

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

**УТВЕРЖДЕНО**

**Директор по цифровизации  
образования**

**Д.И. Гриц**

<b>по дисциплине:</b>	<b>Рабочая программа дисциплины (модуля)</b> Разработка финансовых решений и сервисов
<b>по направлению:</b>	Бизнес-информатика
<b>профиль подготовки:</b>	Финансовые технологии и аналитика центр дополнительного, дополнительного профессионального и онлайн-образования "Пуск" центр дополнительного, дополнительного профессионального и онлайн-образования "Пуск"
<b>курс:</b>	1
<b>квалификация:</b>	магистр

Семестры, формы промежуточной аттестации:

2 (весенний) - Дифференцированный зачет  
3 (осенний) - Экзамен

Аудиторных часов: 72 всего, в том числе:

лекции: 30 час.  
семинары: 42 час.  
лабораторные занятия: 0 час.

Самостоятельная работа: 123 час.

Подготовка к экзамену: 30 час.

Всего часов: 225, всего зач. ед.: 5

Программу составили:

Е.А. Савицкая, начальник отдела  
О.А. Культепина, методист

Программа обсуждена на заседании центра дополнительного, дополнительного профессионального и  
онлайн-образования "Пуск" 13.06.2022

## Аннотация

В рамках дисциплины «Разработка финансовых решений и сервисов» обучающиеся разберут инструменты более продвинутой разработки ПО для финансового сектора и углубят знания Python и освоят базовый Golang.

### 1. Цели и задачи

#### Цель дисциплины

- освоение продвинутых инструментов и технологий по разработке финансовых решений и сервисов.

#### Задачи дисциплины

- овладение навыком сбора требования у заказчиков и подготовки технических алгоритмов;
- изучение на Python масштабируемых систем, ориентированных на высокие нагрузки, большие объемы данных и высокую надежность;
- овладение навыкам управления программными проектами в группах и организации зрелых программных процессов;
- обучение языку GoLang для разработки ПО.

### 2. Перечень формируемых компетенций

Освоение дисциплины направлено на формирование следующих компетенций:

Код и наименование компетенции	Индикаторы достижения компетенции
ОПК-3 Способен принимать решения, осуществлять стратегическое планирование и прогнозирование в профессиональной деятельности с использованием современных методов и программного инструментария сбора, обработки и анализа данных, интеллектуального оборудования и систем искусственного интеллекта	ОПК-3.1 Владеет методами стратегического планирования и прогнозирования в профессиональной деятельности
	ОПК-3.2 Самостоятельно выбирает и обосновывает выбор современных методов и программного инструментария сбора, обработки и анализа данных
ПК-16 Способен готовить аналитические материалы для оценки мероприятий и выработки стратегических решений в области ИКТ	ПК-16.1 Эффективно применяет в ходе профессиональной деятельности методы и инструментарий анализа данных
	ПК-16.2 Знает методы подготовки аналитических материалов для оценки мероприятий и выработки стратегических решений в области ИКТ

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

В результате освоения дисциплины обучающиеся должны

знать:

- требования к ПО в финансовом секторе;
- моделирование систем;
- подходы к технологиям интеграции;
- архитектура информационных систем и хранение данных;
- интерпретаторы;
- основные понятия программной инженерии;
- примеры использования разработки финансового ПО.

уметь:

- анализировать стадии проекта и предлагать решения;
- обрабатывать ошибки микросервисных систем;
- готовить требования к проектированию больших программных систем.

владеть:

- навыками программирования Python для решения финансового ПО;
- навыками программирования Golang.

### 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	Принципы разработки	4	4		20
2	Продвинутый Python для финтеха	6	6		30
3	Интерпретаторы	2	2		16
4	Программная инженерия	6	12		20
5	Разработка Golang	8	10		27
6	Разработка ПО для финансового сектора	4	8		10
Итого часов		30	42		123
Подготовка к экзамену		30 час.			
Общая трудоёмкость		225 час., 5 зач.ед.			

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

Семестр: 2 (Весенний)

##### 1. Принципы разработки

Аналитическая стадия проекта. Концепция и границы продукта. Требования к ПО. Функциональные и нефункциональные требования. Моделирование систем. Документирование. Архитектура информационных систем и хранение данных. Взаимодействие информационных систем (REST, SOAP, GraphQL). Синхронное и асинхронное взаимодействие. Подходы и технологии интеграции. Инфраструктура (DevOps).

##### 2. Продвинутый Python для финтеха

Python и структуры данных в нем. Структура python проекта. Написание автотестов. Сетевое взаимодействие. WSGI. Server-side web. Форматы данных. Проектирование API. Валидация. СУБД. ACID. ORM. NoSQL. Redis. Работа с кэшами. GIL, потоки, процессы. WS, SSE, Longpolling. Асинхронная обработка IO. Asyncio.

##### 3. Интерпретаторы

Как работает интерпретатор. Микросервисы. Внутренности интерпретатора CPython. Принципы построения микросервисных систем, типичные проблемы и способы их решения. Работа с памятью. Основы масштабируемых систем.

Семестр: 3 (Осенний)

##### 4. Программная инженерия

Архитектура программного обеспечения. Дисциплина проектирования больших программных систем. Документирование и чтение архитектуры программной системы. Дизайна и шаблоны проектирования в разработке программного обеспечения. Управление программными проектами в группах. Организация и деятельность зрелых программных процессов. Программирование SAS для финансового анализа.

## 5. Разработка Golang

Язык GoLang (Go) – симбиоз C++ и Python. Синтаксические конструкции языка. Устройство слайсов, структур и интерфейсов. Обработка ошибок. Многопоточное программирование. Оркестрация и синхронизация доступа к переменным. Паттерны pipeline и context. Пакеты database/sql и jack/pgx. Протоколы Golang. Unit- и benchmark-тесты, тестирование http-сервера.

## 6. Разработка ПО для финансового сектора

Разбор кейсов с рынка.

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

Занятия по учебной дисциплине проводятся с использованием дистанционных образовательных технологий. Каждый обучающийся обеспечен доступом к образовательной платформе <https://netology.ru/>.

## 6.Перечень рекомендуемой литературы

### Основная литература

1. Программирование на Python 3 : Подробное руководство [Текст] = Programming in Python 3 : [учеб. пособие для вузов] / М. Саммерфилд; пер. с англ. А. Киселева .— СПб : Символ-Плюс, 2015 .— 608 с.

Рекомендуемая литература для самостоятельного изучения

- 1.Bitcoin: A Peer-to-Peer Electronic Cash System, December 2008. Satoshi Nakamoto
- 2.Blockchain Revolution: How the Technology Behind Bitcoin Is Changing Money, Business, and the World), 2017. Дон Тапскотт, Алекс Тапскотт.
- 3.Introducing Ethereum and Solidity Foundations of Cryptocurrency and Blockchain Programming for Beginners. Chris Dannen
4. Just Enough Bitcoin for Ethereum
5. Ethereum White Paper. A Next-Generation Smart Contract and Decentralized Application Platform.

### Дополнительная литература

1. Микросервисы. Паттерны разработки и рефакторинга, Электронная версия печатной публикации / К. Ричардсон. — Санкт-Петербург, Питер, 2020

Рекомендуемая литература для самостоятельного изучения

1. Mastering Bitcoin: Programming the Open Blockchain, 2017. Andreas M. Antonopoulos
2. Mastering Ethereum, 2018. Andreas M. Antonopoulos, Gavin Wood.
3. Блокчейн изнутри. Как устроен Биткоин, Ethereum и смарт-контракты, 2017. Василий Зубарев.
4. Blockchain for Business - An Introduction to Hyperledger Technologies, 2017. LinuxFoundationX
5. Blockchain for Finance with Python, 2017. Pawel Lachowicz.

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

Th Go Programming Language [go.dev](https://go.dev)

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

1. Образовательная платформа <https://netology.ru/>
2. Webinar.ru
3. GitHub

4. Zoom
5. Google Drive

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

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

В результате изучения дисциплины студент должен знать основные определения дисциплины, уметь применять полученные знания для решения различных задач.

Успешное освоение курса требует:

- посещения всех занятий, предусмотренных учебным планом по дисциплине;
- ведения конспекта занятий;
- напряжённой самостоятельной работы студента.

Самостоятельная работа включает в себя:

- чтение рекомендованной литературы;
- проработку учебного материала, подготовку ответов на вопросы, предназначенных для самостоятельного изучения;
- решение задач, предлагаемых студентам на занятиях;
- подготовку к выполнению заданий текущей и промежуточной аттестации.

Показателем владения материалом служит умение без конспекта отвечать на вопросы по темам дисциплины.

Важно добиться понимания изучаемого материала, а не механического его запоминания. При затруднении изучения отдельных тем, вопросов, следует обращаться за консультациями к преподавателю.

Возможен промежуточный контроль знаний студентов в виде решения задач в соответствии с тематикой занятий.

**ОЦЕНОЧНЫЕ МАТЕРИАЛЫ ПО ДИСЦИПЛИНЕ (МОДУЛЮ)**

<b>по направлению:</b>	Бизнес-информатика		
<b>профиль подготовки:</b>	Финансовые технологии и аналитика	▲	▲
	онлайн-образования "Пуск"	▲	▲
	онлайн-образования "Пуск"		
<b>курс:</b>	1		
<b>квалификация:</b>	магистр		

Семестры, формы промежуточной аттестации:

- 2 (весенний) - Дифференцированный зачет
- 3 (осенний) - Экзамен

**Разработчики:**

Е.А. Савицкая, начальник отдела  
О.А. Культепина, методист

## 1. Компетенции, формируемые в процессе изучения дисциплины

Код и наименование компетенции	Индикаторы достижения компетенции
ОПК-3 Способен принимать решения, осуществлять стратегическое планирование и прогнозирование в профессиональной деятельности с использованием современных методов и программного инструментария сбора, обработки и анализа данных, интеллектуального оборудования и систем искусственного интеллекта	ОПК-3.1 Владеет методами стратегического планирования и прогнозирования в профессиональной деятельности
	ОПК-3.2 Самостоятельно выбирает и обосновывает выбор современных методов и программного инструментария сбора, обработки и анализа данных
ПК-16 Способен готовить аналитические материалы для оценки мероприятий и выработки стратегических решений в области ИКТ	ПК-16.1 Эффективно применяет в ходе профессиональной деятельности методы и инструментари анализ данных
	ПК-16.2 Знает методы подготовки аналитических материалов для оценки мероприятий и выработки стратегических решений в области ИКТ

## 2. Показатели оценивания компетенций

В результате изучения дисциплины «Разработка финансовых решений и сервисов» обучающийся должен:

### знать:

- требования к ПО в финансовом секторе;
- моделирование систем;
- подходы к технологиям интеграции;
- архитектура информационных систем и хранение данных;
- интерпретаторы;
- основные понятия программной инженерии;
- примеры использования разработки финансового ПО.

### уметь:

- анализировать стадии проекта и предлагать решения;
- обрабатывать ошибки микросервисных систем;
- готовить требования к проектированию больших программных систем.

### владеть:

- навыками программирования Python для решения финансового ПО;
- навыками программирования Golang.

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

Проводится в форме опроса по темам предыдущего занятия и в формате автоматически проверяемых тестов на платформе.

Примеры вопросов:

1. Дайте описание аналитической стадии проекта.
2. Опишите концепцию и границы продукта для компании X.
3. Перечислите требования к ПО.
4. Опишите основные протоколы взаимодействия информационных систем (REST, SOAP, GraphQL).
5. Назовите плюсы и минусы инфраструктуры X (DevOps).
6. Опишите структуру Python проекта.
7. Опишите работу Server-side web.
8. Перечислите основные форматы данных.
9. Расскажите об этапах проектирования API.
10. Расскажите, как происходит работа с кэшами. G

11. Опишите работу интерпретатора.
12. Опишите работу микросервисов.
13. Опишите внутренности интерпретатора CPython.
14. Назовите основные принципы построения микросервисных систем, типичные проблемы и способы их решения.
15. Перечислите основы масштабируемых систем.

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

Примеры тем заданий для экзамена:

1. Использование технологий интеграции.
2. Структурирование данных в Python
3. Структура python проекта. Написание автотестов.
4. Использование WSGI. Server-side web
5. Проектирование API.
6. Проектирование СУБД. ACID. ORM.
7. Инструменты разработки NoSQL.
8. Использование GIL, потоков, процессов.
9. Применение синхронной обработки IO. Asyncio.
10. Работа с интерпретатором CPython.
11. Работа с памятью. Основы масштабируемых систем.
12. Использование синтаксиса GoLang для решения задач.
13. Паттерны pipeline и context.
14. Использование пакетов database/sql и jack/pgx.
15. Написание протоколов Golang.

Примеры билетов для экзамена:

Билет 1.

1. Работа СУБД. Использование в финтехе.
2. Откройте базу данных по ссылке и выполните задание по работе с кэшами. Пришлите ссылку на работающий код.

Билет 2.

1. Инфраструктура (DevOps).
2. Подготовьте автотест для системы. Пришлите готовый код Python.

Примеры вопросов для дифференцированного зачета:

1. Аналитическая стадия проекта.
2. Концепция и границы продукта.
3. Требования к ПО.
4. Функциональные и нефункциональные требования.
5. Моделирование систем.
6. Документирование.
7. Архитектура информационных систем и хранение данных.
8. Взаимодействие информационных систем (REST, SOAP, GraphQL).
9. Синхронное и асинхронное взаимодействие.
10. Подходы и технологии интеграции.
11. Инфраструктура (DevOps).
12. Python и структуры данных в нем.
13. Структура python проекта. Написание автотестов.
14. Сетевое взаимодействие. WSGI. Server-side web
15. Форматы данных. Проектирование API. Валидация.



16. СУБД. ACID. ORM.
17. NoSQL. Redis. Работа с кэшами.
18. GIL, потоки, процессы. WS, SSE, Longpolling.
19. Асинхронная обработка IO. Asyncio.
20. Внутренности интерпретатора CPython.
21. Работа с памятью. Основы масштабируемых систем.
22. Синтаксические конструкции языка GoLang.
23. Паттерны pipeline и context.
24. Пакеты database/sql и jack/pgx.
25. Протоколы Golang.

#### Критерии оценивания

- оценка «отлично (10)» 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
- оценка «отлично (9)» выставляется студенту, показавшему всесторонние, систематизированные, глубокие знания учебной программы дисциплины и умение применять их на практике при решении конкретных задач, свободное и правильное обоснование принятых решений
- оценка «отлично (8)» выставляется студенту, показавшему всесторонние систематизированные, глубокие знания учебной программы дисциплины и умение применять их на практике при решении конкретных задач, и правильное обоснование принятых решений
- оценка «хорошо (7)» 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;
- оценка «хорошо (6)» выставляется студенту, если он знает материал, грамотно и по существу излагает его, умеет применять полученные знания на практике, но допускает в ответе или в решении задач некоторые неточности;
- оценка «хорошо (5)» выставляется студенту, если он знает материал, и по существу излагает его, умеет применять полученные знания на практике, но допускает в ответе или в решении задач некоторые неточности;
- оценка «удовлетворительно (4)» выставляется студенту, показавшему фрагментарный, разрозненный характер знаний, недостаточно правильные формулировки базовых понятий, нарушения логической последовательности в изложении программного материала, но при этом он владеет основными разделами учебной программы, необходимыми для дальнейшего обучения и может применять полученные знания по образцу в стандартной ситуации;
- оценка «удовлетворительно (3)» выставляется студенту, показавшему фрагментарный, разрозненный характер знаний, недостаточно правильные формулировки базовых понятий, нарушения логической последовательности в изложении программного материала, но при этом он владеет фрагментарно основными разделами учебной программы, необходимыми для дальнейшего обучения и может применять полученные знания по образцу в стандартной ситуации;
- оценка «неудовлетворительно (2)» выставляется студенту, который не знает большей части основного содержания учебной программы дисциплины, допускает грубые ошибки в формулировках основных понятий дисциплины и не умеет использовать полученные знания при решении типовых практических задач
- оценка «неудовлетворительно (1)» выставляется студенту, который не знает формулировок основных понятий дисциплины.

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

Во время проведения экзамена обучающийся получает вопрос + практическое задание по работе в одном из языков.