

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

УТВЕРЖДЕНО

**Директор физтех-школы
прикладной математики и
информатики**

А.М. Райгородский

	Рабочая программа дисциплины (модуля)
по дисциплине:	Практикум по программированию на языке Python
по направлению:	Информатика и вычислительная техника
профиль подготовки:	Математическое моделирование и компьютерные технологии Физтех-школа Прикладной Математики и Информатики кафедра интеллектуальных систем
курс:	3
квалификация:	бакалавр

Семестр, формы промежуточной аттестации: 6 (весенний) - Зачет

Аудиторных часов: 30 всего, в том числе:

лекции: 0 час.

семинары: 30 час.

лабораторные занятия: 0 час.

Самостоятельная работа: 15 час.

Всего часов: 45, всего зач. ед.: 1

Количество контрольных работ, заданий: 1

Программу составил: А.А. Адуенко, канд. физ.-мат. наук

Программа обсуждена на заседании кафедры интеллектуальных систем 05.05.2023

Аннотация

Язык программирования Python — это мощный инструмент для создания программ самого разнообразного назначения. С его помощью можно решать задачи различных типов.

Курс призван научить программировать на Python, освоить основные объекты и процедуры языка Python, методы обработки ошибок в языке Python.

1. Цели и задачи

Цель дисциплины

- обучение основам программирования на языке Python.

Задачи дисциплины

- познакомить студентов с основными конструкциями, объектами и процедурами языка Python;
- сформировать навыки написания эффективного, простого, понятного и гибкого кода, оптимального с точки зрения повышения скорости и качества разработки;
- научить эффективному управлению памятью, методам обработки ошибок и тестирования кода на языке Python.

2. Перечень формируемых компетенций

Освоение дисциплины направлено на формирование следующих компетенций:

Код и наименование компетенции	Индикаторы достижения компетенции
ПК-1 Способен ставить, формализовывать и решать задачи, в том числе разрабатывать и исследовать математические модели изучаемых явлений и процессов, системно анализировать научные проблемы, получать новые научные результаты	ПК-1.1 Способен находить, анализировать и обобщать информацию об актуальных результатах исследований в рамках тематической области своей профессиональной деятельности
	ПК-1.2 Способен выдвигать гипотезы, строить математические модели для описания изучаемых явлений и процессов, оценивать качество разработанной модели
	ПК-1.3 Способен применять теоретические и (или) экспериментальные методы исследований к конкретной научной задаче и интерпретировать полученные результаты
ПК-2 Способен самостоятельно или в качестве члена (руководителя) малого коллектива организовывать и проводить научные исследования и их апробацию	ПК-2.2 Способен планировать и проводить научные исследования самостоятельно или в качестве члена (руководителя) малого научного коллектива

3. Перечень планируемых результатов обучения по дисциплине (модулю)

В результате освоения дисциплины обучающиеся должны

знать:

- парадигму объектно-ориентированного программирования. Основные объекты и процедуры языка Python. Методы обработки ошибок в языке Python.

уметь:

- писать эффективный код, отлаживать и документировать код на языке Python.

владеть:

- средствами разработки и тестирования программного кода на языке Python, объектами и средствами, предлагаемыми стандартными библиотеками. Технологиями многопоточного и распределенного программирования.

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	Модель памяти. Функциональное программирование.		5		2
2	Обзор библиотек.		5		3
3	Объектно-ориентированное программирование. Обработка ошибок.		5		2
4	Основы языка.		5		3
5	Оформление и тестирование кода. Работа со строками.		5		2
6	Параллельные вычисления в Python. Расширенная работа с объектами.		5		3
Итого часов			30		15
Подготовка к экзамену		0 час.			
Общая трудоёмкость		45 час., 1 зач.ед.			

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

Семестр: 6 (Весенний)

1. Модель памяти. Функциональное программирование.

Хранение объектов в памяти, сборщик мусора. Хранение объектов по ссылке и по значению. Изменяемые и неизменяемые объекты. Модуль `sorted`. Обработка списков, функция `map` и др., лямбда-функции, распаковка списков и словарей. Расширенная обработка аргументов функций. Генераторы и "ленивое" исполнение. Управляющие исключения. Модуль `itertools`.

2. Обзор библиотек.

Библиотеки для обработки аргументов командной строки. Системные библиотеки. Стандартные математические библиотеки. Регулярные выражения и модуль `re`. Библиотеки для работы с HTML/XML. Математические библиотеки: `SciPy` и др. Библиотека `Tkinter`.

3. Объектно-ориентированное программирование. Обработка ошибок.

Классы, объекты. Пользовательские классы, методы и члены. Конструктор класса. Перегрузка операторов. Объекты в Python. Исключения, их генерация и обработка. Пользовательские исключения. Освобождение ресурсов, менеджеры контекстов.

4. Основы языка.

Интерпретатор и его интерактивный режим. Динамическая типизация, базовые типы данных: числовые, `str`, `list`. Основные операторы, оператор `print`. Блоки кода, основные составные операторы: `if`, `while`, `for`. Основные встроенные функции. Создание пользовательских функций. Выражения, приоритеты операторов. Работа с файлами. Тип `dict`, хэширование. Модули, оператор `import`, модуль `sys`.

5. Оформление и тестирование кода. Работа со строками.

Документирование кода. Инструмент `pydoc`. Юнит-тестирование. Модуль `unittest`. Инструменты для тестирования. Инструменты `pylint`, `pyflakes`. Отладочные инструменты. Модули, создание модулей. Пространства имен. Исполнение модулей как скриптов. Встроенные функции строк. Форматирование строк. Модуль `string`. Класс `unicode`, его функции. Кодировки и `Unicode`, кодирование файлов и исходного кода.

6. Параллельные вычисления в Python. Расширенная работа с объектами.

Многопоточные программы и GIL. Многопроцессорные программы. Модификаторы доступа. Наследование, разрешение имен. Метаклассы. Объект `type`. Декораторы.

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

Учебная аудитория, оснащенная компьютером и мультимедийным оборудованием (проектор, звуковая система).

6. Перечень рекомендуемой литературы

Основная литература

1. Марк Лутц. Программирование на Python. (Подробное руководство по языку).
2. Марк Пилгрим. Погружение в Python. (Быстрое введение для хорошо знакомых с другими языками).
3. Официальная документация.

Дополнительная литература

1. Официальный сайт языка
2. Дзен питона
3. Стайлгайд, принятый в сообществе Python (PEP 8)
4. Список операторов, которые можно переопределять.
5. Сайт PyPI и документация по установке пакетов

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

<http://elibrary.ru/defaultx.asp> Научная электронная библиотека;
<http://www.twirpx.com> Все для студента

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

На занятиях используются мультимедийные технологии, включая демонстрацию презентаций.

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

Методические рекомендации позволяют студенту оптимальным образом организовать процесс обучения. В рабочей программе приведено примерное распределение часов аудиторной и внеаудиторной нагрузки по различным темам данной дисциплины.

ОЦЕНОЧНЫЕ МАТЕРИАЛЫ ПО ДИСЦИПЛИНЕ (МОДУЛЮ)

по направлению:	Информатика и вычислительная техника
профиль подготовки:	Математическое моделирование и компьютерные технологии Физтех-школа Прикладной Математики и Информатики кафедра интеллектуальных систем
курс:	3
квалификация:	бакалавр

Семестр, формы промежуточной аттестации: 6 (весенний) - Зачет

Разработчик: А.А. Адуенко, канд. физ.-мат. наук

1. Компетенции, формируемые в процессе изучения дисциплины

Код и наименование компетенции	Индикаторы достижения компетенции
ПК-1 Способен ставить, формализовывать и решать задачи, в том числе разрабатывать и исследовать математические модели изучаемых явлений и процессов, системно анализировать научные проблемы, получать новые научные результаты	ПК-1.1 Способен находить, анализировать и обобщать информацию об актуальных результатах исследований в рамках тематической области своей профессиональной деятельности
	ПК-1.2 Способен выдвигать гипотезы, строить математические модели для описания изучаемых явлений и процессов, оценивать качество разработанной модели
	ПК-1.3 Способен применять теоретические и (или) экспериментальные методы исследований к конкретной научной задаче и интерпретировать полученные результаты
ПК-2 Способен самостоятельно или в качестве члена (руководителя) малого коллектива организовывать и проводить научные исследования и их апробацию	ПК-2.2 Способен планировать и проводить научные исследования самостоятельно или в качестве члена (руководителя) малого научного коллектива

2. Показатели оценивания компетенций

В результате изучения дисциплины «Практикум по программированию на языке Python» обучающийся должен:

знать:

- парадигму объектно-ориентированного программирования. Основные объекты и процедуры языка Python. Методы обработки ошибок в языке Python.

уметь:

- писать эффективный код, отлаживать и документировать код на языке Python.

владеть:

- средствами разработки и тестирования программного кода на языке Python, объектами и средствами, предлагаемыми стандартными библиотеками. Технологиями многопоточного и распределенного программирования.

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

Примеры контрольных заданий:

1. Создание пользовательских функций. Выражения, приоритеты операторов.
2. Работа с файлами. Тип dict, хэширование.
3. Модули, оператор import, модуль sys.
4. Перегрузка операторов. Объекты в Python.
5. Исполнение модулей как скриптов.
6. Встроенные функции строк. Форматирование строк.
7. Модуль string. Класс unicode, его функции.
8. Кодировки и Unicode, кодирование файлов и исходного кода.
9. Расширенная обработка аргументов функций. Генераторы и "ленивое" исполнение.
10. Управляющие исключения. Модуль itertools.
11. Наследование, разрешение имен. Метаклассы. Объект type. Декораторы.

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

1. Основы языка. Интерпретатор и его интерактивный режим.

2. Динамическая типизация, базовые типы данных: числовые, str, list. Основные операторы, оператор print. Блоки кода, основные составные операторы: if, while, for. Основные встроенные функции.
3. Объектно-ориентированное программирование. Обработка ошибок. Классы, объекты. Пользовательские классы, методы и члены. Конструктор класса.
4. Исключения, их генерация и обработка. Пользовательские исключения. Освобождение ресурсов, менеджеры контекстов.
5. Оформление и тестирование кода. Работа со строками. Документирование кода. Инструмент pydoc. Юнит-тестирование. Модуль unittest.
6. Инструменты для тестирования. Инструменты pylint, pyflakes. Отладочные инструменты. Модули, создание модулей. Пространства имен.
7. Модель памяти. Функциональное программирование. Хранение объектов в памяти, сборщик мусора. Хранение объектов по ссылке и по значению.
8. Изменяемые и неизменяемые объекты. Модуль сорю. Обработка списков, функция map и др., лямбда-функции, распаковка списков и словарей.
9. Обзор библиотек. Библиотеки для обработки аргументов командной строки. Системные библиотеки. Стандартные математические библиотеки.
10. Регулярные выражения и модуль re. Библиотеки для работы с HTML/XML. Математические библиотеки: SciPy и др. Библиотека Tkinter.
11. Параллельные вычисления в Python. Расширенная работа с объектами.
12. Многопоточные программы и GIL. Многопроцессорные программы. Модификаторы доступа.

Критерии оценивания

Оценка «Зачтено» выставляется показавшему владение основными разделами программы.

Оценка «Не зачтено» выставляется студенту, у которого отсутствуют знания базовой составляющей дисциплины, допускаются грубые ошибки в изложении материала.

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

Зачет может проводиться по итогам текущей успеваемости и сдачи заданий и других видов работ, предусмотренных программой дисциплины и (или) путем организации специального опроса, проводимого в устной и (или) письменной форме.

При проведении зачета обучающемуся предоставляется 30 минут на подготовку. Опрос обучающегося по билету на зачете не должен превышать одного астрономического часа.