

**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



РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

Применение Python в физических исследованиях

направление подготовки: 03.03.01 Прикладные математика и физика

Форма обучения: очная

| Семестр | Общий объем | Виды учебных занятий (в часах) | | | | | Промежуточная аттестация (в часах) | | | | |
|--|----------------|--------------------------------|---|-------------------------|----------------------------------|--|--|---|-------|------------------------------|---------|
| | | Лекции | Контактная работа обучающихся с преподавателем | | | Самостоятельная работа, не включая период сессии | Самостоятельная подготовка к промежуточной аттестации | Контактная работа обучающихся с преподавателем | | | |
| | | | Лабораторные занятия | Практические занятия | Консультации в период занятий | | | Консультации | Зачет | Дифференцирован ный зачет | Экзамен |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
| 3 | 108 | 32 | | 32 | | 42 | | | | 2 | |
| Всего 108 часов /3 зачётные единицы, из них: - контактная работа 66 часов | | | | | | | | | | | |
| Компетенции ОПК-6 | | | | | | | | | | | |

Ответственный за образовательную программу

д.ф.-м.н., проф.

С. В. Цыбуля

Новосибирск, 2023

Содержание

| | |
|--|----|
| 1. Перечень планируемых результатов обучения по дисциплине, соотнесённых с установленными в программе индикаторами достижения компетенций | 3 |
| 2. Место дисциплины в структуре образовательной программы | 4 |
| 3. Трудоемкость дисциплины в зачетных единицах с указанием количества академических часов, выделенных на контактную работу обучающегося с преподавателем (по видам учебных занятий) и на самостоятельную работу обучающегося | 5 |
| 4. Содержание дисциплины, структурированное по темам (разделам) с указанием отведенного на них количества академических часов и видов учебных занятий..... | 6 |
| 5. Перечень учебной литературы | 9 |
| 6. Перечень учебно-методических материалов по самостоятельной работе обучающихся.. | 9 |
| 7. Перечень ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины | 9 |
| 8. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине | 9 |
| 9. Материально-техническая база, необходимая для осуществления образовательного процесса по дисциплине | 10 |
| 10. Оценочные средства для проведения текущего контроля и промежуточной аттестации по дисциплине..... | 11 |
| Аннотация к рабочей программе дисциплины «Применение Python в физических исследованиях» | 15 |

Приложение 1 Аннотация по дисциплине

Приложение 2 Оценочные средства по дисциплине

1. Перечень планируемых результатов обучения по дисциплине, соотнесённых с установленными в программе индикаторами достижения компетенций

Цель дисциплины – обучение студентов физических специальностей разрабатывать и применять программное обеспечение в своих научных работах. Данный курс позволит не только освоить базовые и продвинутые аспекты разработки программного обеспечения, но и научит использовать его для решения конкретных физических задач. Программа курса начинается с основ операционной системы GNU/Linux, синтаксиса языка Python и методологий разработки кода, а затем плавно переходит к методам численного анализа, обработке и визуализации данных, машинному обучению и нейронным сетям. Особое внимание уделено оптимизации кода и использованию высокопроизводительных вычислений для решения физических задач.

| Результаты освоения образовательной программы (компетенции) | Индикаторы | Результаты обучения по дисциплине |
|--|---|---|
| ОПК-6. Способен разрабатывать алгоритмы и компьютерные программы, пригодные для практического применения | ОПК-6.1 Применяет алгоритмические языки программирования, операционные системы и оболочки, современные среды разработки программного обеспечения ОПК-6.2 Составляет алгоритмы, пишет и отлаживает коды на языке программирования, тестирует работоспособность программы, интегрирует программные модули. | Знать основы операционной системы GNU/Linux, основы языка программирования Python, основы методологий разработки технологических процессов сборки, настройки и развёртывания программного обеспечения (DevOps и SRE), основные инструменты на Python для обработки и анализа данных, основы машинного обучения и нейронных сетей, различные системы управления базами данных (СУБД). Уметь использовать интерактивную оболочку IPython и применять язык Python для решения физических задач, автоматизировать процессы разработки и обеспечивать надежность систем, применять СУБД, измерять время выполнения кода, использовать параллельные вычисления и работать с GPU для ускорения обработки данных и вычислений. |

2. Место дисциплины в структуре образовательной программы

Программа дисциплины «Применение Python в физических исследованиях» составлена в соответствии с требованиями ФГОС ВО по направлению подготовки 03.03.01 Прикладная математика и 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. Дисциплина изучается студентами третьего курса физического факультета.

Для обучения по дисциплине «Применение Python в физических исследованиях» предшествуют дисциплины, читаемые на 1-2 курсах. Среди таких дисциплин можно перечислить: Основы математического анализа (1-й курс), Линейная алгебра и геометрия (1-й курс), Основы программирования (1-й курс), Компьютерное моделирование физических явлений (2-й курс), Обыкновенные дифференциальные уравнения (2-й курс).

Основные цели курса «Применение Python в физических исследованиях»: освоить основы операционной системы Linux и научиться программировать на Python для решения физических задач; узнать основы работы с базами данных и популярными инструментами для анализа больших данных (такие как NumPy и Pandas). Также курс включает введение в системное администрирование, автоматизацию процессов разработки и основы машинного обучения. Кроме того, затрагиваются способы оптимизации программ для повышения их скорости и эффективности.

Практическое содержание предмета направлено на формирование навыков работы с операционной системой Linux, программированием на Python, управлением базами данных и работой с системами контроля версий. Кроме того, студенты изучат методы машинного обучения для моделирования физических явлений и выполнят проектные работы, интегрирующие полученные навыки для решения комплексных задач в области физики.

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

Трудоемкость дисциплины – 3 з.е. (108 ч)

Форма промежуточной аттестации: 3 семестр – дифференцированный зачет

Таблица 3.1

| № | Вид деятельности | Семестр |
|----|--|---------|
| | | 3 |
| 1 | Лекции, ч | 32 |
| 2 | Практические занятия, ч | 32 |
| 4 | Занятия в контактной форме, ч, из них | 66 |
| 5 | из них аудиторных занятий, ч | 64 |
| 6 | в электронной форме, ч | - |
| 7 | консультаций, час. | |
| 8 | промежуточная аттестация, ч | 2 |
| 9 | Самостоятельная работа, час. | 42 |
| 10 | Всего, ч | 108 |

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

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

| № п/п | Раздел дисциплины | Неделя семестра | Виды учебной работы, включая самостоятельную работу студентов и трудоемкость (в часах) | | | | | Контактная работа обучающегося с преподавателем | Промежуточная аттестация (в часах) | |
|--------------------|--|-----------------|--|-----------------|----------------------|-------------------------|--|--|---------------------------------------|---|
| | | | Всего | Аудиторные часы | | | Самостоятельная подготовка к промежуточной аттестации | | | Самостоятельная работа, не включая период сессии |
| | | | | Лекции | Практические занятия | Лабораторные занятия | | | | |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
| Первый семестр | | | | | | | | | | |
| 1 | Основы ОС Linux | 1-2 | 14 | 5 | 5 | | | 4 | | |
| 2 | Введение в предмет и общие знания компьютерных наук и вычислительной физики | 3-4 | 14 | 5 | 5 | | | 4 | | |
| 3 | Инструменты, практика и основы разработки программного обеспечения | 5-7 | 18 | 5 | 5 | | | 8 | | . |
| 4 | Обработка и анализ больших данных современными инструментами | 8-10 | 18 | 5 | 5 | | | 8 | | |
| 5 | Основные методы и способы профилирования и ускорения программ | 11-13 | 18 | 5 | 5 | | | 8 | | |
| 6 | Разбор примеров кодов и скриптов в физических исследованиях, применяемы в институтах СО РАН | 14-16 | 24 | 7 | 7 | | | 10 | | |
| 7 | Дифференцированный зачет | 17 | 2 | | | | | | | 2 |
| Всего за 1 семестр | | | 108 | 32 | 32 | | | 42 | | 2 |

Лекции (32 часа)

Таблица 4.1

| Наименование темы и их содержание | Объем, час |
|---|---------------|
| <i>Основы ОС Linux</i> 1.1. Устройство 1.2. Инструменты 1.3. Внутренности | 5 |
| <i>Введение в предмет и общие знания компьютерных наук и вычислительной физики</i> 2.1. Введение в алгоритмы 2.2. Основные структуры данных 2.3. Рекурсия и сортировки 2.4. Хеш-функции 2.5. Понятия сходимости, аппроксимации и устойчивости | 5 |
| Инструменты, практика и основы разработки программного обеспечения 3.1. Python 3.2. Прикладное программирование 3.3. Жизненный цикл 3.4. Базы данных | 5 |
| Обработка и анализ больших данных современными инструментами 4.1. Инструменты 4.2. Базовые алгоритмы 4.3. Нейронные сети 4.4. Машинное обучение | 5 |
| Основные методы и способы профилирования и ускорения программ 5.1. Скорость выполнения программ и производительность 5.2. Multithreading and GIL 5.3. Асинхронное выполнение задач 5.4. CUDA | 5 |
| Разбор примеров кодов и скриптов в физических исследованиях, применяемы в институтах СО РАН 6.1. Решение дифференциальных уравнений динамики 6.2. Разностные схемы для расчета задач динамики в электрических и магнитных полях 6.3. Решение задач оптимизации различными методами 6.4. Использование машинного обучения и нейронных сетей для физических задач | 7 |

Практические занятия (32 часа)

| Содержание практического занятия | Объем, час |
|----------------------------------|---------------|
|----------------------------------|---------------|

| | |
|---|---|
| <p>Установка Linux на массив из двух дисков. Файловые системы в Linux, включая их виды и механизмы работы. Методы монтирования и техники управления дисками.</p> <p>Практикум по основным инструментам и утилитам в Linux. Основы работы с командной строкой и быстрая аналитика с помощью базовых инструментов Linux. Планировщик процессов и подсистема виртуальной памяти.</p> | 5 |
| <p>Разработка прикладного ПО на языке Python. Синтаксис языка с высоты птичьего полёта. Интерактивная оболочка IPython. Системы контроля версий и контейнеризации программного обеспечения.</p> <p>Проектирование жизненного цикла ПО. Основные этапы жизни программного обеспечения и как они выглядят на примере реальных проектов.</p> | 5 |
| <p>Особенности работы с СУБД. Реляционные и нереляционные базы данных. Основы языка SQL. Построение простого приложения для визуализации данных.</p> | 5 |
| <p>Обработка и анализ экспериментальных и расчетных данных физического эксперимента. Знакомство с современными инструментами и фреймворками для работы с данными на языке Python.</p> <p>Основы машинного обучения и нейронных сетей на примере конкретных задач. Обучение простого классификатора.</p> | 5 |
| <p>Инструменты для профилирования и ускорения программ с помощью различных модулей. Немного о библиотеке NumPy. Компиляция кода на Python на примере инструментов Numba и Cython.</p> | 5 |
| <p>Основы программирования на графических ускорителях.</p> <p>Параллелизм и конкурентность. Использование потоков для параллельных вычислений на Python. Модули для асинхронного и параллельного программирования на Python.</p> <p>Работа с примерами ПО для моделирования физических явлений</p> | 7 |

Самостоятельная работа студентов (42 ч)

| Перечень занятий на СРС | Объем, час |
|--|------------|
| Изучение теоретического материала, не освещаемого на лекциях | 8 |
| Подготовка к практическим работам | 26 |
| Подготовка к дифференцированному зачету | 8 |

5. Перечень учебной литературы

1. Разработка и применение программного обеспечения в физических исследованиях / Вячеслав Федоров. — URL: <https://fuodorov.github.io/physicist-developer-book/>
2. Документация по языку Python. — URL: <https://docs.python.org/3/>

6. Перечень учебно-методических материалов по самостоятельной работе обучающихся

1. Борзунов, С. В. Алгебра и геометрия с примерами на Python : учебное пособие для вузов / С. В. Борзунов, С. Д. Кургалин. — Санкт-Петербург : Лань, 2020 — 444 с. — ISBN978-5-8114-5489-1. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/149336>
2. Рашка, С. Python и машинное обучение: крайне необходимое пособие по новейшей предсказательной аналитике, обязательное для более глубокого понимания методологии машинного обучения : руководство / С. Рашка ; перевод с английского А. В. Логунова. — Москва : ДМК Пресс, 2017 — 418 с. — ISBN 978-5-97060-409-0. — Текст : электронный // URL: <https://e.lanbook.com/book/100905>

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

7.1 Ресурсы сети Интернет

Для освоения дисциплины используются следующие ресурсы:

- электронная информационно-образовательная среда НГУ (ЭИОС);
- образовательные интернет-порталы;
- информационно-телекоммуникационная сеть Интернет.

7.2 Современные профессиональные базы данных:

- <https://arxiv.org>;
- <https://github.com>

7.3 Информационные справочные системы:

- <https://docs.python.org>;
- <https://stackoverflow.com>

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

8.1 Перечень программного обеспечения

Для обеспечения реализации дисциплины используется стандартный комплект программного обеспечения (ПО), включающий регулярно обновляемое лицензионное ПО Windows или Ubuntu, Python

8.2 Информационные справочные системы

- <https://docs.python.org>;
- <https://stackoverflow.com>

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, занятий семинарского типа, курсового проектирования (выполнения курсовых работ), групповых и индивидуальных консультаций, текущего контроля, промежуточной и итоговой аттестации;
2. Помещения для самостоятельной работы обучающихся;
3. Помещения для практических занятий с компьютерным оборудованием
4. Помещения для хранения и профилактического обслуживания учебного оборудования.

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

Помещения для практической работы должны быть оборудованными компьютерной техникой с ОС Windows или Linux с установленным Python3.

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

Для проведения занятий лекционного типа предлагаются следующие наборы демонстрационного оборудования и учебно-наглядных пособий:

- комплект лекций-презентаций по темам дисциплины;
- комплект jupyter notebooks с визуализацией материалов.

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

Перечень результатов обучения по дисциплине их достижения представлен в разделе 1.

10.1 Порядок проведения текущего контроля и промежуточной аттестации по дисциплине

Текущий контроль успеваемости

В ходе реализации дисциплины используются следующие формы текущего контроля успеваемости обучающихся:

- выполнение практических работ, их контроль и оценивание осуществляется по завершении каждой работы.

Промежуточная аттестация

В рамках промежуточной аттестации учитывается текущий контроль успеваемости студента.

Окончательная оценка работы студента в течение семестра происходит на дифференцированном зачёте. Он проводится в конце семестра. На нем обучающемуся предлагается продемонстрировать полученные навыки в ходе решения одной задачи по разработке и применению программного обеспечения в физических исследованиях. Задачи дифференцированного зачета носят комплексный характер, т.к. включает вопросы ситуационно-производственного, практического, а также научно-исследовательского содержания. Во время проведения зачета студенту разрешается использовать десктопы, ноутбуки и программное обеспечение удаленных серверов. В процессе проверки задач студенту могут быть заданы дополнительные устные вопросы по темам дисциплины.

Результаты промежуточной аттестации определяются оценками «отлично», «хорошо», «удовлетворительно», «неудовлетворительно». Оценки «отлично», «хорошо», «удовлетворительно» означают успешное прохождение промежуточной аттестации.

Описание критериев и шкал оценивания индикаторов достижения результатов обучения по дисциплине Физика

| Критерии оценивания результатов обучения | Шкала оценивания |
|--|--------------------------|
| <p><u>Выполнение заданий:</u> Студентом выполнены все работы, включенные в состав изучаемой дисциплины, при этом он;</p> <ul style="list-style-type: none"> • свободно применяет полученные знания при выполнении индивидуального практического задания; • выполнил работу в полном объеме с соблюдением необходимой последовательности действий; • при ответах на контрольные вопросы правильно понимает их сущность, дает точное определение и истолкование основных понятий, использует специальную терминологию дисциплины. <p><u>Дифференцированный зачет:</u> Студент продемонстрировал, полученные при изучении дисциплины навыки, в ходе решения задачи. Задача решена правильно и самостоятельно, была проведена математическая обработка полученных результатов.</p> | <i>Отлично</i> |
| <p><u>Выполнение заданий:</u> Студентом выполнены требования к оценке «отлично», но допущены 2–3 недочета при выполнении практического задания. Выполнил все работы, включенные в состав портфолио, при этом;</p> <ul style="list-style-type: none"> • недочеты, допущенные при выполнении практического задания, студент может исправить самостоятельно или при небольшой помощи преподавателя; • при ответах на контрольные вопросы не допускает серьезных ошибок, легко устраняет отдельные неточности, но затрудняется в применении знаний в новой ситуации. <p><u>Дифференцированный зачет:</u> Задача решена самостоятельно, но допускались не принципиальные неточности. В ответе задачи имеют место несущественные ошибки, которые студент способен исправить самостоятельно или благодаря наводящему вопросу.</p> | <i>Хорошо</i> |
| <p><u>Выполнение заданий:</u> Практическое задание выполнено полностью, но объем выполненной части позволяет получить правильные результаты и выводы, при этом</p> <ul style="list-style-type: none"> • в ходе выполнения работы студент продемонстрировал слабые практические навыки, были допущены ошибки; • студент умеет применять полученные знания при решении простых задач по готовому алгоритму; • при ответах на контрольные вопросы правильно понимает их сущность, но в ответе имеются отдельные пробелы и при самостоятельном воспроизведении материала требует дополнительных и уточняющих вопросов преподавателя. <p><u>Дифференцированный зачет:</u> Студент испытывал трудности и не полностью продемонстрировал навыки, полученные при изучении дисциплины, в ходе решения задачи. Задача решена не полностью, при решении были допущены ошибки, которые были исправлены с помощью преподавателя.</p> | <i>Удовлетворительно</i> |

| | |
|--|-----------------------------------|
| <p><u>Выполнение заданий:</u> Практическое задание выполнено не полностью или не выполнено совсем, при этом</p> <ul style="list-style-type: none"> • объем выполненной работы не позволяет сделать правильные выводы, у студента имеются лишь отдельные представления об изученном материале, большая часть материала не усвоена; • на контрольные вопросы студент не может дать ответов, так как не овладел основными знаниями и умениями в соответствии с требованиями программы. <p><u>Дифференцированный зачет:</u> Студент не решил задачу или задача решена не полностью, при решении были допущены ошибки, которые студент не может исправить даже с помощью преподавателя.</p> | <p><i>Неудовлетворительно</i></p> |
|--|-----------------------------------|

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

Темы практических работ

1. Установка Linux на RAID1. Файловые системы в Linux, включая их виды и механизмы работы. Методы монтирования и техники управления дисками.
2. Практикум по основным инструментам и утилитам в Linux. Основы работы с командной строкой, траблшутинг, дебаг, перформанс-тюнинг и быстрая аналитика с помощью базовых инструментов Linux. Планировщик процессов и подсистема виртуальной памяти. Cgroup и Namespace.
3. Разработка прикладного ПО на языке Python. Синтаксис языка с высоты птичьего полёта. Интерактивная оболочка IPython. Системы контроля версий и контейнеризации программного обеспечения.
4. Проектирование жизненного цикла ПО. Основные этапы жизни программного обеспечения и как они выглядят в проектах ИЯФ СО РАН.
5. Особенности работы с СУБД. Реляционные и нереляционные базы данных. Основы языка SQL. Построение простого dashboard.
6. Обработка и анализ экспериментальных и расчетных данных физического эксперимента. Знакомство с современными инструментами и фреймворками для работы с данными на языке Python.
7. Основы машинного обучения и нейронных сетей.
8. Инструменты для профилирования и ускорения программ с помощью модулей timeit, cProfile и line_profiler. Немного о NumPy. JIT и AOT компиляция кода на Python на примере Numba и Cython.
9. Параллелизм и конкурентность. Модули threading, queue и concurrent.futures. Использование потоков для параллельных вычислений на Python. GIL. Модули asyncio и multiprocessing.
10. Основы программирования на графических ускорителях. CUDA.
11. Работа с примерами ПО для моделирования физических явлений

Примерные задания к дифференцированному зачету

1. Написать ПО для решения заданного дифференциального уравнения параболического типа.
2. Написать ПО для решения классической задачи коммивояжера с помощью эвристических методов.
3. Найти в исходном коде заданного ПО самую нагруженную функцию.
4. Ускорить выполнение заданного ПО с помощью параллелизма на CPU.
5. Привести git-репозиторий заданного ПО в порядок, исправить ошибки и запустить, подготовить ПО к деплою в производственную среду.
6. Минимально инвазивными изменениями в СУБД добиться того, чтобы ПО работало с приемлемой скоростью.
7. Научить классификатор определять тип частицы по заданному массиву данных.

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

Аннотация
к рабочей программе дисциплины
«Применение Python в физических исследованиях»
Направление: 03.03.01 Прикладные математика и физика

Программа дисциплины «Применение Python в физических исследованиях» составлена в соответствии с требованиями ФГОС ВО по направлению подготовки 03.03.01 Прикладная математика и 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. Дисциплина изучается студентами третьего курса физического факультета.

Цель дисциплины – обучение студентов физических специальностей разрабатывать и применять программное обеспечение в своих научных работах. Данный курс позволит не только освоить базовые и продвинутые аспекты разработки программного обеспечения, но и научит использовать его для решения конкретных физических задач. Программа курса начинается с основ операционной системы GNU/Linux, синтаксиса языка Python и методологий разработки кода, а затем плавно переходит к методам численного анализа, обработке и визуализации данных, машинному обучению и нейронным сетям. Особое внимание уделено оптимизации кода и использованию высокопроизводительных вычислений.

| Результаты освоения образовательной программы (компетенции) | Индикаторы | Результаты обучения по дисциплине |
|--|---|--|
| ОПК-6. Способен разрабатывать алгоритмы и компьютерные программы, пригодные для практического применения | ОПК-6.1 Применяет алгоритмические языки программирования, операционные системы и оболочки, современные среды разработки программного обеспечения ОПК-6.2 Составляет алгоритмы, пишет и отлаживает коды на языке программирования, тестирует работоспособность программы, интегрирует программные модули. | Знать основы операционной системы GNU/Linux, основы языка программирования Python, основы методологий разработки DevOps и SRE, основные инструменты на Python для обработки и анализа данных, основы машинного обучения и нейронных сетей, различные системы управления базами данных (СУБД). Уметь использовать интерактивную оболочку IPython и применять язык Python для решения физических задач, автоматизировать процессы разработки и обеспечивать надежность систем, применять СУБД, измерять время выполнения кода, использовать параллельные вычисления и работать с GPU для ускорения обработки данных и вычислений. |

Программой дисциплины предусмотрены следующие виды контроля:

Текущий контроль успеваемости: выполнение заданий на практических работах.

Промежуточная аттестация: дифференцированный зачёт.

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

**Лист актуализации рабочей программы дисциплины
«Применение Python в физических исследованиях»
Направление: 03.03.01 Прикладные математика и физика**

[illegible]