

**Санкт-Петербургский филиал федерального государственного  
автономного образовательного учреждения высшего образования  
«Национальный исследовательский университет  
"Высшая школа экономики"»**

Факультет Санкт-Петербургская школа  
физико-математических и компьютерных наук  
Департамент информатики

**Рабочая программа дисциплины  
Альтернативные языки для JVM**

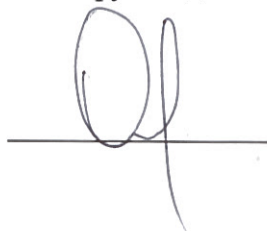
для образовательной программы «Прикладная математика и информатика»  
направления подготовки 01.03.02 «Прикладная математика и информатика»  
уровень бакалавриат

Разработчик: Браславский Павел Исаакович, pbraslavski@hse.ru

Утверждена Академическим руководителем образовательной программы

«31» августа 2018 г.

А.В. Омельченко



Санкт-Петербург, 2018

*Настоящая программа не может быть использована другими подразделениями университета и другими вузами без разрешения подразделения-разработчика программы.*

## 1. Область применения и нормативные ссылки

Настоящая программа учебной дисциплины устанавливает требования к образовательным результатам и определяет содержание и виды учебных занятий и отчетности.

Программа предназначена для преподавателей, ведущих дисциплину «Альтернативные языки для JVM», учебных ассистентов и студентов направления 01.03.02 «Прикладная математика и информатика» подготовки бакалавра, обучающихся по бакалаврской программе «Прикладная математика и информатика» и изучающих дисциплину «Альтернативные языки для JVM».

Рабочая программа дисциплины разработана в соответствии с:

- Образовательным стандартом НИУ ВШЭ по направлению подготовки 01.03.02 «Прикладная математика и информатика» (уровень бакалавриата), утвержденным ученым советом Национального исследовательского университета «Высшая школа экономики», протокол от 03.03.2017 №02.
- Основной профессиональной образовательной программой «Прикладная математика и информатика» направления подготовки 01.03.02 «Прикладная математика и информатика»;
- Объединенным учебным планом университета по образовательной программе «Прикладная математика и информатика», утвержденным в 2018 г.

## 2. Цели освоения дисциплины

Целями освоения дисциплины «Альтернативные языки для JVM» являются формирование у студентов теоретических знаний и практических навыков работы различными языками для виртуальной машины Java.

## 3. Компетенции обучающегося, формируемые в результате освоения дисциплины

В результате освоения дисциплины студент должен:

- Знать различные языки для JVM.
- Уметь создавать программы на Scala, Groovy, Kotlin.
- Иметь навыки (приобрести опыт) использования инструментальных средств для написания обобщенных программ на различных языках для JVM.

В результате освоения дисциплины студент осваивает следующие компетенции:

Компетенция	Код по ОС НИУ ВШЭ	Уровень формирования компетенции	Дескрипторы – основные признаки освоения (показатели достижения результата)	Формы и методы обучения, способствующие формированию и развитию компетенции	Форма контроля уровня сформированности и компетенции
Способен применять и модифицировать математические модели для решения задач в области профессиональной деятельности	ОПК-3	РБ СД МЦ	Знает какие алгоритмы выразимы на языках Kotlin и Scala. Применяет стандартную библиотеку и различные фреймворки данных языков для решения прикладных задач. Разрабатывает собственные библиотеки.	Семинары, подготовка к семинарам, самостоятельная работа	Домашние задания, письменный экзамен
Способен решать задачи профессиональной деятельности с использованием существующих информационно-коммуникационных технологий и с учетом основных	ОПК-4	РБ СД МЦ	Реализует эффективные многопоточные приложения на языке Kotlin. Выбирает правильную архитектуру для достижения наибольшей производительности и безопасности приложения. Владеет инструментами	Семинары, подготовка к семинарам, самостоятельная работа	Домашние задания, письменный экзамен

требований информационной безопасности			многопоточной верификации программ, написанных на JVM-языках.		
Способен разрабатывать и реализовывать в виде программного модуля алгоритм решения поставленной теоретической или прикладной задачи на основе математической модели	ПК-2	РБ СД МЦ	Знает основные принципы функционирования программ на JVM-языках. Создаёт программы на языках Kotlin, Scala и им подобных. Владеет концепциями работы виртуальной машины Java, необходимыми для создания программ на JVM-языках.	Семинары, подготовка к семинарам, самостоятельная работа	Домашние задания, письменный экзамен
Способен разрабатывать программное и информационное обеспечение компьютерных систем, сервисов, вычислительных комплексов, баз данных	ПК-3	РБ СД МЦ	Знает основы программирования на языках программирования Scala и Kotlin. Понимает теоретические концепции и корректно использовать парадигмы данных языков. Использует данные языки в любых проектах для JVM-платформы.	Семинары, подготовка к семинарам, самостоятельная работа	Домашние задания, письменный экзамен
Способен осуществлять планирование профессиональной деятельности, связанной с созданием и использованием информационных систем	ПК-7	РБ СД МЦ	Знает основные этапы создания проекта на JVM языках. Реализует разработанную архитектуру. Отлаживает и тестирует программы, написанные на JVM языках.	Семинары, подготовка к семинарам, самостоятельная работа	Домашние задания, письменный экзамен

#### 4. Место дисциплины в структуре образовательной программы

Для образовательной программы «Прикладная математика и информатика» направления подготовки 01.03.02 «Прикладная математика и информатика» настоящая дисциплина относится к дисциплинам по выбору блока дисциплин.

Для освоения дисциплины необходимы компетенции, полученные в ходе изучения дисциплин:

- Основы и методология программирования
- Язык программирования Java

#### 5. Тематический план учебной дисциплины

Курс рассчитан на 30 часов аудиторной нагрузки, из них 30 часов семинаров, общим объемом 4 зачетных единицы (152 часов).

№	Название раздела	Всего часов	Аудиторные часы			Самостоятельная работа
			Лекции	Семинары	Практические занятия	
1	Введение. Знакомство с ОО и функциональными парадигмами языка Scala	34	0	6	0	28
2	Совместное использование Scala и Java, система типов Scala	42	0	9	0	33
3	Знакомство с языком Kotlin	34	0	6	0	28
4	Совместное использование Kotlin и Java, построение DSL и корутины в языке Kotlin	42	0	9	0	33
ИТОГО		152	0	30	0	122

## 6. Содержание дисциплины

<u>Раздел 1</u> Введение. Знакомство с ОО и функциональными парадигмами языка Scala	
Тема 1	Введение. Знакомство с ОО парадигмой языка программирования Scala. Обзор истории развития “альтернативных JVM-языков”. Обзор базовых (объектно-ориентированных) возможностей языка Scala. Понятие “синтаксического сахара” в языке программирования и примеры его использования в Scala.
Тема 2	Знакомство с функциональной парадигмой языка программирования Scala. Обзор функциональных возможностей Scala. Обзор стандартной библиотеки коллекций языка, сравнение со стандартной библиотекой коллекций языка Java.
<u>Раздел 2</u> Совместное использование Scala и Java, система типов Scala	
Тема 1	Особенности совместного использования Scala и Java. Множественное наследование и детали его реализации в Scala. Описание и примеры использования возможностей Scala, отсутствующих в Java.
Тема 2	Обзор системы типов Scala. Знакомство с понятиями системы типов Scala, вывод типов.
Тема 3	Программирование на системе типов Scala. Ключевое слово <code>implicit</code> и область его применения. Базовые навыки программирования на системе типов, обзор некоторые элементов теории категорий.
<u>Раздел 3</u> Знакомство с языком Kotlin	
Тема 1	Знакомство с парадигмой языка программирования Kotlin. Обзор базовых возможностей языка Kotlin. Основные области применения и принципы разработки ПО с помощью Kotlin.
Тема 2	Обзор нововведений языка программирования Kotlin. Описание и примеры использования возможностей Kotlin, отсутствующих в Java.

<b>Раздел 4</b> Совместное использование Kotlin и Java, построение DSL и корутины в языке Kotlin	
Тема 1	Особенности совместного использования Kotlin и Java. Совместная компиляция Java и Kotlin в рамках одного проекта. Специфика различий в системах типов и их взаимная аппроксимация. Аннотации исходного кода для контроля получаемых JVM-сигнатур.
Тема 2	Построение DSL в Kotlin. Особенности построения доменно-специфичных языков на основе синтаксических особенностей Kotlin. Пример реализации DSL для языка разметки HTML.
Тема 3	Корутины в Kotlin. Введение в концепцию асинхронного программирования. Описание реализации CPS-функций в Kotlin. Обзор существующих библиотек, построенных на основе корутин в Kotlin.

## 7. Оценочные средства

### 7.1. Формы контроля знаний студентов

Тип контроля	Форма контроля	3 год		Параметры
		1 модуль	2 модуль	
Текущий	Домашнее задание №1	*		Письменное домашнее задание
	Домашнее задание №2	*		Письменное домашнее задание
	Домашнее задание №3	*		Письменное домашнее задание
	Домашнее задание №4		*	Письменное домашнее задание
	Домашнее задание №5		*	Письменное домашнее задание
	Домашнее задание №6		*	Письменное домашнее задание
	Домашнее задание №7		*	Письменное домашнее задание
Итоговый	Письменный экзамен		*	Экзамен в письменной форме

### 7.2. Критерии оценки и шкалы, примеры заданий

Текущий контроль успеваемости по дисциплине осуществляется с помощью следующих оценочных средств.

#### ДОМАШНИЕ ЗАДАНИЯ №1-7

*Домашнее задание №№1-7 выдается студентам в одном варианте. Срок выполнения домашнего задания - 2 недели. Каждое домашнее задание должно быть выполнено в виде компилируемого проекта в системе сборки Gradle или SBT, опубликованного в виде ветки на публичном репозитории GitHub. Форма представления обучающимися домашнего задания -*

представленные в письменном виде решения задач.

### Примеры домашних заданий:

#### Задача 1

- Требуется реализовать калькулятор, вычисляющий значение выражения из входного потока и выводящий ответ в выходной поток.
- Калькулятор должен уметь выполнять базовые арифметические операции.

#### Задача 2

- Требуется реализовать мультимножество.
- Реализация должна поддерживать функции высшего порядка.
- Реализация должна быть покрыта набором тестов, проверяющим как семантическую, так и синтаксическую валидность.

#### Задача 3

- Требуется реализовать гетерогенный список.
- Требуется реализовать операцию разделения гетерогенного списка на два по индексу. Возможность выполнения данной операции должна проверяться на этапе компиляции.

#### Задача 4

- Необходимо реализовать решение задачи с сайта [codeforces.ru](https://codeforces.ru/) (<https://codeforces.ru/>) не ниже уровня сложности В на языке программирования Kotlin.
- В комментариях к ветке должна быть ссылка на успешное решение.

#### Задача 5

- Реализуйте интерпретатор простого языка программирования “fun”.
- Для разбора текста программы следует использовать библиотеку “antlr”.

#### Задача 6

- Создайте библиотеку и сопутствующий доменно-специфичный язык (DSL) для описания документа в формате TeX.
- Воспользуйтесь “DslMarker”, чтобы тег “item” нельзя было использовать вне “itemize/enumerate”.

#### Задача 7

- Используя результаты домашнего задания № 5, с помощью корутин добавьте возможность отладки программы на языке “fun”.
- В качестве точек “заморозки” следует использовать активные точки останова.

### Критерии оценивания и шкала оценки домашнего задания №№1-7

Оценка	Критерии выставления оценки
«Отлично» (8-10)	Для получения оценки “отлично” работа должна быть сдана вовремя, успешно скомпилироваться и не содержать стилистических погрешностей, перечисленных в разделе “Style guide” соответствующего языка ( <a href="http://scalalang.org">scalalang.org</a> , <a href="http://kotlinlang.org">kotlinlang.org</a> ). Кроме этого в работе должно отсутствовать дублирование кода, а само решение должно быть одновременно кратким и понятным для проверяющего.
«Хорошо» (6-7)	Для получения оценки “хорошо” работа должна быть сдана вовремя, успешно скомпилироваться и не содержать стилистических погрешностей, перечисленных в разделе “Style guide” соответствующего языка ( <a href="http://scalalang.org">scalalang.org</a> , <a href="http://kotlinlang.org">kotlinlang.org</a> ).
«Удовлетворительно» (4-5)	Для получения оценки “удовлетворительно” работа должна быть сдана вовремя, успешно скомпилироваться.
«Неудовлетворительно» (0-3)	Работа не сдана вовремя, не компилируется, содержит стилистические погрешности.

### 7.2.2. Итоговый контроль по дисциплине

Проверка качества освоения дисциплины производится в форме письменного экзамена.

#### ПИСЬМЕННЫЙ ЭКЗАМЕН

В рамках письменного экзамена требуется решить не менее пяти задач. Каждая задача должна быть реализована двумя способами: type level и value level (для проверки корректности).

##### Пример задачи:

В Англии валюта состоит из фунта стерлингов и пенса (p) и в обращении находится восемь монет:

1p, 2p, 5p, 10p, 20p, 50p, £ 1 (100p) и £ 2 (200p).

Получить £ 2 фунта стерлингов можно следующим образом:

$$1 \times £ 1 + 1 \times 50p + 2 \times 20p + 1 \times 5p + 1 \times 2p + 3 \times 1p$$

Сколько разных способов существует для получения £ 2, используя любое количество монет?

#### Критерии оценивания и шкала оценки письменного экзамена

Оценка	Критерии выставления оценки
«Отлично» (8-10)	Решено 8 задач. Обучающийся глубоко и прочно усвоил весь программный материал, исчерпывающе, последовательно, грамотно и логически стройно его излагает, не затрудняется с ответом при видоизменении задания, свободно справляется с задачами и практическими заданиями, правильно обосновывает принятые решения, умеет самостоятельно обобщать и излагать материал, не допуская ошибок.
«Хорошо» (6-7)	Решено 7 задач. Обучающийся твердо знает программный материал, грамотно и по существу излагает его, не допускает существенных неточностей в ответе на вопрос, может правильно применять теоретические положения и владеет необходимыми умениями и навыками при выполнении практических заданий
«Удовлетворительно» (4-5)	Решено 6 задач. Обучающийся усвоил только основной материал, но не знает отдельных деталей, допускает неточности, недостаточно правильные формулировки, нарушает последовательность в изложении программного материала и испытывает затруднения в выполнении практических заданий.
«Неудовлетворительно» (0-3)	Решено менее 5 задач. Обучающийся не знает значительной части программного материала, допускает существенные ошибки, с большими затруднениями выполняет практические задания, задачи

### 7.3 Порядок формирования оценок по дисциплине

Преподаватель учитывает оценку за текущий контроль (домашние задания).

$$O_{\text{накопленная}} = \frac{O_{д/з1} + O_{д/з2} + O_{д/з3} + O_{д/з4} + O_{д/з5} + O_{д/з6} + O_{д/з7}}{7}$$

Действует следующий способ округления накопленной оценки за текущий контроль: при значениях от 0,1 до 0,4 оценка округляется в меньшую сторону, от 0,5 до 0,9 – в большую.

Результирующая оценка за дисциплину рассчитывается следующим образом:

$$O_{\text{Результирующая}} = 0,5 O_{\text{накопленная}} + 0,5 O_{\text{экзамен}}$$

На экзамене студенту не предоставляется возможность получить дополнительный балл для

компенсации оценки за текущий контроль.

## **8. Образовательные технологии**

Основными образовательными технологиями являются: работа в группах на семинарских занятиях.

### **9. Учебно-методическое и информационное обеспечение дисциплины**

#### **9.1 Основная литература**

1. Pro Android with Kotlin: Developing Modern Mobile Apps / Späth, Peter. Apress. 2018
2. Programming with Scala / Bhim P. Upadhyaya. Springer International Publishing. 2017

#### **9.2 Дополнительная литература**

1. Scala for Java Developers: A Practical Primer Weston, Toby. Apress. 2018
2. A Beginner's Guide to Scala, Object Orientation and Functional Programming. John Hunt. Springer, 2014.
3. Scala Puzzlers / Andrew Phillips; Nermin Šerifovic. Artima. 2014

## **10. Рекомендации для самостоятельной работы студентов**

Самостоятельная работа может рассматриваться как организационная форма обучения – система педагогических условий, обеспечивающих управление учебной деятельностью по освоению знаний и умений в области учебной деятельности без посторонней помощи. Студенту нужно четко понимать, что самостоятельная работа – не просто обязательное, а необходимое условие для получения знаний по дисциплине и развитию компетенций, необходимых в будущей профессиональной деятельности.

Самостоятельная работа проводится с целью:

- систематизации и закрепления полученных на лекциях теоретических знаний;
- углубления и расширения теоретических знаний;
- формирования умений использовать нормативную, правовую, справочную документацию и специальную литературу;
- развития познавательных способностей и активности студентов: творческой инициативы, самостоятельности, ответственности и организованности;
- формирования самостоятельности мышления, способностей к саморазвитию, самосовершенствованию и самореализации;
- формирования практических (общеучебных и профессиональных) умений и навыков;
- развития исследовательских умений;
- получения навыков эффективной самостоятельной профессиональной (практической и научно-теоретической) деятельности.

В учебном процессе выделяют два вида самостоятельной работы:

- аудиторная;
- внеаудиторная.

Аудиторная самостоятельная работа по дисциплине выполняется на учебных занятиях под непосредственным руководством преподавателя и по его заданию.

Внеаудиторная самостоятельная работа - планируемая учебная работа студентов, выполняемая во внеаудиторное время по заданию и при методическом руководстве преподавателя, но без его непосредственного участия.

Самостоятельная работа, не предусмотренная программой учебной дисциплины, раскрывающей и конкретизирующей ее содержание, осуществляется студентом инициативно, с целью реализации собственных учебных и научных интересов.

Для более эффективного выполнения самостоятельной работы по дисциплине преподаватель рекомендует источники для работы, характеризует наиболее рациональную методику самостоятельной



работы, демонстрирует ранее выполненные студентами работы и т. п.

Виды заданий для внеаудиторной самостоятельной работы, их содержание и характер могут иметь вариативный и дифференцированный характер, учитывать индивидуальные особенности студента.

Самостоятельная работа может осуществляться индивидуально или группами студентов online и на занятиях в зависимости от цели, объема, конкретной тематики самостоятельной работы, уровня сложности.

Контроль результатов внеаудиторной самостоятельной работы осуществляется в пределах времени, отведенного на обязательные учебные занятия по дисциплине на практических занятиях.

## **11. Материально-техническое обеспечение дисциплины и информационные технологии, используемые при осуществлении образовательного процесса по дисциплине, включая перечень программного обеспечения информационных справочных систем (при необходимости).**

Для проведения всех занятий используется проектор и компьютер для проекции слайдов. Для самостоятельной работы необходимо следующее ПО: Java, Scala и Kotlin.

## **12. Особенности организации обучения для лиц с ограниченными возможностями здоровья**

В случае необходимости, обучающимся из числа лиц с ограниченными возможностями здоровья (по заявлению обучающегося) могут предлагаться следующие варианты восприятия учебной информации с учетом их индивидуальных психофизических особенностей, в том числе с применением электронного обучения и дистанционных технологий:

1) *для лиц с нарушениями зрения:* в печатной форме увеличенным шрифтом; в форме электронного документа; в форме аудиофайла (перевод учебных материалов в аудиоформат); индивидуальные консультации с привлечением тифлосурдопереводчика; индивидуальные задания и консультации.

2) *для лиц с нарушениями слуха:* в печатной форме; в форме электронного документа; видеоматериалы с субтитрами; индивидуальные консультации с привлечением сурдопереводчика; индивидуальные задания и консультации.

3) *для лиц с нарушениями опорно-двигательного аппарата:* в печатной форме; в форме электронного документа; в форме аудиофайла; индивидуальные задания и консультации.