

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ**
Федеральное государственное автономное образовательное учреждение
высшего образования
«Новосибирский национальный исследовательский государственный университет»
(Новосибирский государственный университет, НГУ)

**Физический факультет
Кафедра физико-технической информатики**



Рабочая программа дисциплины

Язык Python в научных вычислениях

направление подготовки: **03.04.02 Физика**
направленность (профиль): **Все профили подготовки**

Форма обучения
Очная

Семестр	Общий объем	Виды учебных занятий (в часах)				Промежуточная аттестация (в часах)				
		Контактная работа обучающихся с преподавателем			Самостоятельная работа, не включая период сессии	Самостоятельная подготовка к промежуточной аттестации	Контактная работа обучающихся с преподавателем			
		Лекции	Практические занятия	Лабораторные занятия			Консультации	Зачет	Дифференцированный зачет	Экзамен
1	2	3	4	5	6	7	8	9	10	11
1	144		32		90	18	2			2
Всего 144 часа / 4 зачётных единицы, из них: - контактная работа 36 часов										
Компетенции ОПК 3										

Руководитель программы
д.ф.-м.н.

И. Б. Логашенко

Новосибирск, 2022

Содержание

1. Перечень планируемых результатов обучения по дисциплине, соотнесённых с планируемыми результатами освоения образовательной программы.	3
2. Место дисциплины в структуре образовательной программы.	3
3. Трудоёмкость дисциплины в зачётных единицах с указанием количества академических часов, выделенных на контактную работу обучающегося с преподавателем (по видам учебных занятий) и на самостоятельную работу.	4
4. Содержание дисциплины, структурированное по темам (разделам) с указанием отведённого на них количества академических часов и видов учебных занятий.	4
5. Перечень учебной литературы.	7
6. Перечень учебно-методических материалов по самостоятельной работе обучающихся.	8
7. Перечень ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины.	8
8. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине.	8
9. Материально-техническая база, необходимая для осуществления образовательного процесса по дисциплине.	9
10. Оценочные средства для проведения текущего контроля и промежуточной аттестации по дисциплине.	9

1. Перечень планируемых результатов обучения по дисциплине, соотнесённых с планируемыми результатами освоения образовательной программы.

Основной целью учебного курса «Язык Python в научных вычислениях» является ознакомить студентов с современными подходами, применяемыми в программировании, и дать начальные навыки пользования языком программирования Python для научных вычислений.

Курс помогает студентам преодолеть начальный порог вхождения в область современного программирования и получить первичные навыки использования языка Python. Знакомство с языком Python помогает студентам открыть для себя этот исключительно мощный, универсальный и простой в использовании инструмент для решения практически любых задач, где требуется написать программу. По итогам курса студенты должны быть способны к самостоятельному изучению языков и технологий в области программирования.

Дисциплина нацелена на формирование у обучающегося общепрофессиональной компетенции:

Результаты освоения образовательной программы (компетенции)	Индикаторы	Результаты обучения по дисциплине
ОПК-3. Способен применять знания в области информационных технологий, использовать современные компьютерные сети, программные продукты и ресурсы информационно-телекоммуникационной сети "Интернет" (далее - сеть "Интернет") для решения задач профессиональной деятельности, в том числе находящихся за пределами профильной подготовки.	<p>ОПК - 3.1. Применяет профессионально-профилированные знания в области компьютерных технологий для решения профессиональных задач, в том числе находящихся за пределами профильной подготовки.</p> <p>ОПК - 3.2. Применяет знания в области информационных технологий для решения поставленных задач научных исследований с помощью современной аппаратуры, программных продуктов и ресурсов информационно- телекоммуникационной сети с использованием новейшего российского и зарубежного опыта.</p>	<p>Знать основы языка Python;</p> <p>Уметь использовать пакеты для научных вычислений.</p> <p>Владеть языком Python.</p>

2. Место дисциплины в структуре образовательной программы.

Учебный курс «Язык Python в научных вычислениях» реализуется в осеннем семестре 1-го курса магистратуры для студентов, обучающихся по направлению подготовки 03.04.02 Физика (уровень магистратуры). Курс является одной из профессиональных дисциплин по выбору, реализуемых кафедрой физических методов исследования твёрдого тела.

Для успешного освоения курса «Язык Python в научных вычислениях» студенты должны обладать предварительными знаниями основ программирования и владеть английским языком в объеме, достаточном для чтения технической документации.

3. Трудоемкость дисциплины в зачётных единицах с указанием количества академических часов, выделенных на контактную работу обучающегося с преподавателем (по видам учебных занятий) и на самостоятельную работу.

Семестр	Общий объем	Виды учебных занятий (в часах)				Промежуточная аттестация (в часах)				
		Контактная работа обучающихся с преподавателем			Самостоятельная работа, не включая период сессии	Самостоятельная подготовка к промежуточной аттестации	Контактная работа обучающихся с преподавателем			
		Лекции	Практические занятия	Лабораторные занятия			Консультации	Зачет	Дифференцированный зачет	Экзамен
1	2	3	4	5	6	7	8	9	10	11
1	144		32		90	18	2			2
Всего 144 часа / 4 зачётные единицы, из них: - контактная работа 36 часов										
Компетенции ОПК 3										

Реализация дисциплины предусматривает практическую подготовку при проведении следующих видов занятий, предусматривающих участие обучающихся в выполнении отдельных элементов работ, связанных с будущей профессиональной деятельностью: практические занятия, самостоятельная работа студента и её контроль преподавателями с помощью контрольной работы, дифференцированного зачёта.

Программой дисциплины предусмотрены следующие виды контроля:

- текущий контроль успеваемости: опрос в начале занятия по материалу предыдущего, контрольная работа;
- промежуточная аттестация: дифференцированный зачёт.

Общая трудоемкость рабочей программы дисциплины составляет 4 зачетные единицы.

- практические занятия – 32 часа;
 - промежуточная аттестация (подготовка к сдаче экзамена, консультация, экзамен) – 22 часа;
- Объём контактной работы обучающегося с преподавателем (практические занятия, консультация, экзамен) составляет 36 часов.

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

Дисциплина «Язык Python в научных вычислениях» представляет собой полугодовой курс, читаемый на 1-м курсе магистратуры физического факультета НГУ в 1 семестре. Общая трудоемкость дисциплины составляет 4 зачётные единицы, 144 академических часа.

№ п/п	Раздел дисциплины	Неделя семестра	Виды учебной работы, включая самостоятельную работу студен- тов и трудоёмкость (в часах)				Промежуточная аттестация (в часах)
			Всего	Аудиторные часы		Сам. работа во время занятий (не включая период сессии)	
				Лекции	Практические занятия		
1	2	3	4	5	6	7	8
1.	Введение в программирование на языке Python, строки	1	8		2	6	
2.	Списки, кортежи	2	8		2	6	
3.	Ассоциативные массивы – словари. Элементы функционального программирования: list, dict, set comprehensions	3	8		2	6	
4.	Основы объектно-ориентированного программирования. Классы.	4	8		2	6	
5	Наследование, статические методы и поля класса	5	8		2	6	
6	Библиотека NumPy для работы с матрицами	6	8		2	6	
7	Контрольная работа по теме «Библиотека NumPy для работы с матрицами»	7	8		2	6	
8	Библиотека pandas для работы с таблицами с разнородными данными	8	8		2	6	
9	Графические библиотеки Matplotlib и Vokeh	9	8		2	6	
10	Библиотека SymPy для символьных вычислений	10	8		2	6	
11	Библиотека Numba для ускорения выполнения кода	11	8		2		
12	Численное интегрирование	12	8		2	6	
13	Решение обыкновенных дифференциальных уравнений	13	8		2	6	
14	Фреймворк PyQt5 для создания GUI-приложений	14	4		2	2	
15	Библиотека PyQtGraph для построения графиков под PyQt5	15	4		2	2	
16	Фоновое выполнение задач под PyQt5	16	4		2	2	
17.	Самостоятельная работа в период подготовки к промежуточной аттестации		18				18
18	Консультация в период промежуточной аттестации		2				2
19	Экзамен		2				2
Всего			144		32	90	22

Программа и основное содержание практических занятий (32 часа)

Занятие 1. Введение в программирование на языке Python, строки (2 часа)

Введение: основные применения языка Python, его место в области научного программирования. Работа со строками: основные функции, методы, операторы.

Занятие 2. Списки, кортежи (2 часа)

Определение, принцип работы списков и кортежей. Разница между списками и кортежами. Основные функции, методы, операторы. Понятие о вычислительной сложности алгоритмов.

Занятие 3. Ассоциативные массивы – словари. Элементы функционального программирования: list, dict, set comprehensions (2 часа)

Определение, принцип работы, аналогия с hash_map в языке си. Основные функции, методы, операторы. Примеры применения. Аналогия различных видов comprehensions с определением множеств в математике. Синтаксис, особенности компактной формы определения списков, кортежей и множеств. Примеры применения.

Занятие 4. Основы объектно-ориентированного программирования. Классы. (2 часа)

Сходства и различия с классами в языке си. Конструкторы, деструкторы. Методы, поля класса. Контроль видимости методов и полей класса: private, protected, public в си и их аналоги в Python.

Занятие 5. Наследование, статические методы и поля класса (2 часа)

Сходства и отличия от наследования в языке си. Механизм работы множественного наследования.

Занятие 6. Библиотека NumPy для работы с матрицами (2 часа)

Сходства и отличия от MatLab. Основные функции, операторы, классы и методы для работы с одно, двух и многомерными массивами.

Занятие 7. Контрольная работа по теме «Библиотека NumPy для работы с матрицами» (2 часа)

Занятие 8. Библиотека pandas для работы с таблицами с разнородными данными (2 часа)

Отличие от NumPy. Основные классы, методы для работы с одномерными сериями и двумерными датафреймами.

Занятие 9. Графические библиотеки Matplotlib и Vokeh (2 часа)

Построение графиков, гистограмм, подготовка их для добавления в научные публикации. Отличия библиотеки Vokeh от Matplotlib, возможности интерактивного взаимодействия с графиками для анализа данных.

Занятие 10. Библиотека SymPy для символьных вычислений (2 часа)

Сравнение с пакетом Mathematica. Приемы работы. Обзор основных функций, методов и классов.

Занятие 11. Библиотека Numba для ускорения выполнения кода (2 часа)

Способы увеличения производительности кода при помощи Numba. Виртуальная машина LLVM (low-level virtual machine). Как её использование позволяет Numba выполнять код быстрее. Сравнение быстродействия кода на Python, «разогнанного» при помощи Numba и кода на C/C++.

Занятие 12. Численное интегрирование (2 часа)

Применение NumPy+Numba для численного интегрирования несколькими методами вручную, а также при помощи специализированных функций из библиотек NumPy и SciPy. Сравнение быстродействия.

Занятие 13. Решение обыкновенных дифференциальных уравнений (2 часа)

Применение NumPy+Numba для решения дифференциальных уравнений несколькими методами вручную, а также при помощи специализированных функций из библиотеки SciPy. Сравнение быстродействия.

Занятие 14. Фреймворк PyQt5 для создания GUI-приложений (2 часа)

Использование библиотеки QtDesigner для создания графических приложений на Python. Основные виджеты и контролы. Сигналы и сокетты.

Занятие 15. Библиотека PyQtGraph для построения графиков под PyQt5 (2 часа)

Установка библиотеки. Сходства и отличия от Matplotlib и bokeh. Основные классы, методы и функции. Интеграция с QtDesigner.

Занятие 16. Фоновое выполнение задач под PyQt5 (2 часа)

Обоснование необходимости особого подхода к вычислительно сложным задачам в графических приложениях. Многозадачность (multitasking), многопроцессорное выполнение (multiprocessing) и многопоточность (multithreading). Global interpreter lock (GIL) и его значение в многозадачности. Способы обхода GIL.

Самостоятельная работа студентов (108 часов)

Перечень занятий на СРС	Объем, час
Подготовка к практическим занятиям.	90
Подготовка к дифференцированному зачёту	18

5. Перечень учебной литературы.

5.1. Основная литература

- Майкл Доусон Программируем на Python // Питер, 2018.

5.2. Дополнительная литература

- *Простой Python : современный стиль программирования : [пер. с англ.] / Билл Любанович* Санкт-Петербург [и др.] : Питер, 2018 476 с. : ил. ; 23 см (Серия "Бестселлеры O'Reilly") ISBN 978-5-496-02088-6
- *Python 3 / Николай Прохоренко, Владимир Дронов.* Санкт-Петербург : БХВ-Петербург, 2016. 461 с. ; 23 см (Самое необходимое) Факт. дата публикации: 2015Предм. указ.: с.445-461 ISBN 978-5-9775-3631-8

6. Перечень учебно-методических материалов по самостоятельной работе обучающихся.

Самостоятельная работа студентов поддерживается следующими учебными пособиями:

- *Изучаем Python : программирование игр, визуализация данных, веб-приложения : [пер. с англ.] / Эрик Мэттис* 2-е изд Санкт-Петербург [и др.] : Питер, 2018 Пер. изд.: Python crash course: A Hands-On, Project-Based Introduction to Programming / by Eric Matthes. - San-Francisco: No Starch Press, 2016 ISBN 978-5-4461-0479-6
- *Python на практике. Создание качественных программ с использованием параллелизма, библиотек и паттернов : [пер. с англ.] / Марк Саммерфилд.* Москва : ДМК Пресс, 2014. Пер. изд.: Python in Practice. Create Better Programs Using Concurrency, Libraries, and Patterns / Mark Summerfield. - Upper Saddle River: Addison-Wesley, 2014. ISBN 978-5-97060-095-5

7. Перечень ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины.

Для освоения дисциплины используются следующие ресурсы:

1. <http://www.inp.nsk.su/~grozin/python/>
2. <https://www.python.org/>
3. <https://pypi.org/>
4. <http://www.numpy.org/>
5. <https://matplotlib.org/>
6. <https://pandas.pydata.org/>
7. <https://www.sympy.org/>
8. <http://numba.pydata.org/>
9. <https://riverbankcomputing.com/software/pyqt/intro>

8. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине.

Для обеспечения реализации дисциплины используется стандартный комплект программного обеспечения (ПО), включающий регулярно обновляемое лицензионное ПО Windows, Linux либо Mac OS.

Требуется установленный интерпретатор Python и/или среда для научного программирования Anaconda либо права на установку этого ПО.

9. Материально-техническая база, необходимая для осуществления образовательного процесса по дисциплине.

Для реализации дисциплины «Язык Python в научных вычислениях» используются специальные помещения:

1. Аудитории, оборудованные мультимедийным проектором и компьютером, необходимых для презентации электронного варианта лекций и проведения компьютерных демонстраций, а также доступом в сеть «Интернет» для установки/доустановки необходимого ПО и просмотра документации.

2. Помещения для самостоятельной работы обучающихся.

Помещения для самостоятельной работы обучающихся оснащены компьютерной техникой с возможностью подключения к сети "Интернет" и обеспечением доступа в электронную информационно-образовательную среду НГУ.

В случае невозможности проведения занятий в очной форме реализация дисциплины осуществляется с применением онлайн-платформы Zoom где обучение проводится на виртуальных аналогах, позволяющим достигать запланированных результатов по дисциплине.

Материально-техническое обеспечение образовательного процесса по дисциплине для обучающихся из числа лиц с ограниченными возможностями здоровья осуществляется согласно «Порядку организации и осуществления образовательной деятельности по образовательным программам для инвалидов и лиц с ограниченными возможностями здоровья в Новосибирском государственном университете».

10. Оценочные средства для проведения текущего контроля и промежуточной аттестации по дисциплине.

10.1 Порядок проведения текущего контроля и промежуточной аттестации по дисциплине

Текущий контроль

Текущий контроль осуществляется в ходе семестра путем опроса в начале каждого практического занятия по материалам предыдущего практического занятия. Студентам необходимо успешно выполнить контрольную работу, предполагающую решение задач из раздела «Библиотека NumPy для работы с матрицами».

Промежуточная аттестация

Освоение компетенций оценивается согласно шкале оценки уровня сформированности компетенции. Положительная оценка по дисциплине выставляется в том случае, если заявленная компетенция ОПК-3 сформирована не ниже порогового уровня в части, относящейся к формированию способности использовать специализированные знания в области программирования на языке Python в профессиональной деятельности.

Окончательная оценка работы студента в течение семестра происходит на дифференцированном зачёте. Дифференцированный зачёт проводится в конце семестра в экзаменационную сессию в форме практического занятия. Практическое задание подбирается таким образом, чтобы проверить уровень сформированности компетенции ОПК-3.

Вывод об уровне сформированности компетенций принимается преподавателем. Каждый вопрос билета оценивается от 0 до 5 баллов. Положительная оценка ставится, когда все компетенции освоены не ниже порогового уровня. Оценки «отлично», «хорошо», «удовлетворительно» означают успешное прохождение промежуточной аттестации.

Соответствие индикаторов и результатов освоения дисциплины

Таблица 10.1

Индикатор	Результат обучения по дисциплине	Оценочные средства
ОПК - 3.1. Применяет профессионально-профилированные знания в области компьютерных технологий для решения профессиональных задач, в том числе находящихся за пределами профильной подготовки.	Знать основы языка Python;	Проведение опроса после лекций, решение задач, экзамен.
ОПК - 3.2. Применяет знания в области информационных технологий для решения поставленных задач научных исследований с помощью современной аппаратуры, программных продуктов и ресурсов информационно- телекоммуникационной сети с использованием новейшего российского и зарубежного опыта.	Уметь использовать пакеты для научных вычислений. Владеть языком Python.	Проведение опроса после лекций, решение задач, экзамен.

10.2 Описание критериев и шкал оценивания индикаторов достижения результатов обучения по дисциплине «Язык Python в научных вычислениях».

Таблица 10.2

Критерии оценивания результатов обучения	Планируемые результаты обучения (показатели достижения заданного уровня освоения компетенций)	Уровень освоения компетенции			
		Не сформирован (0 баллов)	Пороговый уровень (3 балла)	Базовый уровень (4 балла)	Продвинутый уровень (5 баллов)
1	2	3	4	5	6
Полнота знаний	ОПК 3.1	Уровень знаний ниже минимальных требований. Имеют место грубые ошибки.	Демонстрирует общие знания базовых понятий по темам/разделам дисциплины. Допускается значительное количество негрубых ошибок.	Уровень знаний соответствует программе подготовки по темам/разделам дисциплины. Допускается несколько негрубых/ несущественных ошибок. Не отвечает на дополнительные вопросы.	Уровень знаний соответствует программе подготовки по темам/разделам дисциплины. Свободно и аргументированно отвечает на дополнительные вопросы.
Наличие умений	ОПК 3.2	Отсутствие минимальных умений. Не умеет решать стандартные задачи. Имеют место грубые ошибки.	Продемонстрированы частично основные умения. Решены типовые задачи. Допущены негрубые ошибки.	Продемонстрированы все основные умения. Решены все основные задания с негрубыми ошибками или с недочетами.	Продемонстрированы все основные умения. Решены все основные задания в полном объеме без недочетов и ошибок.

Наличие навыков (владение опытом)	ОПК 3.2	Отсутствие владения материалом по темам/разделам дисциплины. Нет навыков в решении стандартных задач. Наличие грубых ошибок.	Имеется минимальный набор навыков при решении стандартных задач с некоторыми недочетами.	Имеется базовый набор навыков при решении стандартных задач с некоторыми недочетами.	Имеется базовый набор навыков при решении стандартных задач без ошибок и недочетов. Продемонстрированы знания по решению нестандартных задач.
-----------------------------------	---------	--	--	--	---

10.3 Типовые контрольные задания и материалы, необходимые для оценки результатов обучения

Контрольная работа по теме «Библиотека NumPy для работы с матрицами»

1. Напишите программу, которая создаёт вектор NumPy со значениями от 0 до 20 (включительно) и меняет знак чисел от 9 до 15 (включительно).
2. Реализуйте функцию, генерирующую квадратную матрицу размера $n \times n$ с единицами по краям и нулями внутри.
3. Реализуйте функцию, окаймляющую заданный массив нулями.
4. Сгенерируйте матрицу 8×8 с единицами и нулями, расположенными в шахматном порядке.
5. Переведите массив с температурами в градусах по Фаренгейту в градусы по Цельсию
6. Сгенерируйте массив вида

```
double (np.arange(9).reshape(3,3)) ->
array([[0, 0, 1, 1, 2, 2],
       [0, 0, 1, 1, 2, 2],
       [3, 3, 4, 4, 5, 5],
       [3, 3, 4, 4, 5, 5],
       [6, 6, 7, 7, 8, 8],
       [6, 6, 7, 7, 8, 8]])
```

7. Реализуйте функцию, которая для заданных n и m генерирует массив размера $2n \times 2m$ вида

```
mosaic(2, 3) ->
array([[ 'R', 'G', 'R', 'G', 'R', 'G'],
       [ 'G', 'B', 'G', 'B', 'G', 'B'],
       [ 'R', 'G', 'R', 'G', 'R', 'G'],
       [ 'G', 'B', 'G', 'B', 'G', 'B']])
```

8. Сгенерируйте матрицу 5×5 из чисел от 0 до 4 вида

```
[[ 0.  1.  2.  3.  4.]
 [ 0.  1.  2.  3.  4.]
 [ 0.  1.  2.  3.  4.]
 [ 0.  1.  2.  3.  4.]
 [ 0.  1.  2.  3.  4.]
```

9. Просуммируйте элементы массива, индекс которых делится на 2, 5, 9 или 11

```
subsum([0,1,2,3,4,5,6,7,8,9,10]) ->
      * * * * *
44
```

10. Определите, сколько памяти занимает заданный NumPy массив.

11. Выделите первый, второй и четвертый элемент третьей, четвертой и пятой строчек заданного массива.

```
[[1 2 3 4 5 6]
 [ 7 8 9 10 11 12]
 [13 14 15 16 17 18]
 [19 20 21 22 23 24]
 [25 26 27 28 29 30]
 [31 32 33 34 35 36]] ->
1й, 2й, 4й элемент 3й, 4й и 5-й строчки:
[[13 14 16]
 [19 20 22]
 [25 26 28]]
```

12. Найдите корни следующих полиномов:

a) $x^2 - 2x + 1$.

b) $x^4 - 12x^3 + 10x^2 + 7x - 10$

13. Напишите функцию, которая возвращала бы номера строк, в которых содержится хотя бы один элемент больше заданного значения.
14. Реализуйте функцию, которая добавляет между элементами входного массива заданное количество нулей.
15. Напишите функцию, возвращающую сумму максимальных элементов в каждой строке массива.

Типовое задание на экзамен

Реализуйте GUI приложение, которое выполняло бы построение семейства решений системы дифференциальных уравнений для аттрактора Лоренца из окрестности заданной точки в трёхмерном пространстве с возможностью модификации начальных условий, а также параметров системы уравнений при помощи стандартных Qt-виджетов.

**Лист актуализации рабочей программы
по дисциплине «Язык Python в научных вычислениях»
по направлению подготовки 03.04.02 Физика
Профиль6 все профили подготовки**

№	Характеристика внесенных изменений (с указанием пунктов документа)	Дата и № протокола Учёного совета ФФ НГУ	Подпись ответственного