

**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)**

<b>Дисциплина:</b>	Программирование на Python
<b>Направление:</b>	Прикладные математика и физика
<b>Профиль подготовки:</b>	Математическая физика, компьютерные технологии и математическое моделирование в экономике Физтех-школа прикладной математики и информатики Кафедра проблем передачи информации и анализа данных
<b>Курс:</b>	3
<b>Квалификация:</b>	Бакалавр

Семестры, формы промежуточной аттестации: 6 (Весенний) – Дифференцированный зачет

Аудиторных часов: 30 всего, в том числе:

лекции: 30 час.

семинары: 0 час.

лабораторные занятия: 0 час.

Самостоятельная работа: 15 час.

Всего часов: 45, всего зач. ед.: 1

**Программу составил:** **Д.Ю. Дербышев**

**Аннотация**

Содержание курса фокусируется на современных технологиях, используемых при написании программ в больших проектах. Курс охватывает все шаги написания программного обеспечения, включая предварительную настройку среды программирования, способы упрощения и оптимизации кода, подходы к тестированию и валидации, а также составление отчетов.

В процессе работы также фокусируется внимание на способы реализации функциональности, выбранные в стандартных библиотеках Python, и проводится анализ возможных альтернативных решений и сравнение их с выбранным решением.

Оценка выставляется согласно качеству проектной работы, выполненной за семестр, с возможностью повышения оценки во время зачета на последнем занятии.

---

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

### Семестр 6 (весенний)

#### 1. Введение.

Более продвинутое описание стандартных средств разработки, чем уже было. Git branches, pull request, format-patch, т. д. PEP, автоформатирование, black, black-jupiter. Git hooks.

#### 2. Синтаксис python.

Декораторы, итераторы, менеджеры контекста, исключения, ellipsis.

#### 3. Стандартные пакеты.

Некоторые особенности numpy, pandas, scipy. Matplotlib и его альтернативы. Symru. Работа с файловой системой, работа с изображениями.

#### 4. Тестирование и логирование кода.

Pytest, logging.

#### 5. Рефакторинг.

Признаки плохого кода, техники рефакторинга, примеры применения.

#### 6. CLI, создание пакета.

CLI, click, argparse, hydra, setuptools.

#### 7. Отчеты и документация

Docstrings, annotation, notebooks.

#### 8. Оптимизация.

Perf, flamegraph. Сложности замера времени работы, timeit. GIL. Asyncio. Программирование на C: C API, cython. JIT и numba. CUDA вычисления.

### **Перечень основной и дополнительной литературы, необходимой для освоения дисциплины (модуля)**

#### Основная литература

1. Чистый Python. Тонкости программирования для профи, Дэн Бейдер, 2018.
2. Простой Python. Современный стиль программирования, Билл Любанович,

2021.

3. Погружение в Паттерны Проектирования, Александр Швец, 2018.

#### Дополнительная литература

1. Hypermodern Python Tooling, Claudio Jojowicz, 2024.
2. Numerical Python: Scientific Computing and Data Science Applications with Numpy, SciPy and Matplotlib, Robert Johansson, 2019.
3. Python. Сборник упражнений, Стивенсон Б., 2021.

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

- Hypermodern Python - <https://cjolowicz.github.io/posts/>
- Python documentation - <https://docs.python.org/3/>
- Pytest documentation - <https://docs.pytest.org/en/7.1.x/contents.html>