

**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 автоматизации физико-технических исследований**



УТВЕРЖДАЮ  
Декан ФФ, д.ф.-м.н  
В.Е.Блинов  
2022 г.

**Рабочая программа дисциплины**

**ПРАКТИЧЕСКОЕ ПРОГРАММИРОВАНИЕ**

направление подготовки: **03.03.02 Физика**  
направленность (профиль): **Общая и фундаментальная физика**

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
3	108		64		42				2	
Всего 108 часов / 3 зачетные единицы, из них: - контактная работа 66 часов										
Компетенции ОПК-3										

Ответственный за образовательную программу  
д.ф.-м.н., проф.

С. В. Цыбуля

Новосибирск, 2022

## Содержание

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

## Аннотация

к рабочей программе дисциплины «Практическое программирование»

Направление: 03.03.02 Физика

**Направленность (профиль): Общая и фундаментальная физика**

Программа дисциплины «Практическое программирование» составлена в соответствии с требованиями СУОС к уровню бакалавриата по направлению подготовки 03.03.02 Физика, направленность «Общая и фундаментальная 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 автоматизации физико-технических исследований. Дисциплина изучается студентами **второго** курса физического факультета в качестве дисциплины по выбору.

Цель дисциплины – получение базовых знаний о применении языка программирования Python при решении практических задач в области физики.

Результаты освоения образовательной программы (компетенции)	Индикаторы	Результаты обучения по дисциплине
<p><b>ОПК-3.</b> Способен понимать принципы работы современных информационных технологий и использовать их для решения задач профессиональной деятельности.</p>	<p><b>ОПК - 3.1.</b> Применяет различные источники информации для решения задач профессиональной сферы деятельности.</p> <p><b>ОПК – 3.2.</b> Применяет основные приемы, возможности и правила работы со стандартными и специализированными программными продуктами при решении профессиональных задач.</p> <p><b>ОПК – 3.3.</b> Применяет методологию поиска научной и технической информации в сети Интернет и специализированных базах данных.</p>	<p><b>Знать</b> основы работы с файловым вводом-выводом; принципы разработки приложений на языке Python; основные библиотеки для научных и инженерных расчетов; инструменты для создания статических графических приложений: графики; визуализаций распределений данных, построение гистограмм; инструменты для создания динамических графических приложений.</p> <p><b>Уметь</b> оценивать сложность алгоритмов по времени; применять существующие библиотеки для решения практических задач; производить символьные и численные расчеты заданных выражений и функций; ставить корректные технические задания.</p> <p><b>Владеть:</b> принципами отображения динамичных графических сцен, с оценкой возможности их отображения в режиме реального времени; принципами моделирования физических задач на ПК, в том числе основными численными</p>

Результаты освоения образовательной программы (компетенции)	Индикаторы	Результаты обучения по дисциплине
		методами решения математических алгоритмов.

Дисциплина рассчитана на **один** семестр (**3-й**). Преподавание дисциплины предусматривает следующие формы организации учебного процесса: практические занятия, самостоятельная работа студента, дифференцированный зачет.

Программой дисциплины предусмотрены следующие виды контроля:

- Текущий контроль: решение задач из задания для самостоятельного решения;
- Промежуточная аттестация: дифференцированный зачет.

Общая трудоемкость рабочей программы дисциплины составляет **108** академических часов / **3** зачетные единицы.

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

Цель дисциплины – получение базовых знаний о применение языка программирования Python при решении практических задач в области физики.

Результаты освоения образовательной программы (компетенции)	Индикаторы	Результаты обучения по дисциплине
<p><b>ОПК-3.</b> Способен понимать принципы работы современных информационных технологий и использовать их для решения задач профессиональной деятельности.</p>	<p><b>ОПК - 3.1.</b> Применяет различные источники информации для решения задач профессиональной сферы деятельности.</p> <p><b>ОПК – 3.2.</b> Применяет основные приемы, возможности и правила работы со стандартными и специализированными программными продуктами при решении профессиональных задач.</p> <p><b>ОПК – 3.3.</b> Применяет методологию поиска научной и технической информации в сети Интернет и специализированных базах данных.</p>	<p><b>Знать</b> основы работы с файловым вводом-выводом; принципы разработки приложений на языке Python; основные библиотеки для научных и инженерных расчетов; инструменты для создания статических графических приложений: графики; визуализаций распределений данных, построение гистограмм; инструменты для создания динамических графических приложений.</p> <p><b>Уметь</b> оценивать сложность алгоритмов по времени; применять существующие библиотеки для решения практических задач; производить символьные и численные расчеты заданных выражений и функций; ставить корректные технические задания.</p> <p><b>Владеть:</b> принципами отображения динамических графических сцен, с оценкой возможности их отображения в режиме реального времени; принципами моделирования физических задач на ПК, в том числе основными численными методами решения математических алгоритмов.</p>

## 2. Место дисциплины в структуре образовательной программы

Курс реализуется в осеннем семестре 2-го курса для бакалавров, обучающихся по направлению подготовки 03.03.02 Физика. В результате прохождения курса студенты отделения общей и фундаментальной физики физического факультета должны овладеть принципами моделирования физических задач на ПК, применения ПК для расчета физических задач, а также научиться оценивать трудозатраты по программной реализации задачи.

Для спешного освоения курса необходимо знание английского языка на уровне чтения текста и основ линейной алгебры.

После изучения курса студенты могут продолжить изучать программирование на курсах, посвященных объектно-ориентированным языкам или практическому программированию в различных приложениях.

## 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
3	108		64		42				2	
Всего 108 часов / 3 зачетных единицы, из них: - контактная работа 66 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, дифференцированный зачет.

Программой дисциплины предусмотрены следующие виды контроля:

Текущий контроль: опрос студентов в начале каждого занятия, решение задач

Промежуточная аттестация: дифференцированный зачет

Общая трудоемкость рабочей программы дисциплины составляет **108** академических часов/**3** зачетные единицы:

- лабораторные занятия – 64 часа.
- самостоятельная работа обучающегося в течение семестра, не включая период сессии – 42 часа;
- промежуточная аттестация (дифференцированный зачет) – 2 часа.

Объём контактной работы обучающегося с преподавателем (практические занятия, дифференцированный зачет) составляет 66 часов.

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

№ п/п	Раздел дисциплины	Неделя семестра	Виды учебных занятий, включая самостоятельную работу студентов и трудоемкость (в часах)				Промежуточная аттестация (в часах)
			Всего	Аудиторные часы		Сам. работа в течение семестра (не включая период сессии)	
				Лекции (кол-во часов)	Практические занятия (кол-во часов)		
1	2	3	4	5	6	7	8
1	Вводный блок: интерпретатор, исполнение кода, переменные и их основные типы	1			4		
2	Интерактивная графическая веб-оболочка Jupyter: редактирование и запуск программ, отображение результатов вычислений, работа с разметкой LaTeX	2			4		
3	Основные языковые конструкции, подключение библиотек, конструкция import, использование отладчика	3			8	6	
4	Объявление функций и их реализация (выражение def), аргументы функций, возвращение значения.	4			8	6	
5	Списки и работа с ними. Построение сложных списков, строковое представление списков	5-6			8	6	
6	Библиотека NumPy для работы с многомерными массивами	7-8			8	6	
7	Библиотека SciPy для выполнения научных и инженерных расчётов	9-10-11			8	6	
9	Библиотека Matplotlib и модуль Pyplot для построения графиков	12-13-14			8	6	
10	Библиотека SymPy: решение уравнений, вычисление пределов, дифференцирование/интегрирование, дифференциальные уравнения и линейная алгебра	15-16			8	6	
11	Дифференцированный зачет	17					2

<b>ИТОГО</b>		<b>108</b>	<b>64</b>	<b>42</b>	<b>2</b>
--------------	--	------------	-----------	-----------	----------

**Программа практических занятий (64 час)**  
**Длительность каждое занятия – 4 ак. часа.**

1	<b>Лабораторное занятие 1.</b> Язык Python. Интерпретатор. Основные конструкции языка и типы данных. Простейшие конструкции и программы.
2	<b>Лабораторное занятие 2.</b> Работа с массивами данных: списки и кортежи. Особенности реализации циклов. Работа со строками.
3	<b>Лабораторное занятие 3.</b> Подключение внешних библиотек. Утилита PIP. Среды разработки на Python.
4	<b>Лабораторное занятие 4.</b> Графика. Построение диаграмм. Масштабирование. Различные системы координат.
5	<b>Лабораторное занятие 5.</b> Статистические функции обработки данных. Оперирование с большими данными. Сортировка. Поиск.
6	<b>Лабораторное занятие 6.</b> Работа с текстом. Встроенные возможности. Преобразование текста и применение правил русского языка для обработки.
7	<b>Лабораторное занятие 7.</b> Матрица. Возможности Python. Комплексные числа. Работа с большими матрицами.
8	<b>Лабораторное занятие 8.</b> Символьные вычисления в Python. Решение задач в численном и символьном виде.
9	<b>Лабораторное занятие 9.</b> Динамическая графика. Отображение меняющихся величин на графиках.
10	<b>Лабораторное занятие 10.</b> Фурье анализ данных. Фильтрация и обратное преобразование Фурье. Визуализация.
11	<b>Лабораторное занятие 11.</b> Решение задач аппроксимации и сглаживания экспериментальных данных. Функция приближений.
12	<b>Лабораторное занятие 12.</b> Получение данных из сети Интернет программой при решении задач.
13	<b>Лабораторное занятие 13.</b> Решение задачи анализ солнечной активности по открытым данным со спутников НАСА.
14	<b>Лабораторное занятие 14.</b> Численное моделирование физических задач на примере полета тела в поле тяжести.
15	<b>Лабораторное занятие 15.</b> Моделирование движения сложных систем из тысяч объектов.
16	<b>Лабораторное занятие 16.</b> Применение принципов Объектно-ориентированного программирования для описания частиц.



## Самостоятельная работа студентов (42 часа)

Перечень занятий на СРС	Объем, час
Подготовка к практическим занятиям.	22
Изучение разделов дисциплины по учебной литературе, в том числе вопросов, не освещаемых на лекциях.	20

### 5. Перечень учебной литературы

#### 5.1. Основная литература

1. **Вирт, Никлаус.** Алгоритмы и структуры данных / Н. Вирт ; пер. с англ. Д.Б. Подшивалова. Москва : Мир, 1989 360 с. : ил. ; 20 см. Пер. изд.: Algorithms and data structure / Niklaus Wirth. - Englewood Cliffs: Prentice-Hall, 1986

#### 5.2. Дополнительная литература

1. Лысаков К.Ф., Дунаев А.А. Структурный и объектно-базированный подходы в программировании на примере языков С и С++: Учеб. пособие / Новосиб. гос. ун-т. Новосибирск, 2007.

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

2. Вирт Н. Алгоритмы и структуры данных. Невский Диалект, 2005.
3. Майкл Доусон Программируем на Python // Питер, 2018.

### 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.

Взаимодействие обучающегося с преподавателем (синхронное и (или) асинхронное) осуществляется через личный кабинет студента в ЭИОС.

#### 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, MS Office и интерпретатор языка Python.

Использование специализированного программного обеспечения для изучения дисциплины не требуется.

## **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, лабораторных занятий и промежуточной 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**

### **10.1 Порядок проведения текущего контроля и промежуточной аттестации по дисциплине**

#### **Текущий контроль**

Текущий контроль осуществляется в ходе семестра путем опроса студентов в начале каждого занятия на темы, рассмотренные на предыдущем занятии.

#### **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 использования языков описания аппаратуры в профессиональной деятельности.

Окончательная оценка работы студента в течение семестра происходит на зачете с оценкой. Зачет с оценкой проводится в конце семестра в сессию в устной форме. Вопросы подбираются 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

<b>Индикатор</b>	<b>Результат обучения по дисциплине</b>	<b>Оценочные средства</b>
------------------	---	---------------------------

<p><b>ОПК - 3.1.</b> Применяет различные источники информации для решения задач профессиональной сферы деятельности.</p>	<p><b>Знать</b> основы работы с файловым вводом-выводом; принципы разработки приложений на языке Python; основные библиотеки для научных и инженерных расчетов; инструменты для создания статических графических приложений: графики; визуализаций распределений данных, построение гистограмм; инструменты для создания динамических графических приложений.</p>	<p>Опрос студентов, дифференцированный зачет.</p>
<p><b>ОПК – 3.2.</b> Применяет основные приемы, возможности и правила работы со стандартными и специализированными программными продуктами при решении профессиональных задач.</p>	<p><b>Уметь</b> оценивать сложность алгоритмов по времени; применять существующие библиотеки для решения практических задач; производить символьные и численные расчеты заданных выражений и функций; ставить корректные технические задания.</p>	<p>Опрос студентов, дифференцированный зачет.</p>
<p><b>ОПК – 3.3.</b> Применяет методологию поиска научной и технической информации в сети Интернет и специализированных базах данных.</p>	<p><b>Владеть:</b> принципами отображения динамичных графических сцен, с оценкой возможности их отображения в режиме реального времени; принципами моделирования физических задач на ПК, в том числе основными численными методами решения математических алгоритмов.</p>	<p>Опрос студентов, дифференцированный зачет.</p>

## 10.2 Описание критериев и шкал оценивания индикаторов достижения результатов обучения по дисциплине «Практическое программирование».

Таблица 10.2

Критери и оценива ния результ атов обучени я	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.	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	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-3.3	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.

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

#### Задачи для решения с целью получения базовых навыков (Блок 1)

- Сложить 2 и 2, напечатав результат.
- Возвести в квадрат мнимую единицу
- Вычислить квадратный корень из (35344)
- Вычислить арксинус ( $\pi/4$ )
- Сложить все целые числа от одного до ( $2^{25}$ )
- Вычислить факториал (1000)
- Сделать список из (3872) нулей
- Найти элементы списка, отсутствующие в другом
- Удвоить все нечетные числа в списке
- Вывести названия всех клеточек поля для игры в морской бой
- Вычислить, сколько четных чисел от 1 до миллиона начинаются на 793, но при этом не кратны 7
- Считать строку с клавиатуры и развернуть
- Подсчитать количество букв 'ы' в файле
- Найти самое длинное слово в файле
- Найти наибольший общий делитель двух чисел

- Сделать букву заглавной (найти, как)
- Промоделировать Броуновское движение миллиона частиц и вывести зависимость среднего квадрата отклонения от времени
- Найти в директории самый длинный '.txt' файл
- Считать число и вывести его прописью

## Задачи для решения (Блок 2)

1. Решить квадратное уравнение.
  - \* Целочисленные коэффициенты задаются пользователем с клавиатуры.
  - \* Решением занимается отдельная функция.
  - \* Разветвление решения на случай линейного и квадратного уравнения.
  - \* Обобщить решение задачи для комплексных корней.
  - \* Что возвращает ваша функция? Унифицируйте возвращаемый тип.
2. Печать всех простых чисел, не превышающих  $N$ .
  - \*  $N$  задается пользователем с клавиатуры.
  - \* Точная реализация математического определения: "имеет 2 делителя" должна быть реализована внутри функции `is_prime`.
  - \* Сократить программу до одной строки.  
Список простых чисел должен быть получен через списковое включение с фильтрацией. Не допускается использование циклов в явном виде.
3. Вычисление числа  $\pi$  с заданной точностью (кол-во знаков после запятой).
  - \* Не допускается использование циклов в явном виде.
4. Программа для решения уравнения вида  $f(x) = 0$  методом Ньютона.
  - \* Функция  $f(x)$  задается в коде или вводится с клавиатуры.
  - \* Производная функции  $f(x)$  вычисляется аналитически.
  - \* Начальное приближение и требуемая точность вычисления задаются с клавиатуры.
  - \* Промежуточные точки и касательные отобразить графически.
5. Вычисление интеграла функции  $f(x)$  аналитически и численно.
  - \* Функция  $f(x)$  задается в коде или вводится с клавиатуры.
  - \* Пределы интегрирования вводятся с клавиатуры.
  - \* Интеграл функции вычислить двумя способами:
    1. аналитически (`sympy`),
    2. численно (`numpy/scipy`).
6. Вычисление статистик массива.
  - \* Вычислить минимум и максимум.
  - \* Вычислить среднее значение.
  - \* Вычислить среднеквадратическое отклонение.
  - \* Вычислить пятый центральный момент.
  - \* Докажите, что он действительно центральный.
  - \* Вычислить среднее из пяти вышеуказанных чисел.
7. Сортировка.
  - \* Отсортировать массив действительных чисел по возрастанию.
  - \* Отсортировать массив комплексных чисел по модулю.
  - \* Отсортировать массив комплексных чисел по убыванию действительной части.

- \* Отсортировать список строк по длине.
- \* Отсортировать кортеж строк в лексикографическом порядке.
- \* Отсортировать кортеж списков по возрастанию длины.

#### 8. Работа с текстом.

- \* Считать предложение из файла.
- \* Вычислить количество слов в предложении.
- \* Развернуть порядок букв в третьем слове предложения.
- \* В получившемся предложении развернуть порядок слов.
- \* В получившейся строке сделать первую букву заглавной.
- \* Сохранить результат в файл.
- \* Построить гистограмму частотности букв без учета регистра.

#### 9. Приведение текста в соответствие правилам.

- \* Считать строку произвольной длины.
- \* Преобразовать строку следующим образом.
  1. Удалить пробелы, стоящие перед точкой или запятой.
  2. После точек, запятых, двоеточий и т.д. поставить пробел.
  3. Несколько пробелов, следующих подряд, заменить на одиночный.
  4. Первую букву каждого предложения сделать заглавной.
- \* Что нужно сделать, чтобы программа обрабатывала строку на любом языке?

#### 10. Матрицы.

- \* Сгенерировать две случайные матрицы заданного размера.
- \* Вычислить сумму, произведение, определитель и обратные матрицы.
- \* Реализовать сохранение матриц в файл.
- \* Добавить вариант со считыванием матриц из файла.
- \* Выбор формата обосновать, привести плюсы и минусы.
- \* Что нужно сделать, чтобы добавить поддержку матриц комплексных чисел?
- \* Измерить время обращения матриц  $10 \times 10$ ,  $100 \times 100$ ,  $1000 \times 1000$ .

#### 11. Телефонная книга.

- \* Определить тип Person.  
Поля данных: имя, фамилия, номер телефона, дата рождения.
- \* Предоставить возможность сортировать список Person по любому из полей.
- \* Реализовать считывание телефонной книги из файла в текстовом формате.
- \* Реализовать сохранение телефонной книги в файл в совместимом формате.
- \* Реализовать сохранение и считывание через сериализацию (pickle).

#### 12. Определить НОД и НОК

- \* Считать  $\$N\$$  натуральных чисел.
- \* Вычислить и вывести наибольший общий делитель введенных чисел.
- \* Вычислить и вывести наименьшее общее кратное введенных чисел.
- \* Как обобщить произвольную операцию на  $\$N\$$  аргументов?

#### 13. Перевод чисел между системами счисления.

- \* Считать число, старое основание и новое основание.
- \* Вывести число в системе счисления с новым основанием.

#### 14. Найти наибольшую общую подпоследовательность двух списков.

#### 15. Найти вероятность того, что на цифровых часах с секундами

будут одновременно отображаться и "16", и "37".

16. Вычислить  $\pi$  методом Монте-Карло.

- \* Разыграть много случайных  $x$  и  $y$  в квадрате  $[-1, 1] \times [-1, 1]$ .
- \* Подсчитать долю попавших в круг с радиусом 1.
- \* По этой доле оценить значение  $\pi$ .

17. Повторение иностранных слов.

- \* Завести словарь из слов и их переводов.
- \* Запрашивать у пользователя переводы случайных слов.
- \* Слова, переводы которых введены правильно, больше не спрашивать.
- \* Реализовать сохранение состояния между запусками.

18. Подбор бинарного оператора.

- \* Считать целые числа `A`, `B` и `C`.
- \* Вывести всех возможные бинарные операторы Python 3, для которых истинно `A operator B == C`.

19. Обрезание полей у изображения.

- \* Раздобыть или нарисовать однобитное изображение.
- \* Превратить его в булев массив `numpy`.
- \* Обрезать с краев поля из строк и столбцов исключительно ложных пикселей, но  $N \times N$  пикселей оставить.

20. Поиск наибольшего прямоугольного отверстия.

- \* Раздобыть или нарисовать однобитное изображение.
- \* Превратить его в булев массив `numpy`.
- \* Найти и отобразить наибольшую по площади прямоугольную ложную область.

21. Построить график функции в полярных координатах:

$$r(\phi) = 2 \sin\{\phi\} + \frac{\sin\{\phi\}}{\sqrt{|\cos\{\phi\}|}}$$
$$\left\{ \begin{array}{l} \sin\{\phi\} + 7/3 \\ \end{array} \right\}, \pi \leq \phi \leq \pi$$

22. Моделирование взаимодействия "хищник-жертва" в модели Лотки-Вольтерры.

- \* Задать стартовые значения популяций жертв и хищников  $x$ ,  $y$  и коэффициенты связи  $\alpha$ ,  $\beta$ ,  $\gamma$ ,  $\delta$ .
- \* Решить численно или символьно дифференциальные уравнения  $\frac{dx}{dt} = (\alpha - \beta y) x$ ,  $\frac{dy}{dt} = (-\gamma + \delta x) y$ .
- \* Решения  $x(t)$  и  $y(t)$  отобразить графически.

23. Реализовать клеточный автомат "игра Жизнь" (Conway's Game of Life).

24. Тестирование первой задачи.

- \* Решить первую задачу.
- \* Написать с использованием `nosetests` или `pytest` тесты на все случаи из первой задачи, включая те, что бросают исключения.

25. Конвертер валют, который сам узнает курс.
- \* Запросить у пользователя пару валют и сумму.
  - \* Перевести сумму из одной валюты в другую.
  - \* Курс перевода запросить из сети Интернет автоматически.

### Задачи для решения (Блок 3)

1. Фурье-фильтрация сигнала (*обязательно для любой положительной оценки*).
  - \* Смоделировать сигнал из трёх некротных частот, добавить случайный шум.
  - \* Вычислить спектр сигнала, нарисовать график, оси в реальных единицах (Гц).
  - \* Избавиться от шума в исходном сигнале путём вырезания "лишних" частот в фурье-спектре.
  - \* Построить график исходного, зашумлённого и фильтрованного сигнала с легендой.
2. Аппроксимация методом наименьших квадратов.
  - \* Взять модельный сигнал ( $a \cdot \sin(b \cdot x)$ ,  $x=0..2 \cdot \pi$ ), добавить случайный шум, методом наименьших квадратов найти параметры  $a$ ,  $b$ .
  - \* Построить на графике исходный сигнал (линия), зашумлённые данные (точки), результат аппроксимации (линия), вывести легенду, найденные значения.
  - \*\* Добавить случайные выбросы амплитудой порядка  $a \cdot 3$ , применить взвешенный МНК.
  - \*\* Сравнить действие различных весовых функций, построить график.
3. Лабораторная работа 4.4.
  - \* По набору сохранённых циклов перемагничивания построить основную кривую намагничивания.
  - \* Сгладить кривую, получить зависимость дифференциальной магнитной проницаемости.
  - \* По петле гистерезиса предельного цикла вычислить потери на перемагничивание (площадь под кривой) для всех измеренных образцов.
4. Лабораторная работа 6.3.
  - \* По измеренным осциллограммам определить коэффициенты трения для каждой из трубок используя МНК.
  - \* Времена пролёта через катушки определить по минимуму (максимуму) производной исходного сигнала, для подавления шумов использовать фурье-фильтр.
  - \*\* Времена пролёта определить путём вейвлет-анализа, обосновать выбор базисной функции, сравнить погрешности с предыдущим пунктом.
  - \*\* Сравнить вейвлет-спектрограммы для сигналов разной зашумлённости, пояснить смысл величин, отложенных по осям.
5. Анализ солнечной активности.
  - \* По открытым данным со спутников НАСА произвести анализ уровня гамма-излучения от солнца.
  - \* Файлы с данными выкачивать в зависимости от введённого пользователем периода (1986-2017 годы):  
<https://www.ngdc.noaa.gov/stp/satellite/goes/dataaccess.html>  
[https://satdat.ngdc.noaa.gov/sem/goes/data/new\\_avg/](https://satdat.ngdc.noaa.gov/sem/goes/data/new_avg/)
  - \* Организовать кэш (выкачанные данные повторно не загружать).



- \* С целью уменьшения числа отображаемых точек реализовать усреднение по выбранном временному интервалу (час, сутки).
- \* Используя фурье-преобразование определить характерные циклы.

#### 6. Приближения функций.

- \* Функция  $F(x)$  задается аналитически, с клавиатуры.
- \* Приблизить функцию  $F(x)$  полиномом  $N$ -ой степени.
- \* Пользователю вывести коэффициенты, сравнительный график.
- \* Функция  $G(x, p)$  задается аналитически, с клавиатуры.
- \* Приблизить функцию  $F(x)$  функцией  $G(x, p)$  за счет варьирования  $p$ .
- \* Пользователю вывести оптимальное значение  $p$ , сравнительный график.

#### 7. Численно промоделировать полет тела.

- \* Рассчитайте траекторию тела, летящего в поле тяжести по параболе.
- \* Отобразите траекторию на графике.
- \* Решите эту же задачу аналитически (sumru), сравните результат.
- \* Повторите тот же анализ, добавив модель трения о воздух.
- \* Оптимизируйте угол вылета тела (при заданном модуле начальной скорости)

так, чтобы пересечение уровня начальной высоты произошло как можно дальше от начальной точки.

#### 8. Сравнить производительность кода с использованием Numpy, Cython, Numba.

- \* Реализовать либо игру "Жизнь", либо моделирование Броуновского движения.
- \* Ускорить программу через Cython.
- \* Ускорить программу через Numba.
- \* Написать программу через Numpy/Scipy.
- \* Сравнить производительность.

#### 9. Обработать данные из HDF5-файла.

- \* Получить .h5-файл от преподавателя.
- \* Научиться ориентироваться внутри .h5-файла и узнать, что там.
- \* Построить распределение пространственной плотности пучка в плоскости  $z$ - $r$  (цилиндрическая геометрия).
- \* Проинтегрировать ток пучка ( $q \cdot rz / m$ ) по радиусу до 200 микрон.
- \* Построить фурье-спектр тока пучка и подписать продольные частоты.

### Задачи для решения (Блок 4)

#### 1. Численно промоделировать движение тысячи частиц.

- \* Начальные положения:  $x$  равномерно распределен от  $-1$  до  $0$ ;  $y$  — от  $-1$  до  $1$ .
- \* Начальные скорости равномерно распределены по углу, постоянны по модулю.
- \* Частицы отражаются от "стенок коробки"  $(-1, -1) — (-1, 1) — (1, 1) — (1, -1) — (-1, -1)$ , за исключением "отверстия"  $(1, -0.1) — (1, 0.1)$ .
- \* Нарисовать положения всех частиц через время  $t$ .
- \* Либо обойтись не более чем одним циклом на программу, либо ускорять вычисления при помощи Cython или Numba.
- \*\* Анимировать движение частиц.
- \*\* Сделать два отверстия.
- \*\* Дорисовать частицам стрелки — векторы скоростей.
- \*\* Нарисовать двумерную гистограмму плотности частиц через время  $t$ .

2. Напоминания по почте.
  - \* Читать файл со строками вида '2018-12-01 19:00:00 Свидание с Машей'.
  - \* За час до наступления указанного времени ('2018-12-01 19:00:00') отправлять электронное письмо с указанным текстом ('Свидание с Машей') на электронную почту студента.
  - \* Программа должна поддерживать несколько таких строк в любом порядке.
  - \* Программа должна выводить предупреждения для напоминаний "в прошлом".
  
3. Массовое скачивание видео с YouTube.
  - \* Имя пользователя/канала вводится с клавиатуры (пример: 'kvn').
  - \* Вывести, пронумеровав, все видео с этого канала.
  - \* Читать строку с номерами видео через запятую, с поддержкой интервалов (пример: "1,3,6-9").
  - \* Скачать эти видео.
  
4. Датировка снимка Луны.
  - \* Программа запускается с указанием имени файла в качестве аргумента.
  - \* Читать дату совершения снимка из EXIF-тега указанного файла.
  - \* Определить фазу Луны по заданной дате.
  - \* Определить фазу Луны по фотоснимку.
  - \* На основании соответствия фаз вывести мнение о правдоподобности датировки снимка.
  
5. Автоматизированный тестировщик других программ.
  - \* Напишите программу, которая запускает программу, вычисляющую число  $\pi$  с заданной точностью, передает ей искомую точность, считывает ответ и сравнивает его с заранее известным.
  - \* Напишите программу, которая тестирует другую произвольную программу:
    1. запускает произвольную программу,
    2. подает на вход указанные данные,
    3. считывает результат и сравнивает с эталонным.
  - \* Формат конфигурационного файла, в котором будет прописано, что запускать, что подавать на ввод и что ожидать в результате, придумать самостоятельно и обосновать.
  - \* Продемонстрировать работу для как минимум двух программ на разных языках программирования: компилируемом и интерпретируемом.
  - \*\* Реализовать подачу на вход дочерней программы случайных данных.  
Провести поиск данных, приводящих к аварийному завершению программы.
  
6. Кеширующий декоратор.
  - \* Реализовать кеширующий декоратор, который бы запоминал результат выполнения обернутой функции для разных значений аргумента, и, в случае повторного вызова с теми же аргументами, возвращал бы уже вычисленный результат.
  - \* Кеш вычисленных результатов хранить в замыкании.
  - \* Обобщить на случай произвольного количества аргументов.
  - \* Добавить поддержку произвольных словарных аргументов.
  - \* Добавить ограничение на размер кеша, придумать и реализовать политику "выкидывания" результатов.
  - \* Сделать размер кеша параметром декоратора.
  - \* Для каких функций оправдано кеширование?
  
7. RPC.
  - \* Реализовать механизм выполнения функций на другом компьютере.

Идеальный интерфейс: `another_pc.numpy.sum(a, b)`,  
где `another_pc` - Ваш модуль или объект,  
`numpy` - модуль на удаленном компьютере,  
`sum` - функция на удаленном компьютере,  
а вот `a` и `b` - объекты с Вашего компьютера,  
которые пропутешествуют на удаленный компьютер и сложатся там,  
а результат выполнения вернется обратно.

**Итоговая оценка на зачете выставляется с учетом решенных задач во время семестра.**

Критерии оценок:

- «Удовлетворительно»  
15 из Блока №1 + задача №1 из Блока №2
- «Хорошо»  
Критерии на «3» + 1 задача из Блока №2 + 1 задача из Блока №3
- «Отлично»  
Критерии на «4» + расширенное решение любой задачи из 2 или 3 блока.

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 «Практическое программирование»  
по направлению подготовки 03.03.02 Физика  
Профиль «Общая и фундаментальная физика»**

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