

**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:	Технологии разработки на языке Java
по направлению:	Прикладная математика и информатика
профиль подготовки:	Прикладная математика и информатика Физтех-школа Прикладной Математики и Информатики кафедра алгоритмов и технологий программирования
курс:	1
квалификация:	магистр

Семестр, формы промежуточной аттестации: 2 (весенний) - Экзамен

Аудиторных часов: 30 всего, в том числе:

лекции: 0 час.

семинары: 30 час.

лабораторные занятия: 0 час.

Самостоятельная работа: 30 час.

Подготовка к экзамену: 30 час.

Всего часов: 90, всего зач. ед.: 2

Программу составил: О.Н. Ивченко, старший преподаватель

Программа обсуждена на заседании кафедры алгоритмов и технологий программирования 07.02.2025

Аннотация

Дисциплина "Технологии разработки на языке Java" предназначена для студентов, желающих получить глубокие знания и практические навыки в разработке программного обеспечения с использованием языка Java. Курс охватывает широкий спектр тем, начиная от основ языка и заканчивая современными подходами к разработке, таким как объектно-ориентированное программирование, многопоточность, работа с базами данных и веб-технологиями.

Студенты изучат синтаксис и семантику Java, а также ключевые концепции, такие как классы, объекты, наследование, полиморфизм и интерфейсы. Особое внимание будет уделено современным фреймворкам и библиотекам, таким как Spring и Hibernate, которые значительно упрощают разработку сложных приложений.

Курс также включает практические занятия, на которых студенты смогут применить полученные знания для создания реальных проектов. Они освоят инструменты разработки, такие как интегрированные среды разработки (IDE), системы контроля версий и методологии Agile.

Кроме того, дисциплина рассматривает лучшие практики программирования, включая тестирование, отладку и документирование кода. Студенты также познакомятся с принципами проектирования архитектуры приложений и основами DevOps.

1. Цели и задачи

Цель дисциплины

- овладение студентами правил языка программирования Java и современными инструментами разработки, являющимися стандартами в Java-среде.

Задачи дисциплины

- приобретение студентами навыков проектирования и реализации приложений на языке Java с использованием приемов объектно-ориентированного программирования, примитивов многопоточности и веб-технологий;
- овладение студентами современных практик разработки: использование IDE, системы контроля версий, unit-тестирования.

2. Перечень формируемых компетенций

Освоение дисциплины направлено на формирование следующих компетенций:

Код и наименование компетенции	Индикаторы достижения компетенции
УК-2 Способен управлять проектом на всех этапах его жизненного цикла	УК-2.1 Формулирует в рамках обозначенной проблемы цель, задачи, актуальность, значимость (научную, практическую, методическую и иную в зависимости от типа проекта), ожидаемые результаты и возможные сферы их применения
	УК-2.2 Способен прогнозировать результат деятельности и планировать последовательность шагов для достижения данного результата. Формирует план-график реализации проекта в целом и план контроля его выполнения
	УК-2.3 Способен организовать и координировать работу участников проекта, обеспечивать работу команды необходимыми ресурсами
	УК-2.4 Представляет публично результаты проекта (или отдельных его этапов) в форме отчетов, статей, выступлений на научно-практических конференциях, семинарах и т.п.
	ОПК-6.1 Способен анализировать задачу, планировать пути решения, предлагать и комбинировать способы решения
	ОПК-6.2 Способен разрабатывать и модернизировать программное и аппаратное обеспечение информационных и автоматизированных систем

ОПК-6 Способен выбирать и (или) разрабатывать подходы к решению типовых и новых задач в области информатики и вычислительной техники, учитывая особенности и ограничения различных методов решения	ОПК-6.3 Способен использовать исследовательские методы при решении новых задач, применяя знания из различных областей науки (техники)
	ОПК-6.4 Владеет аналитическими и вычислительными методами решения, понимает и учитывает на практике границы применимости получаемых решений
	ОПК-6.5 Способен самостоятельно приобретать, развивать и применять математические, естественнонаучные, социально-экономические и профессиональные знания для решения нестандартных задач, в том числе в новой или незнакомой среде и в междисциплинарном контексте

3. Перечень планируемых результатов обучения по дисциплине (модулю)

В результате освоения дисциплины обучающиеся должны

знать:

- принцип исполнения программ на Java с использованием JVM;
- принцип работы сборки мусора в Java;
- типы данных языка Java;
- управление потоком выполнения в Java;
- основные классы и возможности стандартной библиотеки;
- правила работы с исключениями;
- принципы разработки параметризованных классов и методов (generics);
- внутреннее строение контейнеров стандартной библиотеки и временную сложность операций с ними;
- потоковая обработка данных при помощи Stream API;
- взаимодействие с реляционными СУБД с помощью JDBC API;
- принципы разработки многопоточного кода в Java и инструменты стандартной библиотеки;
- модель памяти Java;
- возможности Java Reflection API;
- применение аннотаций и обработка аннотаций на уровне Reflection API;
- системы непрерывной интеграции и их использование в среде Java.

уметь:

- реализовывать библиотеку общего назначения на языке Java по заданным интерфейсам;
- добавлять в приложение поддержку многопоточности, анализировать потокобезопасность реализации;
- покрывать код unit-тестами с использованием фреймворка JUnit, анализировать покрытие кода тестами;
- работать с распределенной системой контроля версий git;
- использовать средства code review на сервисе Github или GitLab.

владеть:

- навыками работы с объектами и потоками и кругозором в выборе архитектурного решения поставленной задачи.

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	Синтаксис Java		4		5

2	Дополнительные элементы. Массивы, коллекции и др.		5		5
3	ООП в языке Java		5		5
4	Рефлексия в Java		5		5
5	Связь Java приложений с внешними сервисами		5		5
6	Многопоточность в Java		6		5
Итого часов			30		30
Подготовка к экзамену		30 час.			
Общая трудоёмкость		90 час., 2 зач.ед.			

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

Семестр: 2 (Весенний)

1. Синтаксис Java

Ключевые слова, идентификаторы, типы данных, литералы. Ветвления, циклы и метки. Примитивные типы и объекты в Java. Приведение типов. Inboxing, outboxing.

2. Дополнительные элементы. Массивы, коллекции и др.

Массивы и коллекции. Строки и работа с ними.

3. ООП в языке Java

Классы (абстрактные, вложенные, внутренние, анонимные, локальные...), интерфейсы. Класс Object и всё, что с ним связано. Generics, enums (enums as classes). Asserts

4. Рефлексия в Java

Java Reflection API, Аннотации

5. Связь Java приложений с внешними сервисами

Связь с базами данных, JDBC.

Экспорт данных в Excel. Apache POI

Построение графиков, JFreeChart

6. Многопоточность в Java

Создание и объединение потоков. Планирование запуска задач в потоках.

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

1. Аудитория с доской и проектором/плазменной панелью.
2. Компьютеры либо ноутбуки с доступом в интернет.

6.Перечень рекомендуемой литературы

Основная литература

Фонд кафедры:

1. Методы и средства вычислений с объектами. Аппликативные вычислительные системы [Текст] : [учеб. пособие для вузов] / В. Э. Вольфенгаген ; Ин-т актуального образования <ЮрИнфоР-МГУ>, Каф. перспективных компьютерных исследований и информационных технологий. — М. : JurInfoR, 2004. — 789 с. — (Компьютерные науки и информационные технологии). - 2000 экз. - ISBN 5-89158-100-0 (в пер.) .

2. Комбинаторная логика в программировании. Вычисления с объектами в примерах и задачах [Текст] : [учеб. пособие для вузов] / В. Э. Вольфенгаген ; НОУ Ин-т Актуального образования "ЮрИнфоР-МГУ, Каф. перспективных компьт. исслед. и информ. технологий. — 3-е изд., доп. и перераб. — М. : Ин-т "ЮрИнфоР-МГУ, 2008. — 384 с.

Дополнительная литература

Фонд кафедры:

1. Параллельное программирование многопоточных систем с разделяемой памятью [Текст] : учеб. пособие для вузов / А. Г. Тормасов. — М : Физматкнига, 2014. — 208 с.

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

<https://www.coursera.org/specializations/big-data-engineering> - специализация из 5 курсов, посвящённая тематике обработки больших данных.

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

Java

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

Успешное освоение курса требует напряжённой самостоятельной работы студента. В программе курса приведено минимально необходимое время для работы студента над темой. Самостоятельная работа включает в себя:

- проработку учебного материала (по учебной и научной литературе), подготовку ответов на вопросы, предназначенных для самостоятельного изучения, доказательство отдельных утверждений, свойств;
- выполнение двух индивидуальных домашних заданий.

Промежуточный контроль знаний проводится в виде письменных опросов по теории, а также студенту в ходе освоения курса необходимо выполнить две домашние индивидуальные работы с их последующей защитой:

ОЦЕНОЧНЫЕ МАТЕРИАЛЫ ПО ДИСЦИПЛИНЕ (МОДУЛЮ)

по направлению:	Прикладная математика и информатика
профиль подготовки:	Прикладная математика и информатика Физтех-школа Прикладной Математики и Информатики кафедра алгоритмов и технологий программирования
курс:	<u>1</u>
квалификация:	магистр
Семестр, формы промежуточной аттестации: 2 (весенний) - Экзамен	
Разработчик:	О.Н. Ивченко, старший преподаватель

1. Компетенции, формируемые в процессе изучения дисциплины

Код и наименование компетенции	Индикаторы достижения компетенции
УК-2 Способен управлять проектом на всех этапах его жизненного цикла	УК-2.1 Формулирует в рамках обозначенной проблемы цель, задачи, актуальность, значимость (научную, практическую, методическую и иную в зависимости от типа проекта), ожидаемые результаты и возможные сферы их применения
	УК-2.2 Способен прогнозировать результат деятельности и планировать последовательность шагов для достижения данного результата. Формирует план-график реализации проекта в целом и план контроля его выполнения
	УК-2.3 Способен организовать и координировать работу участников проекта, обеспечивать работу команды необходимыми ресурсами
	УК-2.4 Представляет публично результаты проекта (или отдельных его этапов) в форме отчетов, статей, выступлений на научно-практических конференциях, семинарах и т.п.
ОПК-6 Способен выбирать и (или) разрабатывать подходы к решению типовых и новых задач в области информатики и вычислительной техники, учитывая особенности и ограничения различных методов решения	ОПК-6.1 Способен анализировать задачу, планировать пути решения, предлагать и комбинировать способы решения
	ОПК-6.2 Способен разрабатывать и модернизировать программное и аппаратное обеспечение информационных и автоматизированных систем
	ОПК-6.3 Способен использовать исследовательские методы при решении новых задач, применяя знания из различных областей науки (техники)
	ОПК-6.4 Владеет аналитическими и вычислительными методами решения, понимает и учитывает на практике границы применимости получаемых решений
	ОПК-6.5 Способен самостоятельно приобретать, развивать и применять математические, естественнонаучные, социально-экономические и профессиональные знания для решения нестандартных задач, в том числе в новой или незнакомой среде и в междисциплинарном контексте

2. Показатели оценивания компетенций

В результате изучения дисциплины «Технологии разработки на языке Java» обучающийся должен:

знать:

- принцип исполнения программ на Java с использованием JVM;
- принцип работы сборки мусора в Java;
- типы данных языка Java;
- управление потоком выполнения в Java;
- основные классы и возможности стандартной библиотеки;
- правила работы с исключениями;
- принципы разработки параметризованных классов и методов (generics);
- внутреннее строение контейнеров стандартной библиотеки и временную сложность операций с ними;
- потоковая обработка данных при помощи Stream API;
- взаимодействие с реляционными СУБД с помощью JDBC API;
- принципы разработки многопоточного кода в Java и инструменты стандартной библиотеки;
- модель памяти Java;
- возможности Java Reflection API;
- применение аннотаций и обработка аннотаций на уровне Reflection API;
- системы непрерывной интеграции и их использование в среде Java.

уметь:

- реализовывать библиотеку общего назначения на языке Java по заданным интерфейсам;
- добавлять в приложение поддержку многопоточности, анализировать потокобезопасность реализации;
- покрывать код unit-тестами с использованием фреймворка JUnit, анализировать покрытие кода тестами;
- работать с распределенной системой контроля версий git;
- использовать средства code review на сервисе Github или GitLab.

владеть:

- навыками работы с объектами и потоками и кругозором в выборе архитектурного решения поставленной задачи.

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

1. Какая утилита используется для компиляции Java-программ:

- java
- javah
- javap
- javac
- jar

2. Что такое Java build tools. Опишите преимущества системы сборки Maven.

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

1. Чем отличается переопределение от перегрузки?
2. Чем отличается абстрактный класс от интерфейса? Укажите, как изменились эти отличия в Java 8+ .
3. Какими способами можно создать Stream в Java.
4. Чем различаются JRE, JVM и JDK?
5. О чем говорит ключевое слово final?
6. Какими значениями инициализируются переменные по умолчанию?
7. Для чего используется модификатор abstract?
8. Что такое маркерные интерфейсы? Чем они отличаются от обычных?
9. Для чего в Java используются статические блоки инициализации?
10. Может ли статический метод быть переопределён или перегружен?

Пример билета:

1. Чем различаются JRE, JVM и JDK?

2. Для чего в Java используются статические блоки инициализации?

Критерии оценивания

отлично

10 всесторонние, систематизированные, глубокие знания учебной программы дисциплины и умение уверенно применять их на практике при решении конкретных задач, свободное и правильное обоснование принятых решений;

9 систематизированные, глубокие знания учебной программы дисциплины и умение уверенно применять их на практике при решении конкретных задач, правильное обоснование принятых решений;

8 глубокие знания учебной программы дисциплины и умение применять их на практике при решении конкретных задач, правильное обоснование принятых решений;

хорошо

7 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;

6 знает материал, грамотно излагает его, умеет применять полученные знания на практике, но допускает в ответе или в решении задач некоторые неточности;

5 знает основной материал, грамотно излагает его, умеет применять полученные знания на практике, но допускает в ответе или в решении задач неточности;

удовлетворительно

4 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;

3 характер знаний достаточен для дальнейшего обучения и может применять полученные знания по образцу в стандартной ситуации;

неудовлетворительно

2 не знает большей части основного содержания учебной программы дисциплины, допускает грубые ошибки в формулировках основных понятий дисциплины и не умеет правильно использовать полученные знания при решении типовых практических задач.

1 не знает формулировок основных понятий дисциплины и не умеет использовать полученные знания при решении типовых практических задач.

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

Дифференцированный зачет проводится в письменной форме. Во время проведения зачета обучающиеся могут пользоваться своими конспектами.