

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

**УТВЕРЖДЕНО**  
**Проректор по учебной работе**

**А.А. Воронов**

	<b>Рабочая программа дисциплины (модуля)</b>
<b>по дисциплине:</b>	Язык Python
<b>по направлению:</b>	Техническая физика
<b>профиль подготовки:</b>	Техническая физика космических летательных аппаратов Физтех-школа Аэрокосмических Технологий кафедра информатики и вычислительной математики
<b>курс:</b>	2
<b>квалификация:</b>	бакалавр

Семестр, формы промежуточной аттестации: 3 (осенний) - Дифференцированный зачет

Аудиторных часов: 60 всего, в том числе:

лекции: 0 час.

семинары: 0 час.

лабораторные занятия: 60 час.

Самостоятельная работа: 120 час.

Всего часов: 180, всего зач. ед.: 4

Количество контрольных работ, заданий: 4

Программу составил: К.А. Беклемышева, канд. физ.-мат. наук, доцент

Программа обсуждена на заседании кафедры информатики и вычислительной математики 27.04.2022

## Аннотация

Семестровый курс, посвящённый языку программирования Python и введению в многопоточность. Язык Python изучается как второй, изучаемый после C++ - предполагается, что слушатели курса уже знакомы с программированием в целом, основными алгоритмами и структурами данных, ООП. В силу этого дополнительно рассматривается написание собственных модулей на C++ для последующего использования из Python. Из фундаментальных тем в рамках курса рассматриваются введение в многопоточность и сетевое взаимодействие.

### 1. Цели и задачи

#### Цель дисциплины

- Формирование представлений о языках высокого уровня (на примере Python) и их месте в прикладной разработке программ;
- формирование представлений о механизмах многопоточности и параллельного исполнения в современных операционных системах.

#### Задачи дисциплины

- Изучение языка Python, его сопоставление с языком C++, формирование представлений о сильных и слабых сторонах языков для различных задач;
- изучение взаимодействия программ на языках высокого уровня с бинарными модулями;
- изучение фундаментальных механизмов обеспечения многопоточности в современных операционных системах и их использования из языков высокого уровня.

### 2. Перечень формируемых компетенций

Освоение дисциплины направлено на формирование следующих компетенций:

Код и наименование компетенции	Индикаторы достижения компетенции
ОПК-2 Способен применять методы математического анализа, математического моделирования и оптимизации для решения задач, возникающих в ходе профессиональной деятельности	ОПК-2.2 Способен строить математические модели, производить количественные расчеты и оценки
ПК-1 Способен применять эффективные методы исследования физико-технических объектов и процессов, проводить испытания технологических процессов и (или) изделий с использованием современных аналитических средств технической физики	ПК-1.2 Владеет аналитическими, вычислительными и экспериментальными методами исследования

### 3. Перечень планируемых результатов обучения по дисциплине (модулю)

В результате освоения дисциплины обучающиеся должны

знать:

- Конструкции языка программирования Python;
- общие принципы взаимодействия программ на языках высокого уровня с бинарными модулями;
- фундаментальные механизмы обеспечения многопоточности в современных операционных системах;
- основные понятия и принципы сетевого взаимодействия программ.

уметь:

- Разрабатывать программы на языке Python;
- использовать внешние модули при разработке на языке Python;
- использовать современные средства написания и отладки программ;
- использовать механизмы многопоточности и сетевого взаимодействия при разработке программ.

владеть:

- Языком программирования Python;
- современными средствами написания и отладки программ;
- методами создания программ с использованием модулей.

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

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

№	Тема (раздел) дисциплины	Трудоемкость по видам учебных занятий, включая самостоятельную работу, час.			
		Лекции	Семинары	Лаборат. работы	Самост. работа
1	Язык Python (как второй, после C++)			12	24
2	Окружение Python			8	16
3	Некоторые библиотеки Python			8	16
4	Базовые механизмы многопоточности (изложение на C / C++)			12	24
5	Понятие о продвинутых механизмах многопоточности			4	8
6	Многопоточность в Python			4	8
7	Введение в сетевое взаимодействие (на C++ / Python)			12	24
Итого часов				60	120
Подготовка к экзамену		0 час.			
Общая трудоёмкость		180 час., 4 зач.ед.			

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

Семестр: 3 (Осенний)

###### 1. Язык Python (как второй, после C++)

- особенности системы типов, duck typing
- базовые конструкции языка
- типовые контейнеры и алгоритмы (по аналогии с C++ STL)
- конструкции, специфичные для языка Python
- ООП в Python

###### 2. Окружение Python

- базовое окружение Python (интерпретатор, модули, env-ы)
- вызов бинарных модулей из программы на Python
- написание собственных модулей на C++ для последующего использования из Python

###### 3. Некоторые библиотеки Python

- matplotlib для визуализации данных
- NumPy для вычислений

###### 4. Базовые механизмы многопоточности (изложение на C / C++)

- процессы (fork, exec, PID)

- треды (create, join, TID)
- понятие race condition и critical section
- механизмы IPC и примитивы синхронизации (pipe, fifo, семафоры, мьютексы, shared memory, message queue, сигналы)

#### 5. Понятие о продвинутых механизмах многопоточности

- понятие о продвинутых механизмах многопоточности (корутины, async IO)

#### 6. Многопоточность в Python

- устройство многопоточных программ в Python (threading, multiprocessing, GIL)

#### 7. Введение в сетевое взаимодействие (на C++ / Python)

- сокеты, порты, адреса, базовое подключение как поток байтов
- понятие о протоколах транспортного уровня (на примере TCP / UDP)
- протоколы прикладного уровня (на примере HTTP и requests)
- понятие о RPC

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

Учебная аудитория, оснащенная мультимедиапроектором и экраном для чтения лекций.  
Учебный сетевой компьютерный класс с установленной операционной системой Linux, подключенный к сети Интернет.

### 6. Перечень рекомендуемой литературы

#### Основная литература

1. Основы операционных систем [Текст] : курс лекций : учеб. пособие для вузов / В. Е. Карпов, К. А. Коньков ; под ред. В. П. Иванникова .— М. : ИНТУИТ. РУ : Интернет-Университет Информационных Технологий, 2004 .— 628 с.
- Лутц, М. Python: карманный справочник / М. Лутц ; пер. с англ. И. В. Берштейна. — 5-е изд. — М : Вильямс, 2015. — 320 с. — ISBN 978-5-8459-1965-6.

#### Дополнительная литература

1. Операционная система UNIX [Текст] : учеб. пособие для вузов / А. М. Робачевский, С. А. Немнюгин, О. Л. Стесик .— 2-е изд., перераб. и доп. — СПб. : БХВ-Петербург, 2005, 2007, 2010 .— 656 с.
- Саммерфилд, М. Python на практике: создание качественных программ с использованием параллелизма, библиотек и паттернов / М. Саммерфилд; пер. с англ. А. А. Слинкина. — М : ДМК Пресс, 2014. — 338 с. — ISBN 978-5-97060-095-5.

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

Не используются

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

На лекциях и лабораторных занятиях используются мультимедийные технологии, включая демонстрацию презентаций. На компьютерах в компьютерных классах должна быть установлены операционная система Linux. Студенты могут использовать среду разработки PyCharm. При её отсутствии используется интерпретатор python в командной строке.

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

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

Успешное освоение курса требует напряжённой самостоятельной работы студента. В программе курса приведено минимально необходимое время для работы студента над темой. Самостоятельная работа включает в себя:

- чтение и конспектирование рекомендованной литературы;
- проработку учебного материала (по конспектам лекций, учебной и научной литературе), подготовку ответов на вопросы, предназначенных для самостоятельного изучения, доказательство отдельных утверждений, свойств;
- решение задач, предлагаемых студентам на лекциях и лабораторных занятиях;
- подготовку к лабораторным занятиям, дифференцированному зачёту.

Руководство и контроль за самостоятельной работой студента осуществляется в форме индивидуальных консультаций.

Показателем владения материалом служит умение решать задачи. Для формирования умения применять теоретические знания на практике студенту необходимо решать как можно больше задач. При решении задач каждый шаг решения необходимо аргументировать, ссылаясь на известные теоретические сведения.

При подготовке к лабораторным занятиям необходимо повторять ранее изученные основные понятия. В начале занятия, как правило, проводится короткий (10-15 минут) опрос по материалу прошедших занятий в устной или письменной форме. Обычно придерживаются следующей схемы: изучение материала лекции по конспекту в тот же день, когда была прослушана лекция (10-15 минут); повторение материала накануне следующей лекции (10-15 минут), проработка учебного материала по конспектам лекций, учебной и научной литературе, подготовка ответов на вопросы, предназначенных для самостоятельного изучения (1 час неделю), подготовка к практическому занятию, решение задач (1 час). Важно добиться понимания изучаемого материала, а не механического его запоминания. При затруднении изучения отдельных тем, вопросов, следует обращаться за консультациями к лектору или преподавателю, ведущему лабораторные занятия.

Обязательным требованием является выполнение домашних заданий, которые в отдельных случаях могут быть оформлены как контексты с автоматическим тестированием решений. Проверка текста решения преподавателем является обязательным элементом контроля, вне зависимости от наличия автоматического тестирования.

Промежуточный контроль знаний проводится в виде контрольных работ, на которых студенту предлагается решить несколько задач. Также студенту в ходе освоения курса необходимо выполнить две домашние индивидуальные работы с их последующей защитой.

**ОЦЕНОЧНЫЕ МАТЕРИАЛЫ ПО ДИСЦИПЛИНЕ (МОДУЛЮ)**

**по направлению:** Техническая физика  
**профиль подготовки:** Техническая физика космических летательных аппаратов  
Физтех-школа Аэрокосмических Технологий  
кафедра информатики и вычислительной математики  
**курс:** 2  
**квалификация:** бакалавр

Семестр, формы промежуточной аттестации: 3 (осенний) - Дифференцированный зачет

**Разработчик:** К.А. Беклемышева, канд. физ.-мат. наук, доцент

## 1. Компетенции, формируемые в процессе изучения дисциплины

Код и наименование компетенции	Индикаторы достижения компетенции
ОПК-2 Способен применять методы математического анализа, математического моделирования и оптимизации для решения задач, возникающих в ходе профессиональной деятельности	ОПК-2.2 Способен строить математические модели, производить количественные расчеты и оценки
ПК-1 Способен применять эффективные методы исследования физико-технических объектов и процессов, проводить испытания технологических процессов и (или) изделий с использованием современных аналитических средств технической физики	ПК-1.2 Владеет аналитическими, вычислительными и экспериментальными методами исследования

## 2. Показатели оценивания компетенций

В результате изучения дисциплины «Язык Python» обучающийся должен:

### знать:

- Конструкции языка программирования Python;
- общие принципы взаимодействия программ на языках высокого уровня с бинарными модулями;
- фундаментальные механизмы обеспечения многопоточности в современных операционных системах;
- основные понятия и принципы сетевого взаимодействия программ.

### уметь:

- Разрабатывать программы на языке Python;
- использовать внешние модули при разработке на языке Python;
- использовать современные средства написания и отладки программ;
- использовать механизмы многопоточности и сетевого взаимодействия при разработке программ.

### владеть:

- Языком программирования Python;
- современными средствами написания и отладки программ;
- методами создания программ с использованием модулей.

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

Текущий контроль усвоения материала студентами осуществляется на основании выполнения домашних и контрольных работ.

Примеры задач:

1. Считайте со стандартного ввода набор записей, содержащих множество полей. Выполните фильтрацию, сортировку, группировку по заданным правилам. Выведите результат на стандартный поток вывода.
2. Реализуйте заданные геометрические примитивы в 2D и/или 3D. Оформите как модуль, содержащий набор объектов.
3. Импортируйте модуль, содержащий геометрические примитивы. Используя их, реализуйте расчёт взаимного движения и столкновения объектов.
4. Считайте файл с данными в формате csv. Выполните фильтрацию, сортировку, группировку по заданным правилам. Результат визуализируйте с использованием matplotlib.
5. Реализуйте многопоточное приложение, обрабатывающее директорию с файлами. Для обработки каждого файла должен запускаться отдельный поток.
6. Реализуйте пару приложений, двусторонне взаимодействующих через pipe, используя семафор для синхронизации.

7. Напишите простую командную оболочку, содержащую набор встроенных функций (builtins) и дающую возможность запускать сторонние программы.
8. Напишите многопоточный сетевой сервер и клиентов к нему, реализующих консольный текстовый чат.

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

Примеры вопросов дифференцированного зачета:

1. Особенности системы типов Python, duck typing.
2. Конструкции языка Python. Конструкции, общие с C++. Конструкции, специфичные для языка Python, и их применение.
3. ООП в Python. Сходства и различия с C++.
4. Типовые контейнеры и алгоритмы Python. Их сходства и отличия от C++ STL.
5. Базовое окружение Python (интерпретатор, модули, env-ы).
6. Вызов бинарных модулей из программы на Python.
7. Библиотека matplotlib для визуализации данных.
8. Библиотека NumPy для вычислений.
9. Базовые механизмы многопоточности: процессы (fork, exec, PID).
10. Базовые механизмы многопоточности: треды (create, join, TID).
11. Базовые механизмы многопоточности: понятие race condition и critical section.
12. Механизмы IPC и примитивы синхронизации: pipe, fifo.
13. Механизмы IPC и примитивы синхронизации: семафоры, мьютексы.
14. Механизмы IPC и примитивы синхронизации: сигналы.
15. Многопоточность в Python: threading, multiprocessing, GIL.
16. Базовые понятия сетевого взаимодействия: сокеты, порты, адреса.
17. Протоколы транспортного уровня (на примере TCP / UDP).
18. Протоколы прикладного уровня (на примере HTTP и requests).
19. Механизм RPC. Его примеры в Python или C++.

#### **Критерии оценивания**

- 10 - выставляется студенту, показавшему всесторонние, систематизированные, глубокие знания учебной программы дисциплины, проявляющему интерес к данной предметной области, продемонстрировавшему умение уверенно и творчески применять их на практике при решении конкретных задач, свободное и правильное обоснование принятых решений.
- 9 - выставляется студенту, показавшему всесторонние, систематизированные, глубокие знания учебной программы дисциплины и умение уверенно применять их на практике при решении конкретных задач, свободное и правильное обоснование принятых решений.
- 8 - выставляется студенту, показавшему всесторонние, систематизированные, глубокие знания учебной программы дисциплины и умение уверенно применять их на практике при решении конкретных задач, правильное обоснование принятых решений, с некоторыми недочетами.
- 7 - выставляется студенту, если он твердо знает материал, грамотно и по существу излагает его, умеет применять полученные знания на практике, но недостаточно грамотно обосновывает полученные результаты.
- 6 - выставляется студенту, если он твердо знает материал, грамотно и по существу излагает его, умеет применять полученные знания на практике, но допускает в ответе или в решении задач некоторые неточности.
- 5 - выставляется студенту, если он в основном знает материал, грамотно и по существу излагает его, умеет применять полученные знания на практике, но допускает в ответе или в решении задач достаточно большое количество неточностей.
- 4 - выставляется студенту, показавшему фрагментарный, разрозненный характер знаний, недостаточно правильные формулировки базовых понятий, нарушения логической последовательности в изложении программного материала, но при этом он освоил основные разделы учебной программы, необходимые для дальнейшего обучения, и может применять полученные знания по образцу в стандартной ситуации.



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

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

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

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

Аттестация по дисциплине осуществляется в форме дифференцированного зачета. При успешном устном ответе студента на вопрос дифференцированного зачета, оценка за него выставляется с учётом работы студента в течение семестра на основе оценок за контрольные работы и домашние задания.