

**Федеральное государственное автономное образовательное
учреждение высшего образования
«Московский физико-технический институт
(национальный исследовательский университет)»**

УТВЕРЖДЕНО
Проректор по учебной работе

А.А. Воронов

	Рабочая программа дисциплины (модуля)
по дисциплине:	Продвинутое программирование на языке Python
по направлению:	Прикладные математика и физика
профиль подготовки:	Общая и прикладная физика
	Физтех-школа физики и исследований им. Ландау
	Физтех-кластер академической и научной карьеры
курс:	1
квалификация:	магистр

Семестр, формы промежуточной аттестации: 1 (осенний) - Дифференцированный зачет

Аудиторных часов: 30 всего, в том числе:

лекции: 30 час.

семинары: 0 час.

лабораторные занятия: 0 час.

Самостоятельная работа: 15 час.

Всего часов: 45, всего зач. ед.: 1

Программу составил: М.Е. Зелёный, ассистент

Программа обсуждена на заседании Физтех-кластера академической и научной карьеры 30.04.2021

Аннотация

Современные научные исследования невозможны без повсеместной компьютеризации: экспериментальные установки представляют собой комплексы сложных приборов с большим количеством датчиков, генерирующие колоссальные объемы экспериментальных данных, а расчеты и моделирование требуют большого числа вычислительных операций, также важно уметь представлять все эти данные в наглядном виде. Поэтому этот курс рассчитан чтобы научить в той или иной мере решать эти задачи с помощью языка программирования Python. В курсе будут изучены продвинутые возможности синтаксиса Python, рассмотрено практическое применение стандартной библиотеки и некоторых библиотек для научного программирования. В частности будет рассмотрено использование инструментов разработки, тестирования и дистрибуции программ на Python, инструменты для численного моделирования и анализ данных, применение различных стилей программирования, работа с хранением, чтением, преобразованием и передачей данных в различных форматах, использование Python для системного, сетевого и Web программирования, разработка пользовательских интерфейсов, реализация асинхронного, параллельного и многопоточного выполнения программы.

1. Цели и задачи

Цель дисциплины

Научить решать практические научные задачи с помощью Python

Задачи дисциплины

Изучение возможностей языка Python. Изучение инструментов стандартной библиотеки. Изучение научных библиотек для Python

2. Перечень формируемых компетенций

Освоение дисциплины направлено на формирование следующих компетенций:

Код и наименование компетенции	Индикаторы достижения компетенции
ОПК-2 Имеет представление об актуальных проблемах науки и техники в области своей профессиональной деятельности, способен на научном языке формулировать профессиональные задачи	ОПК-2.3 Владеет профессиональной терминологией, используемой в современной научно-технической литературе, обладает навыками устного и письменного изложения результатов научной деятельности в рамках профессиональной коммуникации
ОПК-3 Способен выбирать и (или) разрабатывать подходы к решению типовых и новых задач в области профессиональной деятельности, учитывая особенности и ограничения различных методов решения	ОПК-3.1 Способен анализировать задачу, планировать пути решения, предлагать и комбинировать способы решения
	ОПК-3.2 Способен использовать исследовательские методы при решении новых задач, применяя знания в различных областях науки (техники)
	ОПК-3.3 Владеет аналитическими и вычислительными методами решения, понимает и учитывает на практике границы применимости получаемых решений
ПК-1 Способен ставить, формализовывать и решать задачи, в том числе разрабатывать и исследовать математические модели изучаемых явлений и процессов, системно анализировать научные проблемы, получать новые научные результаты	ПК-1.2 Способен выдвигать гипотезы, строить математические модели для описания изучаемых явлений и процессов, оценивать качество разработанной модели
ПК-3 Способен профессионально работать с исследовательским и испытательным оборудованием (приборами и установками, специализированными пакетами прикладных программ) в избранной предметной области	ПК-3.1 Понимает принципы работы используемого оборудования (специализированных пакетов прикладных программ)
	ПК-3.2 Способен проводить эксперимент (моделирование) с использованием исследовательского оборудования (пакетов прикладных программ)
	ПК-3.3 Способен оценивать точность полученных экспериментальных (численных) результатов

3. Перечень планируемых результатов обучения по дисциплине (модулю)

В результате освоения дисциплины обучающиеся должны

знать:

Синтаксис языка Python, принципы работы трансляторов, принципы разработки программного обеспечения и ведения документации.

уметь:

Писать сценарии и программы в объектно-ориентированном и функциональном стилях для численных расчетов, обработки и визуализации данных, уметь работать различными форматами данных, создавать консольные, графические и web интерфейсы.

владеть:

Основными стандартными и научными библиотеками.

4. Содержание дисциплины (модуля), структурированное по темам (разделам) с указанием отведенного на них количества академических часов и видов учебных занятий

4.1. Разделы дисциплины (модуля) и трудоемкости по видам учебных занятий

№	Тема (раздел) дисциплины	Трудоемкость по видам учебных занятий, включая самостоятельную работу, час.			
		Лекции	Семинары	Лаборат. работы	Самост. работа
1	Парадигмы программирования: процедурная, функциональная и объективно-ориентированная	4			2
2	Паттерны проектирования и их реализация в Python	4			2
3	Инструменты разработки	2			1
4	Стандартные типы данных	4			2
5	Дополнительные типы данных	2			1
6	Системное программирование	2			1
7	Ввод/вывод и хранение данных	2			1
8	Численное моделирование и анализ	2			1
9	Визуализация данных	2			1
10	Пользовательские интерфейсы	2			1
11	Сетевые Web технологии	2			1
12	Оптимизация производительности	2			1
Итого часов		30			15
Подготовка к экзамену		0 час.			
Общая трудоёмкость		45 час., 1 зач.ед.			

4.2. Содержание дисциплины (модуля), структурированное по темам (разделам)

Семестр: 1 (Осенний)

1. Парадигмы программирования: процедурная, функциональная и объективно-ориентированная

ООП в Python. Магические методы и метаклассы. Менеджеры контекста. Декораторы классов. ФП в Python. Обработка коллекций в функциональном стиле. Инструменты функционального программирования. Декораторы функций. Пакеты и модули.

2. Паттерны проектирования и их реализация в Python

Принципы проектирования программного обеспечения. SOLID. Паттерны проектирования. MVC.

3. Инструменты разработки

Инструменты разработки: инструменты анализа кода, отладчик, профилирование кода. Соглашения о стиле кода. Виртуальное окружение. Документация: аннотация типов, документирование кода, автоматическая генерация документации. Соглашения о стиле кода. Формат ReStructuredText. Тестирование: модульное тестирование, CI/CD. Дистрибуция программного обеспечения, распространение пакетов Python.

4. Стандартные типы данных

Стандартные типы данных. Длинная арифметика. Работа со строками: шаблоны строк, регулярные выражения. Базовые коллекции.

5. Дополнительные типы данных

Дополнительные типы данных: перечисления, время и дата, классы данных, расширения стандартных коллекций.

6. Системное программирование

Работа с файловой системой. Управление процессами, межпроцессорное взаимодействие. Взаимодействие с ОС. Управление физическими устройствами. USB.

7. Ввод/вывод и хранение данных

Ввод/Вывод: работа с бинарными и текстовыми данными; сериализация/десериализация структура данных в текстовые форматы JSON, XML, бинарный формат Protobuf и с использованием встроенной сериализации; хранение данных в научных форматах HDF5 и ROOT; Numpy IO; архивация данных, отображение данных в память. Основы баз данных. Использование СУБД Sqlite3.

8. Численное моделирование и анализ

Численное моделирование и анализ данных с использованием Numpy и Scipy. Многомерные массивы, их расположение в памяти, операции с массивами. Линейная алгебра. Статистическая обработка данных. Решение оптимизационных задач. Численное интегрирование. Интерполяция. Обработка сигналов.

9. Визуализация данных

Создание графических материалов, анимации и интерактивной инфографики.

10. Пользовательские интерфейсы

Логирование, разбор аргументов командной строки, создание командной оболочки. Событийно-ориентированное программирование и создание GUI приложений. Типовые архитектуры GUI приложений;

11. Сетевые Web технологии

Сетевое программирование --- передача данных через сокеты. Использование встроенного сервера. Основы веб-разработки.

12. Оптимизация производительности

Оптимизация производительности. Использование LLVM с Numba. Cython. JIT-компиляция с PyPy. Вызов нативного кода.

5. Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине (модулю)

Компьютерный класс с проектором, usb-измерительные устройства.

6. Перечень рекомендуемой литературы

Основная литература

1. Программирование на Python [Текст]/М. Лутц, -СПб., Символ-Плюс, 2002

Дополнительная литература

1. Паттерны проектирования [Текст] : [учеб. пособие для вузов] / Эр. Фримен, Эл. Фримен ; при участии К. Сьерра, Б. Бейтса ; [пер. с англ. Е. Матвеева] .— СПб. : Питер, 2012 .— 656 с.

7. Перечень ресурсов информационно-телекоммуникационной сети "Интернет", необходимых для освоения дисциплины (модуля)

<https://www.python.org/> сайт документации на язык Python и стандартные библиотеки

<https://pypi.org/> сайт пакетного менеджера PyPi

<https://docs.conda.io/en/latest/> сайт пакетного менеджера Conda

<https://www.scipy.org/> сайт документации на библиотеки Numpy и SciPy.

<https://matplotlib.org/> сайт документации библиотеки Matplotlib

<https://plotly.com/python/> сайт документации библиотеки Plotly

<https://www.pytables.org/> сайт документации библиотеки Pytables

<https://www.hdfgroup.org/solutions/hdf5/> сайт документации формата данных HDF5

<https://root.cern/manual/python/> сайт документации библиотеки PyROOT

8. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине (модулю), включая перечень необходимого программного обеспечения и информационных справочных систем (при необходимости)

Интерпретатор CPython, интегрированная среда разработки Pycharm CE, сервис видеоконференций Google Meet, сервис презентации Google Presentain, сервис распространения исходного кода Github

9. Методические указания для обучающихся по освоению дисциплины (модуля)

Студент, изучающий дисциплину, должен, с одной стороны, овладеть общим понятийным аппаратом, а с другой стороны, должен научиться применять теоретические знания на практике.

В результате изучения дисциплины студент должен знать основные определения и понятия, уметь применять полученные знания для решения различных задач.

Успешное освоение курса требует:

- посещения всех занятий, предусмотренных учебным планом по дисциплине;
- ведения конспекта занятий;
- напряжённой самостоятельной работы студента.

Самостоятельная работа включает в себя:

- чтение рекомендованной литературы;
- проработку учебного материала, подготовку ответов на вопросы, предназначенных для самостоятельного изучения;
- решение задач, предлагаемых студентам на занятиях;
- подготовку к выполнению заданий текущей и промежуточной аттестации.

Показателем владения материалом служит умение без конспекта отвечать на вопросы по темам дисциплины.

Важно добиться понимания изучаемого материала, а не механического его запоминания. При затруднении изучения отдельных тем, вопросов, следует обращаться за консультациями преподавателю.

Возможен промежуточный контроль знаний студентов в виде решения задач в соответствии с тематикой занятий.

Литература для самостоятельного изучения:

- 1.Лутц М. Изучаем Python, 4-е издание. – Пер. с англ. – СПб.: Символ-Плюс, 2011. –1280 с., ил.
- 2.Лутц М. Программирование на Python, том I, 4-е издание. – Пер. с англ. – СПб.: Символ-Плюс, 2011. – 992 с., ил.
- 3.Лутц М. Программирование на Python, том II, 4-е издание. – Пер. с англ. – СПб.: Сим-вол-Плюс, 2011. – 992 с., ил.
- 4.Travis E. Oliphant, Guide to NumPy
- 5.Tosi S. Matplotlib for Python developers. – Packt Publishing Ltd, 2009.
- 6.Python 3 и PyQt 5. Разработка приложений. Н. А Прохоренок, В. А. Дронов.
- 7.Фримен Э. и др. Паттерны проектирования //СПб.: Питер. – 2011.

ПРИЛОЖЕНИЕ

ОЦЕНОЧНЫЕ МАТЕРИАЛЫ ПО ДИСЦИПЛИНЕ (МОДУЛЮ)

по направлению:	Прикладные математика и физика
профиль подготовки:	Общая и прикладная физика Физтех-школа физики и исследований им. Ландау Физтех-кластер академической и научной карьеры (Разработка и применение программного обеспечения в физических исследованиях)
курс:	1
квалификация:	магистр
Семестр, формы промежуточной аттестации: 1 (осенний) - Дифференцированный зачет	
Разработчик:	М.Е. Зелёный, ассистент

1. Компетенции, формируемые в процессе изучения дисциплины

Код и наименование компетенции	Индикаторы достижения компетенции
ОПК-2 Имеет представление об актуальных проблемах науки и техники в области своей профессиональной деятельности, способен на научном языке формулировать профессиональные задачи	ОПК-2.3 Владеет профессиональной терминологией, используемой в современной научно-технической литературе, обладает навыками устного и письменного изложения результатов научной деятельности в рамках профессиональной коммуникации
ОПК-3 Способен выбирать и (или) разрабатывать подходы к решению типовых и новых задач в области профессиональной деятельности, учитывая особенности и ограничения различных методов решения	ОПК-3.1 Способен анализировать задачу, планировать пути решения, предлагать и комбинировать способы решения
	ОПК-3.2 Способен использовать исследовательские методы при решении новых задач, применяя знания в различных областях науки (техники)
	ОПК-3.3 Владеет аналитическими и вычислительными методами решения, понимает и учитывает на практике границы применимости получаемых решений
ПК-1 Способен ставить, формализовывать и решать задачи, в том числе разрабатывать и исследовать математические модели изучаемых явлений и процессов, системно анализировать научные проблемы, получать новые научные результаты	ПК-1.2 Способен выдвигать гипотезы, строить математические модели для описания изучаемых явлений и процессов, оценивать качество разработанной модели
ПК-3 Способен профессионально работать с исследовательским и испытательным оборудованием (приборами и установками, специализированными пакетами прикладных программ) в избранной предметной области	ПК-3.1 Понимает принципы работы используемого оборудования (специализированных пакетов прикладных программ)
	ПК-3.2 Способен проводить эксперимент (моделирование) с использованием исследовательского оборудования (пакетов прикладных программ)
	ПК-3.3 Способен оценивать точность полученных экспериментальных (численных) результатов

2. Показатели оценивания компетенций

В результате изучения дисциплины «Продвинутое программирование на языке Python» обучающийся должен:

знать:

Синтаксис языка Python, принципы работы трансляторов, принципы разработки программного обеспечения и ведения документации.

уметь:

Писать сценарии и программы в объектно-ориентированном и функциональном стилях для численных расчетов, обработки и визуализации данных, уметь работать различными форматами данных, создавать консольные, графические и web интерфейсы.

владеть:

Основными стандартными и научными библиотеками.

3. Перечень типовых (примерных) вопросов, заданий, тем для подготовки к текущему контролю

- 1.Реализовать виртуальный прибор по заданному описанию, управляемые с помощью консольной или графической программы и способный к сетевым коммуникациям, сохраняющий свои показания в базу данных.
- 2.Произвести обработку данных из базы: построить график, провести гистограммирование, найти параметры фитирующей функции для полученного распределения.
3. Напишите декоратор, который хранит вызовы другой функции.
4. Реализуйте JSON Encoder позволяющий сериализовать numpy record array.

5. Создайте программу подключающуюся к сетевому прибору и выходящую данные с него в виде веб-страницы.
6. Создать консольный интерфейс для управления прибором используя модуль `cmd`.
7. Считайте данные с USB-устройства в базу данных.

4. Перечень типовых (примерных) вопросов и тем для проведения промежуточной аттестации обучающихся

Перечень контрольных вопросов (не менее пяти вопросов/заданий)

1. В чем разница между списком и кортежем?
2. Как выполняется интерполяция строк?
3. Что такое декоратор?
4. В чем разница между методами экземпляра, класса и статическими методами в Python?
5. Объясните, как работает функция `reduce`?
6. Переменные в Python передаются по ссылке или по значению?
7. Назовите примеры изменяемых и неизменяемых объектов

Критерии оценивания

Оценка отлично 10 баллов - выставляется студенту, показавшему всесторонние, систематизированные, глубокие знания учебной программы дисциплины, проявляющему интерес к данной предметной области, продемонстрировавшему умение уверенно и творчески применять их на практике при решении конкретных задач, свободное и правильное обоснование принятых решений.

Оценка отлично 9 баллов - выставляется студенту, показавшему всесторонние, систематизированные, глубокие знания учебной программы дисциплины и умение уверенно применять их на практике при решении конкретных задач, свободное и правильное обоснование принятых решений.

Оценка отлично 8 баллов - выставляется студенту, показавшему всесторонние, систематизированные, глубокие знания учебной программы дисциплины и умение уверенно применять их на практике при решении конкретных задач, правильное обоснование принятых решений, с некоторыми недочетами.

Оценка хорошо 7 баллов - выставляется студенту, если он твердо знает материал, грамотно и по существу излагает его, умеет применять полученные знания на практике, но недостаточно грамотно обосновывает полученные результаты.

Оценка хорошо 6 баллов - выставляется студенту, если он твердо знает материал, грамотно и по существу излагает его, умеет применять полученные знания на практике, но допускает в ответе или в решении задач некоторые неточности.

Оценка хорошо 5 баллов - выставляется студенту, если он в основном знает материал, грамотно и по существу излагает его, умеет применять полученные знания на практике, но допускает в ответе или в решении задач достаточно большое количество неточностей.

Оценка удовлетворительно 4 балла - выставляется студенту, показавшему фрагментарный, разрозненный характер знаний, недостаточно правильные формулировки базовых понятий, нарушения логической последовательности в изложении программного материала, но при этом он освоил основные разделы учебной программы, необходимые для дальнейшего обучения, и может применять полученные знания по образцу в стандартной ситуации.

Оценка удовлетворительно 3 балла - выставляется студенту, показавшему фрагментарный, разрозненный характер знаний, допускающему ошибки в формулировках базовых понятий, нарушения логической последовательности в изложении программного материала, слабо владеет основными разделами учебной программы, необходимыми для дальнейшего обучения и с трудом применяет полученные знания даже в стандартной ситуации.

Оценка неудовлетворительно 2 балла - выставляется студенту, который не знает большей части основного содержания учебной программы дисциплины, допускает грубые ошибки в формулировках основных принципов и не умеет использовать полученные знания при решении типовых задач.

Оценка неудовлетворительно 1 балл - выставляется студенту, который не знает основного содержания учебной программы дисциплины, допускает грубейшие ошибки в формулировках базовых понятий дисциплины и вообще не имеет навыков решения типовых практических задач.

5. Методические материалы, определяющие процедуры оценивания знаний, умений, навыков и (или) опыта деятельности

Дифференцированный зачёт проводится в письменной (устной) форме по билетам. В каждом билете представлено два теоретических вопроса. При проведении зачёта и экзамена обучающемуся предоставляется 30 минут на подготовку. Опрос обучающегося не должен превышать одного астрономического часа.