

**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**

|                            |   |
|----------------------------|---|
|                            | <b>Рабочая программа дисциплины (модуля)</b>  |
| <b>по дисциплине:</b>      | Введение в идиоматический Kotlin  |
| <b>по направлению:</b>     | Прикладная математика и информатика   |
| <b>профиль подготовки:</b> | Проектирование и разработка комплексных бизнес-приложений<br>Физтех-школа Прикладной Математики и Информатики<br>кафедра алгоритмов и технологий программирования |
| <b>курс:</b>               | 3   |
| <b>квалификация:</b>       | бакалавр  |

Семестр, формы промежуточной аттестации: 6 (весенний) - Дифференцированный зачет

Аудиторных часов: 60 всего, в том числе:

лекции: 30 час.

семинары: 30 час.

лабораторные занятия: 0 час.

Самостоятельная работа: 75 час.

Всего часов: 135, всего зач. ед.: 3

Программу составил: О.Н. Ивченко, заведующий кафедрой

Программа обсуждена на заседании кафедры алгоритмов и технологий программирования 12.02.2024

## Аннотация

Дисциплина "Введение в идиоматический Kotlin" предоставляет участникам необходимые знания, умения и навыки для эффективной работы с языком Kotlin. Освоив представленные концепции, студенты смогут разрабатывать качественные приложения, используя лучшие практики программирования. Kotlin обладает рядом существенных преимуществ в качестве начального языка для продвинутого научного программирования:

- строгая типизация, четко построенная система типов;
- высокая производительность;
- автоматическое управление памятью;
- полная совместимость с огромным количеством библиотек на Java;
- лучший инструментарий;
- обширное сообщество;
- возможность коммерческого применения.

Дисциплина проходит при участии JetBrains и при поддержке JetBrains Research. Наиболее активные студенты получают возможность участвовать в летних стажировках в JetBrains. Также для студентов старших курсов есть возможность проходить НИР в Лаборатории методов ядерно-физических экспериментов МФТИ (участник JetBrains Research) и в московском офисе JetBrains. Студенты обучаются работать на языке Kotlin и применять его для решения научных задач. Будет сделан упор на практические аспекты и примеры, так что для его понимания не нужны никакие дополнительные знания. Для практических примеров будет использоваться среда разработки IntelliJ IDEA Community Edition.

## 1. Цели и задачи

### Цель дисциплины

- ознакомить студентов с идиоматическим стилем программирования на языке Kotlin, который включает в себя использование его возможностей для создания чистого, понятного и эффективного кода;
- научатся студенты применять лучшие практики и шаблоны проектирования, характерные для Kotlin;
- освоить ключевые концепции и особенности языка.

### Задачи дисциплины

- ознакомление с синтаксисом и основными конструкциями языка;
- понимание типизации, управления потоками и обработки исключений;
- изучение принципов идиоматического Kotlin, таких как использование расширений, лямбда-функций и коллекций;
- понимание концепций функционального программирования в Kotlin;
- изучение популярных библиотек и фреймворков, таких как Ktor, Coroutines и другие;
- применение библиотек для решения реальных задач;
- освоение принципов чистого кода и рефакторинга;
- изучение паттернов проектирования, применимых в Kotlin;
- применение полученных знаний для создания масштабируемых и поддерживаемых проектов.

## 2. Перечень формируемых компетенций

Освоение дисциплины направлено на формирование следующих компетенций:

| Код и наименование компетенции  | Индикаторы достижения компетенции   |
|---|---|
| УК-1 Способен осуществлять поиск, критический анализ и синтез информации, применять системный подход для решения поставленных задач | УК-1.1 Анализирует задачу, выделяя этапы ее решения, действия по решению задачи                           |
|   | УК-1.2 Находит, критически анализирует и выбирает информацию, необходимую для решения поставленной задачи |
|   | УК-1.3 Рассматривает различные варианты решения задачи, оценивает их преимущества и недостатки            |
|   | УК-1.4 Грамотно, логично, аргументированно формирует собственные суждения и оценки                        |

|   |   |
|---|---|
|   | УК-1.5 Определяет и оценивает практические последствия возможных вариантов решения задачи   |
| ОПК-1 Способен применять фундаментальные знания, полученные в области физико-математических и (или) естественных наук и использовать их в профессиональной деятельности   | ОПК-1.1 Способен анализировать поставленную задачу, намечать пути ее решения  |
|   | ОПК-1.2 Способен строить математические модели, производить количественные расчеты и оценки   |
|   | ОПК-1.3 Способен определять границы применимости полученных результатов   |
| ОПК-2 Способен использовать современные информационные технологии и программные средства при решении задач профессиональной деятельности, соблюдая требования информационной безопасности                                       | ОПК-2.1 Способен применять современные вычислительную технику и сервисы сети Интернет в области (сфере) профессиональной деятельности   |
|   | ОПК-2.2 Знает и умеет применять численные математические методы и прикладное программное обеспечение для решения научных задач в профессиональной области                                 |
|   | ОПК-2.3 Знает основные требования информационной безопасности   |
| ОПК-3 Способен составлять и оформлять научные и (или) технические (технологические, инновационные) отчеты (публикации, проекты)   | ОПК-3.1 Знает основные правила оформления научных публикаций и научно-технической документации, в том числе с использованием прикладного программного обеспечения                         |
|   | ОПК-3.2 Владеет на практике методологией составления научно-технических отчетов (проектов)  |
|   | ОПК-3.3 Владеет методами визуального и графического представления результатов научной (научно-технической, инновационной технологической) деятельности в виде отчетов, научных публикаций |
| 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-2.1 Знает принципы построения научной работы, методы сбора и анализа полученного материала, способы аргументации   |
|   | ПК-2.2 Способен планировать и проводить научные исследования самостоятельно или в качестве члена (руководителя) малого научного коллектива  |
|   | ПК-2.3 Способен проводить апробацию результатов научно-исследовательской работы посредством публикации научных статей и участия в конференциях  |

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

В результате освоения дисциплины обучающиеся должны знать:

- основные модели памяти, используемые в языках программирования и различных архитектурах процессоров, уметь описывать различия этих моделей памяти;
- принцип исполнения программ на Java с использованием JVM;
- типы данных языка Kotlin;
- управление потоком выполнения в Kotlin;
- иерархию классов стандартной библиотеки;
- идеологию функционального программирования;
- основы численных методов.

уметь:

- писать код на Kotlin;
- интегрировать сторонние библиотеки в свои проекты;
- работать с асинхронным кодом с помощью Coroutines;
- решать вычислительные задачи с использованием языка программирования высокого уровня;
- создавать архитектуру приложений с учетом масштабируемости и поддержки;
- применять паттерны проектирования в своих проектах;
- тестировать код;
- рефакторить код.

владеть:

- владеть инструментами разработки (IDE, системы сборки);
- создавать чистый, понятный и поддерживаемый код;
- делиться опытом и получать обратную связь от других разработчиков;
- навыками работы с объектами и потоками, и кругозором в выборе архитектурного решения поставленной задачи;
- анализом кода;
- разработкой сложных приложений.

#### 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                     | От жесткого к мягкому                 | 4   | 4        |                 | 10             |
| 2                     | Инструменты современного программиста | 6   | 6        |                 | 15             |
| 3                     | Язык Kotlin                           | 10  | 10       |                 | 20             |
| 4                     | Архитектура программы                 | 4   | 4        |                 | 15             |
| 5                     | Научное программирование              | 6   | 6        |                 | 15             |
| Итого часов           |                                       | 30  | 30       |                 | 75             |
| Подготовка к экзамену |                                       | 0 час.  |          |                 |                |
| Общая трудоёмкость    |                                       | 135 час., 3 зач.ед.   |          |                 |                |

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

Семестр: 6 (Весенний)

###### 1. От жесткого к мягкому

В рамках изучения данной темы студенты изучают следующие вопросы.

- Программа как набор инструкций. Эволюция программ.
- Структура памяти. Segmentation fault.
- Парадигмы программирования. Генеалогия языков.
- Виртуальные машины, байт-код.
- Компиляция и оптимизации.
- Статическая и динамическая линковка. Библиотеки.
- Структура программы. Точки входа.

## 2. Инструменты современного программиста

В рамках изучения данной темы студенты изучают следующие вопросы.

- Системы автоматической сборки.
- Системы контроля версий.
- Интегрированные среды разработки.

## 3. Язык Kotlin

В рамках изучения данной темы студенты изучают следующие вопросы.

- Переменные, классы и объекты.
- Control flow. Процедурный и функциональный подход.
- Замыкания.
- Структуры данных и операции над ними.
- Свойства и делегаты.
- Параметрические типы.
- Расширения.
- Боксинг.
- Мультиплатформные проекты.

## 4. Архитектура программы

В рамках изучения данной темы студенты изучают следующие вопросы.

- Абстракции и интерфейсы.
- Основы коллективной разработки при помощи современных инструментов.
- Идеология объектного программирования. Разделение поведений.
- Идеология функционального программирования.

## 5. Научное программирование

В рамках изучения данной темы студенты изучают следующие вопросы.

- Основы численных методов. Понятие о численной точности. Сложность алгоритмов.
- Численное дифференцирование и интегрирование.
- Генераторы случайных чисел и Монте-Карло моделирование.
- Высокопроизводительные параллельные и конкурентные вычисления.
- Проблема ввода-вывода и основные методы ее решения.
- Системы сбора данных. Протоколы обмена данными.
- Основы работы с большими данными.
- Поточная обработка данных.

## 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. Kotlin в действии, Электрон. версия печ. публикации / Д. Жемеров, С. Исакова. — Москва, ДМК Пресс, 2018

## Дополнительная литература

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

- Kotlin in Action - Дмитрий Жемеров, Светлана Исакова.
- Kotlin Programming: The Big Nerd Ranch Guide - Josh Skeen, David Greenhalgh.

### **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)**

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

- посещения студентом всех видов аудиторных занятий;
- ведения конспекта в ходе лекционных занятий;
- качественной самостоятельной подготовки к практическим занятиям, активной работы на них;
- активной самостоятельной и аудиторной работы студента;
- своевременной сдачи преподавателю заданий по аудиторным видам работ.

**ОЦЕНОЧНЫЕ МАТЕРИАЛЫ ПО ДИСЦИПЛИНЕ (МОДУЛЮ)**

|                            |   |
|----------------------------|---|
| <b>по направлению:</b>     | Прикладная математика и информатика   |
| <b>профиль подготовки:</b> | Проектирование и разработка комплексных бизнес-приложений<br>Физтех-школа Прикладной Математики и Информатики<br>кафедра алгоритмов и технологий программирования |
| <b>курс:</b>               | <u>3</u>  |
| <b>квалификация:</b>       | бакалавр  |

Семестр, формы промежуточной аттестации: 6 (весенний) - Дифференцированный зачет

**Разработчик:** О.Н. Ивченко, заведующий кафедрой

# 1. Компетенции, формируемые в процессе изучения дисциплины

| Код и наименование компетенции  | Индикаторы достижения компетенции   |
|---|---|
| УК-1 Способен осуществлять поиск, критический анализ и синтез информации, применять системный подход для решения поставленных задач   | УК-1.1 Анализирует задачу, выделяя этапы ее решения, действия по решению задачи   |
|   | УК-1.2 Находит, критически анализирует и выбирает информацию, необходимую для решения поставленной задачи   |
|   | УК-1.3 Рассматривает различные варианты решения задачи, оценивает их преимущества и недостатки  |
|   | УК-1.4 Грамотно, логично, аргументированно формирует собственные суждения и оценки  |
|   | УК-1.5 Определяет и оценивает практические последствия возможных вариантов решения задачи   |
| ОПК-1 Способен применять фундаментальные знания, полученные в области физико-математических и (или) естественных наук и использовать их в профессиональной деятельности   | ОПК-1.1 Способен анализировать поставленную задачу, намечать пути ее решения  |
|   | ОПК-1.2 Способен строить математические модели, производить количественные расчеты и оценки   |
|   | ОПК-1.3 Способен определять границы применимости полученных результатов   |
| ОПК-2 Способен использовать современные информационные технологии и программные средства при решении задач профессиональной деятельности, соблюдая требования информационной безопасности                                       | ОПК-2.1 Способен применять современные вычислительную технику и сервисы сети Интернет в области (сфере) профессиональной деятельности   |
|   | ОПК-2.2 Знает и умеет применять численные математические методы и прикладное программное обеспечение для решения научных задач в профессиональной области                                 |
|   | ОПК-2.3 Знает основные требования информационной безопасности   |
| ОПК-3 Способен составлять и оформлять научные и (или) технические (технологические, инновационные) отчеты (публикации, проекты)   | ОПК-3.1 Знает основные правила оформления научных публикаций и научно-технической документации, в том числе с использованием прикладного программного обеспечения                         |
|   | ОПК-3.2 Владеет на практике методологией составления научно-технических отчетов (проектов)  |
|   | ОПК-3.3 Владеет методами визуального и графического представления результатов научной (научно-технической, инновационной технологической) деятельности в виде отчетов, научных публикаций |
| 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-2.1 Знает принципы построения научной работы, методы сбора и анализа полученного материала, способы аргументации   |
|   | 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 «Введение в идиоматический Kotlin» обучающийся должен:

### знать:

- основные модели памяти, используемые в языках программирования и различных архитектурах процессоров, уметь описывать различия этих моделей памяти;
- принцип исполнения программ на Java с использованием JVM;
- типы данных языка Kotlin;
- управление потоком выполнения в Kotlin;
- иерархию классов стандартной библиотеки;
- идеологию функционального программирования;
- основы численных методов.

### уметь:

- писать код на Kotlin;
- интегрировать сторонние библиотеки в свои проекты;
- работать с асинхронным кодом с помощью Coroutines;
- решать вычислительные задачи с использованием языка программирования высокого уровня;
- создавать архитектуру приложений с учетом масштабируемости и поддержки;
- применять паттерны проектирования в своих проектах;
- тестировать код;
- рефакторить код.

### владеть:

- владеть инструментами разработки (IDE, системы сборки);
- создавать чистый, понятный и поддерживаемый код;
- делиться опытом и получать обратную связь от других разработчиков;
- навыками работы с объектами и потоками, и кругозором в выборе архитектурного решения поставленной задачи;
- анализом кода;
- разработкой сложных приложений.

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

Примеры практических заданий в течение семестра.

1. Напишите программу, которая выводит на экран "Hello, Kotlin!". Затем измените программу так, чтобы она принимала имя пользователя в качестве аргумента и выводила "Hello, [имя]!".
2. Создайте список чисел от 1 до 100. Напишите функцию, которая возвращает только четные числа из этого списка. Используйте функциональные возможности Kotlin, такие как `filter`
3. Напишите расширение функции для класса `String`, которое будет возвращать количество гласных букв в строке. Протестируйте эту функцию на нескольких строках.
4. Реализуйте функцию, которая принимает строку и пытается преобразовать её в целое число. Если преобразование не удалось, функция должна возвращать значение `null`. Используйте обработку исключений для управления ошибками.
5. Напишите функцию `applyOperation`, которая принимает два числа и функцию (лямбда-выражение), выполняющую операцию над этими числами. Протестируйте её с различными операциями (сложение, вычитание, умножение).
6. Создайте класс `Person` с полями `name` и `age`. Реализуйте метод `greet`, который выводит приветствие с указанием имени и возраста. Создайте несколько объектов этого класса и вызовите метод `greet`.
7. Создайте `data class Book` с полями `title`, `author` и `year`. Напишите функцию, которая принимает список книг и возвращает список книг, написанных после 2000 года.
8. Создайте функцию, которая принимает строку и возвращает её длину. Если строка равна `null`, функция должна возвращать 0. Используйте безопасный вызов и оператор Эльвиса.

9. Напишите простую программу, использующую корутины для выполнения асинхронной задачи (например, имитация загрузки данных). Реализуйте функцию, которая делает "задержку" на 2 секунды перед возвратом результата.
10. Используйте библиотеку `Kotlinx.serialization` для сериализации и десериализации объекта. Создайте класс `User` с полями `name` и `email`, затем сериализуйте его в JSON и обратно.

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

1. Что такое Kotlin и какие его основные преимущества по сравнению с Java?
2. Как объявить переменные в Kotlin? В чем разница между `val` и `var`?
3. Что такое типы данных в Kotlin? Приведите примеры.
4. Как работает система типов в Kotlin? Что такое nullable типы?
5. Как использовать условные операторы (`if`, `when`) в Kotlin? Приведите примеры.
6. Как реализовать циклы в Kotlin? В чем разница между `for`, `while` и `do while`?
7. Как объявить функцию в Kotlin? Что такое параметры по умолчанию?
8. Что такое лямбда-выражения и как они используются в Kotlin?
9. Как работают функции высшего порядка в Kotlin? Приведите пример.
10. Как создать и использовать списки, множества и карты в Kotlin?
11. Что такое расширяющие функции и как они применяются к коллекциям?
12. Как использовать стандартные функции для работы с коллекциями, такие как `map`, `filter`, `reduce`?
13. Как объявить класс и объект в Kotlin? В чем разница между классом и интерфейсом?
14. Что такое наследование в Kotlin и как оно реализуется?
15. Как работают data-классы в Kotlin? В каких случаях их следует использовать?
16. Что такое идиоматический код на Kotlin? Приведите примеры идиоматических решений.
17. Как использовать sealed классы для реализации ограниченных классов?
18. Что такое оператор `infix` и как его использовать?
19. Как обрабатывать исключения в Kotlin? В чем отличие от Java?
20. Что такое try-catch блоки и как они работают в контексте Kotlin?
21. Что такое корутины в Kotlin и как они помогают в асинхронном программировании?
22. Как создать корутину и запустить ее? Приведите пример использования `launch` и `async`.
23. Напишите функцию, которая принимает список чисел и возвращает их сумму.
24. Создайте data-класс для представления пользователя с полями `name`, `age`, и реализуйте метод для вывода информации о пользователе.
25. Реализуйте функцию, которая принимает строку и возвращает количество гласных букв в ней.

#### Критерии оценивания

Оценка "Отлично" (10) - полностью и вовремя решены все задачи без ошибок. Продemonстрирован грамотный подход к решению задач, реализованы оптимальные алгоритмы, код оформлен в едином удобочитаемом стиле.

Оценка "Отлично" (9) - полностью и вовремя решены все задачи без ошибок. Продemonстрирован грамотный подход к решению задач, реализованы оптимальные алгоритмы.

Оценка "Отлично" (8) - полностью и вовремя решены все задачи без ошибок. Продemonстрирован грамотный подход к решению задач.

Оценка "Хорошо" (7) - полностью решены все задачи. Допущены несущественные ошибки.

Оценка "Хорошо" (6) - полностью решено большинство задач. В некоторых задачах допущены и не исправлены ошибки, либо некоторые задачи решены частично.

Оценка "Хорошо" (5) - полностью решено две трети задач. В некоторых задачах допущены и не исправлены ошибки, либо некоторые задачи решены частично.

Оценка "Удовлетворительно" (4) - полностью решено более половины задач. В остальных задачах допущены и не исправлены ошибки, либо некоторые задачи решены частично.

Оценка "Удовлетворительно" (3) - полностью решено более половины задач.

Оценка "Неудовлетворительно" (2) - решено менее половины задач.

Оценка "Неудовлетворительно" (1) - не решено ни одной задачи.

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

Во время проведения дифференцированного зачета обучающиеся могут пользоваться программой дисциплины.