Министерство науки и высшего образования Российской Федерации

Муромский институт (филиал)

федерального государственного бюджетного образовательного учреждения высшего образования

«Владимирский государственный университет имени Александра Григорьевича и Николая Григорьевича Столетовых»

(МИ ВлГУ)

Кафедра *ПИн*

«УТВЕРЖДАЮ»
Заместитель директора по УР
_____Д.Е. Андрианов
19.05.2026

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

Технологии платформы Java

Направление подготовки *09.04.04 Программная инженерия*

Профиль подготовки *Технологии разработки интеллектуальных систем*

Семестр	Трудоемкость, час./зач. ед.	Лекции, час.	Практические занятия, час.	Лабораторные работы, час.	Консультация, час.	Контроль, час.	Всего (контактная работа), час.	СРС, час.	Форма промежуточного контроля (экз., зач., зач. с оц.)
1	**144 / 4**	**32**		**32**	**3,2**	**0,25**	**67,45**	**76,55**	**Зач. с оц.**
Итого	**144 / 4**	**32**		**32**	**3,2**	**0,25**	**67,45**	**76,55**	

Муром, 2026 г.

1. Цель освоения дисциплины

Целью дисциплины является формирование у студентов углублённых знаний и практических навыков работы с современными технологиями платформы Java, включая низкоуровневую оптимизацию и инструментарий JVM (байт‑код, сборщики мусора), многопоточное и параллельное программирование (модель памяти JMM, виртуальные потоки, потокобезопасные коллекции, Executors), а также асинхронные и реактивные подходы (Netty, Spring WebFlux). Задачи дисциплины — научить студентов анализировать производительность кода, проектировать масштабируемые и отказоустойчивые системы с эффективным управлением ресурсами, а также применять современные парадигмы для создания высоконагруженных веб‑приложений.

2. Место дисциплины в структуре ОПОП ВО

Дисциплина базируется на знаниях принципов объектно-ориентированного программирования, полученных на бакалавриате.

3. Планируемые результаты обучения по дисциплине

Планируемые результаты обучения по дисциплине, соотнесенные с планируемыми результатами освоения ОПОП (компетенциями и индикаторами достижения компетенций)

Формируемые компетенции (код, содержание компетенции)	Планируемые результаты обучения по дисциплине, в соответствии с индикатором достижения компетенции		Наименование оценочного средства
	Индикатор достижения компетенции	Результаты обучения по дисциплине	
ПК-2 Владение методами программной реализации распределенных информационных систем	ПК-2.1 Реализует методы и программные интерфейсы взаимодействия с внешними программными компонентами	Принципы работы асинхронных и реактивных API для интеграции с внешними системами (ПК-2.1) Механизмы многопоточного и многопроцессного взаимодействия (ПК-2.1) Разрабатывать клиент-серверные приложения с использованием неблокирующего ввода-вывода (ПК-2.1) Настраивать пулы потоков и обрабатывать асинхронные вызовы внешних компонентов через реактивные стримы (ПК-2.1) Инструментами для тестирования производительности и отладки взаимодействия с внешними компонентами (ПК-2.1) Техниками реализации кастомных протоколов обмена данными с помощью Netty и байт-код инструментации (ПК-2.1)	Вопросы к опросу

4. Структура и содержание дисциплины

Общая трудоемкость дисциплины составляет 4 зачетные единицы, 144 часа.

4.1. Форма обучения: очная

Уровень базового образования: высшее.
Срок обучения 2г.

4.1.1. Структура дисциплины

№ п\п	Раздел (тема) дисциплины	Семестр	Контактная работа обучающихся с педагогическим работником							Самостоятельная работа	Форма текущего контроля успеваемости (по неделям семестра), форма промежуточной аттестации(по семестрам)
			Лекции	Практические занятия	Лабораторные работы	Контрольные работы	КП / КР	Консультация	Контроль		
1	JVM и инструменты производительности	1	10							17	Устный опрос
2	Многопоточность и параллелизм	1	12		16					35	Устный опрос
3	Реактивное и асинхронное программирование	1	10		16					24,55	Устный опрос
Всего за семестр		144	32		32			3,2	0,25	76,55	Зач. с оц.
Итого		144	32		32			3,2	0,25	76,55	

4.1.2. Содержание дисциплины

4.1.2.1. Перечень лекций

Семестр 1

Раздел 1. JVM и инструменты производительности

Лекция 1.

Инструменты для преобразования контейнеров, unsafe, jmh (2 часа).

Лекция 2.

Аннотации (2 часа).

Лекция 3.

Сборщик мусора (2 часа).

Лекция 4.

Байт код, class-loader, инструментация, asm (2 часа).

Лекция 5.

Типы ссылок. Кэширование (2 часа).

Раздел 2. Многопоточность и параллелизм

Лекция 6.

Thread (2 часа).

Лекция 7.

JMM (2 часа).

Лекция 8.

Executors (2 часа).

Лекция 9.

Потокобезопасные коллекции (2 часа).

Лекция 10.

Виртуальные потоки (2 часа).

Лекция 11.

Многопроцессные приложения (2 часа).

Раздел 3. Реактивное и асинхронное программирование

Лекция 12.

Функциональное программирование в Java (2 часа).

Лекция 13.

Web сервер (2 часа).

Лекция 14.

Asynchronous Web applications (2 часа).

Лекция 15.

Netty (2 часа).

Лекция 16.

Реактивное программирование. Spring Webflux (2 часа).

4.1.2.2. Перечень практических занятий

Не планируется.

4.1.2.3. Перечень лабораторных работ

Семестр 1

Раздел 2. Многопоточность и параллелизм

Лабораторная 1.

Thread. Executors (4 часа).

Лабораторная 2.

Потокобезопасные коллекции (4 часа).

Лабораторная 3.

Виртуальные потоки (4 часа).

Лабораторная 4.

Многопроцессные приложения (4 часа).

Раздел 3. Реактивное и асинхронное программирование

Лабораторная 5.

Web сервер (4 часа).

Лабораторная 6.

Asynchronous Web applications (4 часа).

Лабораторная 7.

Netty (4 часа).

Лабораторная 8.

Реактивное программирование. Spring Webflux (4 часа).

4.1.2.4. Перечень тем и учебно-методическое обеспечение самостоятельной работы

Перечень тем, вынесенных на самостоятельное изучение:

1. Алгоритмы сборки мусора.
2. Анализ байт-кода ASM.

3. JMH и микробенчмаркинг.
4. JMM и happens‑before.
5. Пулы потоков Executors.
6. Виртуальные потоки Project Loom.
7. Многопроцессное взаимодействие.
8. Реактивные потоки и Publisher.
9. Netty: EventLoop и каналы.
10. Spring WebFlux и маршрутизация.

Для самостоятельной работы используются методические указания по освоению дисциплины и издания из списка приведенной ниже основной и дополнительной литературы.

4.1.2.5. Перечень тем контрольных работ, рефератов, ТР, РГР, РПР

Не планируется.

4.1.2.6. Примерный перечень тем курсовых работ (проектов)

Не планируется.

5. Образовательные технологии

В процессе изучения дисциплины применяется контактная технология преподавания (за исключением самостоятельно изучаемых студентами вопросов). При проведении лабораторных работ применяется имитационный или симуляционный подход. Шаги решения задач студентам демонстрируются при помощи мультимедийной техники. В дальнейшем студенты самостоятельно решают аналогичные задания.

6. Оценочные средства для текущего контроля успеваемости, промежуточной аттестации по итогам освоения дисциплины.

Фонды оценочных материалов (средств) приведены в приложении.

7. Учебно-методическое и информационное обеспечение дисциплины.

7.1. Основная учебно-методическая литература по дисциплине

1. Блох, Дж. Java. Эффективное программирование / Дж. Блох ; перевод В. Стрельцов ; под редакцией Р. Усманова. — 3-е изд. — Саратов : Профобразование, 2024. — 310 c. — ISBN 978-5-4488-0127-3. — Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. — URL: https://www.iprbookshop.ru/145890.html (дата обращения: 18.05.2026). — Режим доступа: для авторизир. пользователей - https://www.iprbookshop.ru/145890.html

2. Свистунов, А. Н. Построение распределенных систем на Java : учебное пособие / А. Н. Свистунов. — 4-е изд. — Москва : Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2025. — 316 c. — ISBN 978-5-4497-0940-0. — Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. — URL: https://www.iprbookshop.ru/146380.html (дата обращения: 18.05.2026). — Режим доступа: для авторизир. пользователей - https://www.iprbookshop.ru/146380.html

3. Вязовик, Н. А. Программирование на Java : учебное пособие / Н. А. Вязовик. — 4-е изд. — Москва : Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2025. — 601 c. — ISBN 978-5-4497-0852-6. — Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. — URL: https://www.iprbookshop.ru/146383.html (дата обращения: 18.05.2026). — Режим доступа: для авторизир. пользователей - https://www.iprbookshop.ru/146383.html

7.2. Дополнительная учебно-методическая литература по дисциплине

1. Мухамедзянов, Р. Р. JAVA. Серверные приложения / Р. Р. Мухамедзянов. — Москва : СОЛОН-Р, 2021. — 336 c. — ISBN 5-93455-134-5. — Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. — URL: https://www.iprbookshop.ru/141949.html

(дата обращения: 18.05.2026). — Режим доступа: для авторизир. пользователей - https://www.iprbookshop.ru/141949.html

2. Ермаков, А. В. Разработка приложений на языке JAVA : учебное пособие / А. В. Ермаков, М. С. Королёв, А. К. Кузьмин. — Саратов : Саратовский государственный технический университет имени Ю.А. Гагарина, ЭБС АСВ, 2024. — 128 с. — ISBN 978-5-7433-3635-7. — Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. — URL: https://www.iprbookshop.ru/150088.html (дата обращения: 18.05.2026). — Режим доступа: для авторизир. пользователей. - DOI: https://doi.org/10.23682/150088 - https://www.iprbookshop.ru/150088.html

7.3. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине, включая перечень программного обеспечения и информационных справочных систем

В образовательном процессе используются информационные технологии, реализованные на основе информационно-образовательного портала института (www.mivlgu.ru/iop), и инфокоммуникационной сети института:

- предоставление учебно-методических материалов в электронном виде;
- взаимодействие участников образовательного процесса через локальную сеть института и Интернет;
- предоставление сведений о результатах учебной деятельности в электронном личном кабинете обучающегося.

Информационные справочные системы:

Электронная библиотека ВлГУ - http://e.lib.vlsu.ru/

электронная библиотечная системы "IPRBooks" (http://www.iprbookshop.ru/);

Программное обеспечение:

IntelliJ IDEA (Apache License 2.0)

7.4. Перечень ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины

iprbookshop.ru
e.lib.vlsu.ru
mivlgu.ru/iop

8. Материально-техническое обеспечение дисциплины

Лаборатория программирования и баз данных

12 шт. компьютеров Intel Core i5-10150 3,70 GHz/ 16Gb(DDr4)/ SSD-150Gb / Haff 23,8’; проектор ACER P1100 DLP Projector EMEA; экран проекционный настенный DRAPPER Apex STAR; маршрутизатор Gigabit Switch TEG-S16S; плоттер HP Design Jet T610. Маркерная доска. Доступ к сети Интернет.

9. Методические указания по освоению дисциплины

Для успешного освоения теоретического материала обучающийся: знакомится со списком рекомендуемой основной и дополнительной литературы; уточняет у преподавателя, каким дополнительным пособиям следует отдать предпочтение; ведет конспект лекций и прорабатывает лекционный материал, пользуясь как конспектом, так и учебными пособиями.

До выполнения лабораторных работ обучающийся изучает соответствующий раздел теории. Перед занятием студент знакомится с описанием заданий для выполнения работы, внимательно изучает содержание и порядок проведения лабораторной работы. Лабораторная работа проводятся в компьютерном классе. Обучающиеся выполняют индивидуальную задачу компьютерного моделирования в соответствии с заданием на лабораторную работу. Полученные результаты исследований сводятся в отчет и защищаются по традиционной методике в классе на следующем лабораторном занятии. Необходимый теоретический

материал, индивидуальное задание, шаги выполнения лабораторной работы и требование к отчету приведены в методических указаниях, размещенных на информационно-образовательном портале института.

Самостоятельная работа оказывает важное влияние на формирование личности будущего специалиста, она планируется обучающимся самостоятельно. Каждый обучающийся самостоятельно определяет режим своей работы и меру труда, затрачиваемого на овладение учебным содержанием дисциплины. Он выполняет внеаудиторную работу и изучение разделов, выносимых на самостоятельную работу, по личному индивидуальному плану, в зависимости от его подготовки, времени и других условий.

Форма заключительного контроля при промежуточной аттестации – зачет с оценкой. Для проведения промежуточной аттестации по дисциплине разработаны фонд оценочных средств и балльно-рейтинговая система оценки учебной деятельности студентов. Оценка по дисциплине выставляется в информационной системе и носит интегрированный характер, учитывающий результаты оценивания участия студентов в аудиторных занятиях, качества и своевременности выполнения заданий в ходе изучения дисциплины и промежуточной аттестации.

Программа составлена в соответствии с требованиями ФГОС ВО по направлению *09.04.04 Программная инженерия* и профилю подготовки *Технологии разработки интеллектуальных систем*
Рабочую программу составил *к.т.н., доцент Кульков Я.Ю.*_____

Программа рассмотрена и одобрена на заседании кафедры *ПИн*

протокол № 28 от 05.05.2026 года.
Заведующий кафедрой *ПИн* _____*Кульков Я.Ю.*
(Подпись)

Рабочая программа рассмотрена и одобрена на заседании учебно-методической комиссии факультета

протокол № 9 от 12.05.2026 года.
Председатель комиссии ФИТР _____ *Кутарова Е.И.*
(Подпись) (Ф.И.О.)

Приложение

Фонд оценочных материалов (средств) по дисциплине
Технологии платформы Java

1. Оценочные материалы для проведения текущего контроля успеваемости по дисциплине

1. Какой инструмент используется для написания микробенчмарков в Java?
- JMH
- JVisualVM
- JMC
- Async Profiler

2. Какая аннотация в Java указывает на функциональный интерфейс?
- @FunctionalInterface
- @Override
- @SuppressWarnings
- @SafeVarargs

3. Какой сборщик мусора в Java 17+ является стандартным для серверных приложений с низкой паузой?
- G1GC
- Parallel GC
- Serial GC
- ZGC

4. Какая библиотека позволяет генерировать и модифицировать байт-код Java во время выполнения?
- ASM
- Javassist
- ByteBuddy
- Все перечисленные

5. Какой тип ссылки используется в реализации кэша, чтобы объекты удалялись только при нехватке памяти, но не при каждой сборке мусора?
- SoftReference
- WeakReference
- PhantomReference
- StrongReference

6. Какое отношение в Java Memory Model определяет гарантии видимости изменений между потоками?
- happens-before
- synchronized
- volatile
- happens-after

7. Какой фабричный метод Executors создаёт пул потоков, который переиспользует ранее созданные потоки и добавляет новые по мере необходимости?
- newCachedThreadPool()
- newFixedThreadPool()
- newSingleThreadExecutor()
- newWorkStealingPool()

8. Что такое виртуальные потоки в Java?

- Легковесные потоки, реализованные на уровне JDK
- Потоки, управляемые операционной системой
- Потоки с приоритетом выше обычных
- Потоки, выполняющиеся только на одном ядре

9. Какой интерфейс в Netty представляет собой цепочку обработчиков для входящих и исходящих данных?
- ChannelPipeline
- ChannelHandler
- EventLoopGroup
- ByteBuf

10. Какой тип издателя в Project Reactor представляет асинхронную последовательность из 0 или 1 элемента?
- Mono
- Flux
- Publisher
- Processor

Общее распределение баллов текущего контроля по видам учебных работ для студентов

Рейтинг-контроль 1	устный опрос	30
Рейтинг-контроль 2	устный опрос	30
Рейтинг-контроль 3	устный опрос	30
Посещение занятий студентом		0
Дополнительные баллы (бонусы)		0
Выполнение семестрового плана самостоятельной работы	отчеты	10

2. Промежуточная аттестация по дисциплине

Перечень вопросов к экзамену / зачету / зачету с оценкой.
Перечень практических задач / заданий к экзамену / зачету / зачету с оценкой (при наличии)

Вопросы для устного опроса:

1. Объясните, как работает класс Unsafe в Java. Какие возможности он предоставляет и почему его использование считается опасным?
2. Опишите основные виды ссылок в Java (Strong, Soft, Weak, Phantom). Приведите примеры практического применения каждого типа, включая кэширование.
3. Как работает инструментация байт-кода с помощью библиотеки ASM? Назовите типичные сценарии использования в реальных проектах.
4. Что такое Java Memory Model (JMM)? Объясните отношения happens-before на примере ключевых слов volatile и synchronized.
5. Сравните пулы потоков, создаваемые методами newFixedThreadPool(), newCachedThreadPool() и newSingleThreadExecutor(). В каких случаях каждый из них предпочтительнее?
6. Что такое виртуальные потоки (Project Loom)? Чем они отличаются от классических потоков операционной системы? Приведите пример эффективного использования.
7. Какие потокобезопасные коллекции входят в пакет java.util.concurrent? Объясните принцип работы ConcurrentHashMap в сравнении с Hashtable и Collections.synchronizedMap().

8. Объясните разницу между асинхронным сервлетом (Asynchronous Web Applications) и реактивным программированием на основе Netty. Когда какой подход уместен?

9. Опишите архитектуру Netty: основные компоненты (EventLoop, ChannelPipeline, ChannelHandler). Как Netty достигает высокой производительности при обработке большого числа соединений?

10. Что такое реактивное программирование в контексте Spring WebFlux? Сравните подходы на основе Mono/Flux с традиционным блокирующим Spring MVC.

Методические материалы, характеризующие процедуры оценивания

Промежуточная аттестация проводится в форме устного экзамена: каждый билет включает два теоретических вопроса и одно практическое задание (анализ или проектирование архитектуры). На подготовку отводится до 30 минут, после чего студент устно отвечает на вопросы билета, при необходимости преподаватель задаёт уточняющие вопросы. Оценка выставляется с учётом полноты ответа, корректности решения практической задачи, владения терминологией и аргументации; семестровый рейтинг может учитываться при спорной оценке.

Максимальная сумма баллов, набираемая студентом по дисциплине равна 100.

Оценка в баллах	Оценка по шкале	Обоснование	***Уровень сформированности компетенций***
Более 80	«Отлично»	Содержание курса освоено полностью, без пробелов, необходимые практические навыки работы с освоенным материалом сформированы, все предусмотренные программой обучения учебные задания выполнены, качество их выполнения оценено числом баллов, близким к максимальному	***Высокий уровень***
66-80	«Хорошо»	Содержание курса освоено полностью, без пробелов, некоторые практические навыки работы с освоенным материалом сформированы недостаточно, все предусмотренные программой обучения учебные задания выполнены, качество выполнения ни одного из них не оценено минимальным числом баллов, некоторые виды заданий выполнены с ошибками	***Продвинутый уровень***

50-65	«Удовлетворительно»	Содержание курса освоено частично, но пробелы не носят существенного характера, необходимые практические навыки работы с освоенным материалом в основном сформированы, большинство предусмотренных программой обучения учебных заданий выполнено, некоторые из выполненных заданий, возможно, содержат ошибки	***Пороговый уровень***
Менее 50	«Неудовлетворительно»	Содержание курса не освоено, необходимые практические навыки работы не сформированы, выполненные учебные задания содержат грубые ошибки	***Компетенции не сформированы***

3. Задания в тестовой форме по дисциплине

Примеры заданий:
ПК-2.1

1. Какой инструмент используется для написания микробенчмарков в Java?
- JMH
- JVisualVM
- JMC
- Async Profiler

2. Какая аннотация в Java указывает на функциональный интерфейс?
- @FunctionalInterface
- @Override
- @SuppressWarnings
- @SafeVarargs

3. Какой сборщик мусора в Java 17+ является стандартным для серверных приложений с низкой паузой?
- G1GC
- Parallel GC
- Serial GC
- ZGC

4. Какая библиотека позволяет генерировать и модифицировать байт-код Java во время выполнения?
- ASM
- Javassist
- ByteBuddy
- Все перечисленные

5. Какой тип ссылки используется в реализации кэша, чтобы объекты удалялись только при нехватке памяти, но не при каждой сборке мусора?
- SoftReference
- WeakReference
- PhantomReference
- StrongReference

6. Какое отношение в Java Memory Model определяет гарантии видимости изменений между потоками?

- happens-before
- synchronized
- volatile
- happens-after

7. Какой фабричный метод Executors создаёт пул потоков, который переиспользует ранее созданные потоки и добавляет новые по мере необходимости?

- newCachedThreadPool()
- newFixedThreadPool()
- newSingleThreadExecutor()
- newWorkStealingPool()

8. Что такое виртуальные потоки в Java?

- Легковесные потоки, реализованные на уровне JDK
- Потоки, управляемые операционной системой
- Потоки с приоритетом выше обычных
- Потоки, выполняющиеся только на одном ядре

9. Какой интерфейс в Netty представляет собой цепочку обработчиков для входящих и исходящих данных?

- ChannelPipeline
- ChannelHandler
- EventLoopGroup
- ByteBuf

10. Какой тип издателя в Project Reactor представляет асинхронную последовательность из 0 или 1 элемента?

- Mono
- Flux
- Publisher
- Processor

Полный перечень тестовых заданий с указанием правильных ответов, размещен в банке вопросов на информационно-образовательном портале института по ссылке https://www.mivlgu.ru/iop/course/view.php?id=4569

Оценка рассчитывается как процент правильно выполненных тестовых заданий из их общего числа.