциями

В результате освоения дисциплины у обучающихся должны быть сформированы:

Формируемые компетенции (код компетенции)	Планируемые результаты обучения по дисциплине (модулю)
УК-1	<i>З-1 Знать:</i> базовый синтаксис языка программирования Python <i>З-2 Знать:</i> как создавать программные пакеты на языке Python <i>У-1 Уметь:</i> писать документацию для кода <i>У-2 Уметь:</i> писать автоматические тесты для программного кода <i>В-1 Владеть:</i> навыками параллельного программирования на языке Python <i>В-2 Владеть:</i> навыками работы с астрономическими пакетами и базами данных
ОПК-1	<i>З-1 Знать:</i> научные библиотеки, необходимы для работы с астрономическими данными <i>У-1 Уметь:</i> решать задачи визуализации данных <i>В-12 Владеть:</i> навыками самостоятельной работы

4. **Форма обучения:** очная.

5. **Язык обучения:** русский.

6. **Содержание дисциплины**

Тема 1. Базовые сведения о языке

Python как высокоуровневый объектно-ориентированный язык. Встроенные типы и операторы. Использование Python для функционального программирования: итераторы, генераторы, списковые включения, лямбда-функции. Некоторые модули стандартной библиотеки: os, sys, json, functools, itertools, collections, ... Файл с расширением .py как модуль

Тема 2. Numpy

Загрузка и сохранение данных. Одномерные и многомерные массивы, индексация и срезы. Правила трансляции. Решение задач линейной алгебры.

Тема 3. Matplotlib

Построение простейших двумерных графиков. Использование LaTeX.

Тема 4. Scipy

Интерполяция данных. Задачи оптимизации. Численное интегрирование. Статистические тесты, распределения случайных величин.

Тема 5. Astropy

Загрузка данных в текстовых и бинарных форматах. Работа со временем и координатами, матчинг каталогов. Вычисления в размерных величинах. Поиск точечных объектов на кадре.

Тема 6. Astroquery

Доступ к публично доступным источникам данных, таким как Gaia, CDS, и тп

Тема 7. Параллельное программирование

Global Interpreter Lock. Многопоточное программирование. Модуль стандартной библиотеки multiprocessing. Сторонний модуль joblib.

Тема 8. Создание пакетов

Средства для создания пакетов и правила оформления. Документация кода. Модульное и интеграционное тестирование

Тема 9. Взаимодействие с другими языками.

Написание модулей Python на языках Cython, C/C++. Модуль numba как JIT-компилятор Python. Модуль стандартной библиотеки ctypes

7. Объем дисциплины

НАЗВАНИЕ ДИСЦИПЛИНЫ	Трудоемкость в зачетных единицах	объем учебной нагрузки в ак. часах				
		Общая трудоемкость	в том числе ауд. занятий			Самостоятельная работа студентов
			Общая аудиторная нагрузка	Лекций	Семинаров	
Научное программирование на языке Python	2	72	36	36	-	36

8. Структурированное по темам (разделам) содержание дисциплины (модуля) с указанием отведенного на них количества академических часов и виды учебных занятий

Изучение курса «Научное программирование на языке Python» включает в себя лекции, на которых рассматриваются возможности пакетов и синтаксис языка; семинарские занятия, на которых студенты решают задачи; самостоятельную работу, заключающуюся в написании индивидуального исследовательского проекта и решении домашнего задания.

№ темы	Наименование раздела дисциплины	Виды учебной нагрузки и их трудоемкость, часы					Форма текущего контроля успеваемости и промежуточной аттестации
		Всего часов	Лекции	Научно-практические занятия	Семинары	Самостоятельная работа	
1	Базовые сведения о языке	6	4	-	-	2	
2	Numpy	6	4	-	-	2	
3	Matplotlib	8	4	-	-	4	
4	Scipy	8	4	-	-	4	
5	Astropy	8	4	-	-	4	
6	Astroquery	8	4	-	-	4	
7	Параллельное программирование	8	4	-	-	4	
8	Создание пакетов	8	4	-	-	4	
9	Взаимодействие с другими языками.	8	4	-	-	4	
	Промежуточная аттестация	4				4	
							Зачет в форме письменной работы с последующим собеседованием
ИТОГО:		72	36		-	36	

9. Текущий контроль и промежуточная аттестация.

Текущий контроль по дисциплине «Научное программирование на языке Python» осуществляется на семинарских занятиях и заключается в оценке ответов на вопросы и активности при обсуждении проблем, изучаемых в рамках тем лекционных занятий,

Промежуточная аттестация по дисциплине «Научное программирование на языке Python» проводится в десятом семестре в форме письменной проверочной работы с последующим собеседованием.

Результаты сдачи зачета оцениваются по шкале «неудовлетворительно», «удовлетворительно», «хорошо», «отлично». Оценки «отлично», «хорошо», «удовлетворительно» означают успешное прохождение промежуточной аттестации.

10. Фонд оценочных средств (ФОС) для оценивания результатов обучения по дисциплине (модулю).

Требования к структуре и содержанию фонда оценочных средств текущего контроля и промежуточной аттестации по дисциплине

Перечень оценочных средств, применяемых на каждом этапе проведения текущего контроля и промежуточной аттестации по дисциплине, представлены в таблице

Наименование оценочного средства	Краткая характеристика оценочного средства	Представление оценочного средства в фонде
Оценочные средства текущего контроля		
Тематический опрос (в форме ответов на вопросы)	Средство контроля, организованное как специальная беседа по тематике предыдущей лекции и рассчитанное на выяснение объема и качества знаний, усвоенных обучающимися по определенному разделу, теме, проблеме.	Перечень тем, изучаемых в рамках дисциплины
Собеседование (в форме беседы, дискуссии по теме)	Средство контроля, организованное как свободная беседа, дискуссия по тематике изучаемой дисциплины, рассчитанное на выяснение объема знаний обучающегося по всем изученным разделам, темам.	Перечень тем, изучаемых в рамках дисциплины
Тестирование	Средство контроля, позволяющее получить оценку уровня фактических знаний студента по изученной теме.	Образцы тестов
Оценочные средства промежуточной аттестации		
Письменная работа	Средство, позволяющее оценить умение решать несложные задачи и проводить оценки физических величин, иллюстрирующие материал пройденных тем.,	Перечень вопросов к экзамену
Собеседование	Средство, позволяющее получить экспертную оценку знаний, умений и навыков по для оценивания и анализа различных фактов и явлений в своей профессиональной области.	Требования к порядку проведения собеседования

11. Шкала оценивания.

Планируемые результаты обучения	Критерии оценивания результатов обучения			
	2	3	4	5
ЗНАТЬ: Базовый синтаксис языка Python УК-1 3-1	Студент не знает элементарных синтаксических правил Python	Студент знает элементарные правила синтаксиса, но ошибается при написании таких продвинутых конструкций	Студент уверенно владеет базовым синтаксисом языка	Студент уверенно владеет базовым синтаксисом языка и оформляет свой код согласно основным требованиям

		как классы и декораторы		PEP8
Уметь: Создавать программные пакеты для Python, а также документировать и тестировать их УК-1 У-1	Студент не знает, что такое пакеты и модули, не умеет создавать проекты	Студент способен оформить свой код в виде пакета, но не знает как писать документацию и тестировать код	Студент способен оформить свой код в виде пакета и понимает как правильно документировать его компоненты	Студент умеет пакетировать, документировать и модульно тестировать код
Владеть: Навыками параллельного программирования УК-1 В-1	Студент не имеет представления о параллельном программировании	Студент владеет основными техническими навыками запуска параллельного кода, но не понимает особенностей его работы	Студент владеет либо навыком многопоточного, либо многопроцессорного запуска кода, хорошо разбираясь в его ограничениях	Студент владеет и навыком многопоточного, и многопроцессорного запуска кода, хорошо разбираясь в его ограничениях

12. Типовые контрольные задания или иные материалы для проведения текущего контроля успеваемости.

Контроль успеваемости будет проводиться по рассмотрению индивидуальных иссле

13. Типовые контрольные задания или иные материалы для проведения промежуточной аттестации.

https://github.com/hombit/scientific_python

14. Материально-техническая база, информационные технологии, программное обеспечение и информационные справочные системы

Основная литература.

C H Swaroop, a byte of python, <https://python.swaroopch.com>

G. van Rossum et al, the python tutorial, <https://docs.python.org/3/tutorial/>

Материально-техническое обеспечение

В соответствии с требованиями п. 5.3. образовательного стандарта МГУ по направлению подготовки «Астрономия».

Курс может быть прочитан в поточной аудитории при наличии: работающих электрических розеток, компьютера, проектора, экрана, учебной доски.