

**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**

академик РАН



УТВЕРЖДАЮ
Декан ФФ
А. Е. Бондарь
2020 г.

Рабочая программа дисциплины

ЯЗЫК PYTHON В НАУЧНЫХ ВЫЧИСЛЕНИЯХ

направление подготовки: **03.04.02 Физика, Курс 1, семестр 1**
направленность (профиль): **все профили**

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	72			32	38				2	
Всего 72 часа / 2 зачётные единицы, из них: - контактная работа 34 часа - в интерактивных формах 32 часа										
Компетенции ПК-1, ПК-2										

Разработчики:

к.ф.-м.н., ст. преподаватель
ассистент

А. Г. Грозин
Л. В. Максимов

Заведующий кафедрой ФТИ ФФ НГУ
к.ф.-м.н.

П. П. Кроковный

Руководитель программы
д.ф.-м.н.

И. Б. Логашенко

Новосибирск, 2020

Содержание	
Аннотация	3
1. Перечень планируемых результатов обучения по дисциплине, соотнесённых с планируемыми результатами освоения образовательной программы.	4
2. Место дисциплины в структуре образовательной программы.	4
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.	5
5. Перечень учебной литературы.	10
6. Перечень учебно-методических материалов по самостоятельной работе обучающихся.	10
7. Перечень ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины.	10
8. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине.	11
9. Материально-техническая база, необходимая для осуществления образовательного процесса по дисциплине.	11
10. Оценочные средства для проведения текущего контроля и промежуточной аттестации по дисциплине.	12

Аннотация

к рабочей программе дисциплины «Язык Python в научных вычислениях» Направление: 03.04.02 Физика Направленность (профиль): все профили

Программа дисциплины «Язык Python в научных вычислениях» составлена в соответствии с требованиями СУОС к уровню магистратуры по направлению подготовки 03.04.02 Физика, а также задачами, стоящими перед Новосибирским государственным университетом по реализации Программы развития НГУ. Дисциплина реализуется на физическом факультете Федерального государственного автономного образовательного учреждения высшего профессионального образования Новосибирский национальный исследовательский государственный университет (НГУ) кафедрой физико-технической информатики в качестве факультативной дисциплины. Дисциплина изучается студентами **первого** курса **магистратуры** физического факультета в осеннем семестре.

Цель курса – ознакомить студентов с современными подходами, применяемыми в программировании, и дать начальные навыки пользования языком программирования Python для научных вычислений.

Дисциплина нацелена на формирование у выпускника следующих компетенций:

- способности самостоятельно ставить конкретные задачи научных исследований в области физики и решать их с помощью современной аппаратуры и информационных технологий с использованием новейшего российского и зарубежного опыта (ПК-1);
- способности свободно владеть разделами физики, необходимыми для решения научно-инновационных задач, и применять результаты научных исследований в инновационной деятельности (ПК-2).

В результате освоения дисциплины обучающийся должен:

Знать: основы языка Python.

Уметь: использовать пакеты и фреймворки для Python.

Владеть: языком Python.

Преподавание дисциплины предусматривает следующие формы организации учебного процесса: лабораторные работы, самостоятельная работа обучающихся, дифференцированный зачет.

Программой дисциплины предусмотрены следующие виды контроля:

текущий контроль успеваемости: задания для самостоятельного решения

промежуточная аттестация: дифференцированный зачет.

Общая трудоемкость рабочей программы дисциплины составляет 2 зачетные единицы /72 академических часа.

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.

Факультативный 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 направлена на формирование следующих профессиональных компетенций:

ПК-1: - способность самостоятельно ставить конкретные задачи научных исследований в области физики и решать их с помощью современной аппаратуры и информационных технологий с использованием новейшего российского и зарубежного опыта.

ПК-2 – способность свободно владеть разделами физики, необходимыми для решения научно-инновационных задач, и применять результаты научных исследований в инновационной деятельности.

В результате освоения дисциплины обучающийся должен:

- **Знать:**
 - основы языка Python (ПК 1.1, ПК 2.1);
- **Уметь:**
 - использовать пакеты и фреймворки для Python (ПК 1.2, ПК 2.2).
- **Владеть:**
 - языком Python (ПК 1.3, ПК 2.3).

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. Курс является факультативной дисциплиной, реализуемой кафедрой физико-технической информатики.

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

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.

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
				32	38				2	
Всего 72 часа / 2 зачётные единицы, из них: - контактная работа 34 часа - в интерактивных формах 32 часа										
Компетенции ПК-1, ПК-2										

Реализация дисциплины предусматривает практическую подготовку при проведении следующих видов занятий, предусматривающих участие обучающихся в выполнении отдельных элементов работ, связанных с будущей профессиональной деятельностью.

Преподавание дисциплины предусматривает следующие формы организации учебного процесса: лабораторные занятия, самостоятельная работа студента и её контроль преподавателями с помощью заданий, дифференцированный зачёт.

Программой дисциплины предусмотрены следующие виды контроля:

- текущий контроль успеваемости: задания для самостоятельного решения;
- промежуточная аттестация: дифференцированный зачёт.

Общая трудоемкость рабочей программы дисциплины составляет 2 зачетные единицы.

- лабораторные занятия – 32 часа;
- самостоятельная работа обучающегося в течение семестра, не включая период сессии – 38 часов;
- промежуточная аттестация (дифференцированный зачёт) – 2 часа.

Объём контактной работы обучающегося с преподавателем (лабораторные занятия, дифференцированный зачёт) составляет 34 часа.

Работа с обучающимися в интерактивных формах составляет 32 часа (лабораторные занятия).

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

Общая трудоемкость дисциплины составляет 2 зачётные единицы, 72 академических часа.

№ п/п	Раздел дисциплины	Неделя семестра	Виды учебной работы, включая самостоятельную работу студентов и трудоёмкость (в часах)				Консультации перед экзаменом (часов)	Промежуточная аттестация (в часах)	
			Всего	Аудиторные часы		Сам. работа во время занятий (не включая период сессии)			Сам. работа во время промежуточной аттестации
				Лекции	Лабораторные				

					заня- тия				
1	2	3	4	5	6	7	8	9	10
1.	Введение в программирование на языке Python, строки	1	4		2	2			
2.	Списки, кортежи	2	4		2	2			
3.	Ассоциативные массивы – словари. Элементы функционального программирования: list, dict, set comprehensions	3	4		2	2			
4.	Основы объектно-ориентированного программирования. Классы.	4	4		2	2			
5.	Наследование, статические методы и поля класса	5	4		2	2			
6.	Библиотека NumPy для работы с матрицами	6-7	4		2	2			
7.	Контрольная работа по теме «Библиотека NumPy для работы с матрицами»	7	4		2	2			
8.	Библиотека pandas для работы с таблицами с разнородными данными	8	4		2	2			
9.	Графические библиотеки Matplotlib и Bokeh	9	4		2	2			
10.	Библиотека SymPy для символьных вычислений	10	4		2	2			
11.	Библиотека Numba для ускорения выполнения кода	11	4		2	2			
12.	Численное интегрирование	12	4		2	2			
13.	Решение обыкновенных дифференциальных уравнений	13	4		2	2			

14.	Фреймворк PyQt5 для создания GUI-приложений	14	6		2	4			
15.	Библиотека PyQtGraph для построения графиков под PyQt5	15	6		2	4			
16.	Фоновое выполнение задач под PyQt5	16	6		2	4			
17.	Дифференцированный зачёт	17	2						2
Всего			72		32	38			2

Программа лабораторных 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 Bokeh (2 часа)

Построение графиков, гистограмм, подготовка их для добавления в научные публикации. Отличия библиотеки Bokeh 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 (38 часов)

Перечень занятий на СРС	Объем, час
Подготовка к лабораторным занятиям.	32
Подготовка к дифференцированному зачёту	6

5. Перечень учебной литературы.

5.1. Основная литература

1. **Элбон, Крис.** Машинное обучение с использованием Python. Сборник рецептов. [Практические решения от предобработки до глубокого обучения : пер. с англ.] / Крис Элбон Санкт-Петербург : БХВ-Петербург, 2019

5.2. Дополнительная литература

1. Лутц, Марк. Изучаем Python : [эффективное объектно-ориентированное программирование : уровень подготовки читателей: средний : пер. с англ.] / Марк Лутц. 3-е изд. Москва ; Санкт-Петербург : Символ, 2009. 843 с. : ил., табл. ; 24 см.
2. Саммерфилд, Марк. Программирование на Python 3 : подробное руководство : [пер. с англ.] / Марк Саммерфилд. Санкт-Петербург ; Москва : Символ, 2009. 607 с. : ил. ; 23 см. (High tech) .
3. Прохоренко, Николай Анатольевич. Python 3 и PyQt 5 : разработка приложений / Николай Прохоренко, Владимир Дронов. Санкт-Петербург : БХВ-Петербург, 2017. 832 с. : ил. ; 23 см.
4. Мэттиз, Эрик. Изучаем Python : программирование игр, визуализация данных, веб-приложения : [пер. с англ.] / Эрик Мэттиз. 2-е изд. Санкт-Петербург [и др.] : Питер, 2018. 491, [2] с. : ил., табл. ; 23 см. (Библиотека программиста) . ISBN 978-5-4461-0479-6.
5. Седер, Наоми. Python : экспресс-курс : [пер. с англ.] / Наоми Седер. 3-е изд. Санкт-Петербург [и др.] : ПИТЕР, 2019. 480 с. : ил., табл. ; 23 см. (Библиотека программиста) .

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

Самостоятельная работа студентов поддерживается следующими учебными пособиями:

1. Соловьев, Игорь Алексеевич (д-р физ.-мат. наук; 1949-). Вычислительная математика на смартфонах, коммуникаторах и ноутбуках с использованием программных сред Python : учебное пособие : [для студентов инженерных специальностей вузов] / И.А. Соловьёв, А.В. Червяков, А.Ю. Репин. Санкт-Петербург [и др.] : Лань, 2011. 265 с. : ил., табл. ; 21 см. (Учебники для вузов, Специальная литература) .

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

Для освоения дисциплины используются следующие ресурсы:

- электронная информационно-образовательная среда НГУ (ЭИОС);
- образовательные интернет-порталы;

- информационно-телекоммуникационная сеть Интернет.

Интернет-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>

7.1 Современные профессиональные базы данных

Не используются.

7.2. Информационные справочные системы

Не используются.

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 и MS Office.

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 используются специальные помещения:

1. Учебные аудитории для проведения занятий лекционного типа, практических занятий, курсового проектирования (выполнения курсовых работ), групповых и индивидуальных консультаций, текущего контроля, промежуточной и итоговой аттестации.

2. Помещения для самостоятельной работы обучающихся.

Учебные аудитории укомплектованы специализированной мебелью и техническими средствами обучения, служащими для представления учебной информации большой аудитории.

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

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.

Порядок проведения текущего контроля и промежуточной аттестации по дисциплине

Текущий контроль

Текущий контроль осуществляется в ходе семестра путем проверки решения задач студентами 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 заявленные компетенции ПК-1 и ПК-2 сформированы 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 форме практического занятия. Практическое задание подбирается таким образом, чтобы проверить уровень сформированности компетенций ПК-1 и ПК-2.

Вывод об уровне сформированности компетенций принимается преподавателем. Положительная оценка ставится, когда все компетенции освоены не ниже порогового уровня. Оценки «отлично», «хорошо», «удовлетворительно» означают успешное прохождение промежуточной аттестации.

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

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	ПК 1.1 ПК 2.1	Уровень знаний ниже минимальных требований. Имеют место грубые ошибки.	Минимально допустимый уровень знаний. 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	ПК 1.2 ПК 2.2	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)	ПК 1.3 ПК 2.3	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.

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 дифференцированный зачёт

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.

Оценочные материалы по промежуточной аттестации, предназначенные для проверки соответствия уровня подготовки по дисциплине требованиям СУОС, хранятся на кафедре-разработчике РПД в печатном и электронном виде.

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

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