

Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Баламирзоев Назим Лиодинович
Должность: Ректор
Дата подписания: 06.04.2026 13:53:03
Уникальный программный ключ:
5cf0d6f89e80f49a334f6a4ba58e91f3326b9926

Приложение А

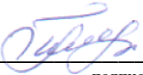
(обязательное к рабочей программе дисциплины)

Министерство науки и высшего образования Российской Федерации
ФГБОУ ВО «Дагестанский государственный технический университет»


ФОНД ОЦЕНОЧНЫХ СРЕДСТВ

по дисциплине «Программирование на языке Java»

Уровень образования	<u>специалитет</u> (бакалавриат/магистратура/специалитет)
Специальность	<u>10.05.03 Информационная безопасность автоматизированных систем</u> (код, наименование специальности)
Специализация	<u>Безопасность открытых информационных систем</u> (наименование)

Разработчик  Качаева Г.И.
подпись (ФИО уч. степень, уч. звание)

Фонд оценочных средств обсужден на заседании кафедры ИБиПИ «15» октября 2025г., протокол № 2

Зав. кафедрой  Качаева Г.И.
подпись (ФИО уч. степень, уч. звание)

г. Махачкала 2025

СОДЕРЖАНИЕ

1. Область применения, цели и задачи фонда оценочных средств	3
2. Описание показателей и критериев оценивания компетенций, формируемых в процессе освоения дисциплины (модуля)	3
2.1. Перечень компетенций с указанием этапов их формирования в процессе освоения ОПОП	4
2.1.2. Этапы формирования компетенций	6
2.2. Показатели уровней сформированности компетенций на этапах их формирования, описание шкал оценивания	7
2.2.1. Показатели уровней сформированности компетенций на этапах их формирования	7
2.2.2. Описание шкал оценивания	9
3. Типовые контрольные задания, иные материалы и методические рекомендации, необходимые для оценки сформированности компетенций в процессе освоения ОПОП	10
3.1. Задания и вопросы для входного контроля	10
3.2. Оценочные средства и критерии сформированности компетенций	10
3.2.1. Аттестационная контрольная работа №1	10
3.2.2. Аттестационная контрольная работа №2	10
3.2.3. Аттестационная контрольная работа №3	11
3.2.4. Список вопросов к зачету	11
3.3. Вопросы по остаточным знаниям	12

1. Область применения, цели и задачи фонда оценочных средств

Фонд оценочных средств (ФОС) является неотъемлемой частью рабочей программы дисциплины «Программирование на языке Java» и предназначен для контроля и оценки образовательных достижений обучающихся (в т.ч. по самостоятельной работе студентов, далее – СРС), освоивших программу данной дисциплины.

Целью фонда оценочных средств является установление соответствия уровня подготовки обучающихся требованиям ФГОС ВО по специальности 10.05.03 Информационная безопасность автоматизированных систем.

Рабочей программой дисциплины «Программирование на языке Java» предусмотрено формирование следующих компетенций:

ОПК-2 Способен применять информационно-коммуникационные технологии, программные средства системного и прикладного назначения, в том числе отечественного производства, для решения задач профессиональной деятельности

2. Описание показателей и критериев оценивания компетенций, формируемых в процессе освоения дисциплины (модуля)

Описание показателей и критериев оценивания компетенций, формируемых в процессе освоения дисциплины (модуля), и используемые оценочные средства приведены в таблице 1.

Перечень оценочных средств, рекомендуемых для заполнения таблицы 1 (в ФОС не приводится, используется только для заполнения таблицы)

- Эссе
- Устный опрос
- Вопросы для проведения зачета

2.1. Перечень компетенций с указанием этапов их формирования в процессе освоения ОПОП

Таблица 1

Код и наименование формируемой компетенции	Код и наименование индикатора достижения формируемой компетенции	Критерии оценивания	Наименование контролируемых разделов и тем ¹
<p>ОПК-2 Способен применять информационно-коммуникационные технологии, программные средства системного и прикладного назначения, в том числе отечественного производства, для решения задач профессиональной деятельности</p>	<p>ОПК-2.1.3 знает типовые структуры и принципы организации компьютерных сетей назначение, функции и обобщённую структуру операционных систем назначение и основные компоненты систем баз данных</p>	<p>Знать: Типовые топологии компьютерных сетей, модели OSI и TCP/IP, основные протоколы стека TCP/IP. Назначение и функции операционных систем (управление процессами, памятью, вводом-выводом, файловыми системами). Принципы организации многозадачности и синхронизации. Основные компоненты систем управления базами данных (СУБД): ядро, процессор запросов, подсистема хранения, транзакции. Модели данных (иерархическая, сетевая, реляционная, объектная). Роль сетевых технологий, ОС и СУБД в разработке приложений на Java (клиент-серверные приложения, JDBC, многопоточность). Уметь: Идентифицировать уровни модели OSI при передаче данных в сети. Использовать базовые сетевые утилиты для диагностики. Различать типы операционных систем и их особенности при разработке кроссплатформенных приложений. Проектировать простые схемы баз данных для Java-приложений. Понимать взаимодействие Java-программы с СУБД через JDBC. Владеть: Навыками настройки сетевых параметров в приложениях Java (сокеты, URLConnection). Навыками работы с файловыми системами и процессами через Java API. Навыками подключения к базам данных, выполнения запросов и обработки результатов с использованием JDBC.</p>	<p>№№ 1-17</p>
	<p>ОПК-2.2.2 умеет составлять SQL запросы и осуществлять</p>	<p>Знать: Синтаксис основных команд SQL: SELECT, INSERT, UPDATE, DELETE. Типы JOIN, агрегатные функции (COUNT, SUM, AVG, MIN, MAX), группировка GROUP BY, фильтрация</p>	<p>№№ 1-17</p>

¹ Наименования разделов и тем должен соответствовать рабочей программе дисциплины.

	удалённый доступ к базам данных	<p>HAVING. Подзапросы, индексы, транзакции. Принципы удалённого доступа к базам данных: JDBC API (DriverManager, Connection, Statement, PreparedStatement, ResultSet), пулы соединений, обработка исключений SQLException. Основы работы с системами управления базами данных (MySQL, PostgreSQL, Oracle).</p> <p>Уметь: Писать SQL-запросы средней сложности для выборки, вставки, обновления и удаления данных. Использовать параметризованные запросы для защиты от SQL-инъекций. Устанавливать соединение с удалённой базой данных из Java-приложения, выполнять запросы и обрабатывать результаты. Работать с транзакциями (commit, rollback, setAutoCommit). Обрабатывать ошибки при работе с БД.</p> <p>Владеть: Навыками разработки Java-приложений, взаимодействующих с базами данных через JDBC. Навыками отладки SQL-запросов и анализа производительности. Умением использовать современные практики доступа к данным (DAO, репозитории). Навыками настройки подключения к удалённым базам данных (URL, драйверы, параметры аутентификации).</p>	
--	---------------------------------	--	--

2.1.2. Этапы формирования компетенций

Сформированность компетенций по дисциплине «Программирование на языке Java» определяется на следующих этапах:

1. **Этап текущих аттестаций** (Для проведения текущих аттестаций могут быть использованы оценочные средства, указанные в разделе 2)

2. **Этап промежуточных аттестаций** (Для проведения промежуточной аттестации могут быть использованы другие оценочные средства)

Таблица 2

Код и наименование формируемой компетенции	Код и наименование индикатора достижения формируемой компетенции	Этапы формирования компетенции					Этап промежуточной аттестации	
		Этап текущих аттестаций				18-20 неделя		
		1-5 неделя	6-10 неделя	11-15 неделя	1-17 неделя			
		Текущая аттестация №1	Текущая аттестация №2	Текущая аттестация №3	СРС			КР/КП
1	2	3	4	5	6	7		
ОПК-2 Способен применять информационно-коммуникационные технологии, программные средства системного и прикладного назначения, в том числе отечественного производства, для решения задач профессиональной деятельности	ОПК-2.1.3 знает типовые структуры и принципы организации компьютерных сетей назначение, функции и обобщённую структуру операционных систем назначение и основные компоненты систем баз данных	Контрольная работа №1	Контрольная работа №2	Контрольная работа №3			Вопросы для проведения зачета	
	ОПК-2.2.2 умеет составлять SQL запросы и осуществлять удалённый доступ к базам данных	Контрольная работа №1	Контрольная работа №2	Контрольная работа №3			Вопросы для проведения зачета	

СРС – самостоятельная работа студентов;

КР – курсовая работа;

КП – курсовой проект.

2.2. Показатели уровней сформированности компетенций на этапах их формирования, описание шкал оценивания

2.2.1. Показатели уровней сформированности компетенций на этапах их формирования

Результатом освоения дисциплины «Программирование на языке Java» является установление одного из уровней сформированности компетенций: высокий, повышенный, базовый, низкий.

Таблица 3

Уровень	Универсальные компетенции	Общепрофессиональные/ профессиональные компетенции
Высокий (оценка «отлично», «зачтено»)	Сформированы четкие системные знания и представления по дисциплине. Ответы на вопросы оценочных средств полные и верные. Даны развернутые ответы на дополнительные вопросы. Обучающимся продемонстрирован высокий уровень освоения компетенции	Обучающимся усвоена взаимосвязь основных понятий дисциплины, в том числе для решения профессиональных задач. Ответы на вопросы оценочных средств самостоятельны, исчерпывающие, содержание вопроса/задания оценочного средства раскрыто полно, профессионально, грамотно. Даны ответы на дополнительные вопросы. Обучающимся продемонстрирован высокий уровень освоения компетенции
Повышенный (оценка «хорошо», «зачтено»)	Знания и представления по дисциплине сформированы на повышенном уровне. В ответах на вопросы/задания оценочных средств изложено понимание вопроса, дано достаточно подробное описание ответа, приведены и раскрыты в тезисной форме основные понятия. Ответ отражает полное знание материала, а также наличие, с незначительными пробелами, умений и навыков по изучаемой дисциплине. Допустимы единичные негрубые ошибки. Обучающимся продемонстрирован повышенный уровень освоения компетенции	Сформированы в целом системные знания и представления по дисциплине. Ответы на вопросы оценочных средств полные, грамотные. Продемонстрирован повышенный уровень владения практическими умениями и навыками. Допустимы единичные негрубые ошибки по ходу ответа, в применении умений и навыков
Базовый (оценка «удовлетворительно», «зачтено»)	Ответ отражает теоретические знания основного материала дисциплины в объеме, необходимом для дальнейшего освоения ОПОП. Обучающийся допускает неточности в ответе, но обладает необходимыми знаниями для их	Обучающийся владеет знаниями основного материал на базовом уровне. Ответы на вопросы оценочных средств неполные, допущены существенные ошибки. Продемонстрирован базовый уровень владения

Уровень	Универсальные компетенции	Общепрофессиональные/ профессиональные компетенции
	устранения. Обучающимся продемонстрирован базовый уровень освоения компетенции	практическими умениями и навыками, соответствующий минимально необходимому уровню для решения профессиональных задач
Низкий (оценка «неудовлетворительно», «не зачтено»)	Демонстрирует полное отсутствие теоретических знаний материала дисциплины, отсутствие практических умений и навыков	

Показатели уровней сформированности компетенций могут быть изменены, дополнены и адаптированы к конкретной рабочей программе дисциплины.

2.2.2. Описание шкал оценивания

В ФГБОУ ВО «ДГТУ» внедрена модульно-рейтинговая система оценки учебной деятельности студентов. В соответствии с этой системой применяются пятибалльная, двадцатибалльная и стобальная шкалы знаний, умений, навыков.

Шкалы оценивания			Критерии оценивания
пятибалльная	двадцатибалльная	стобальная	
«Отлично» - 5 баллов	«Отлично» - 18-20 баллов	«Отлично» - 85 – 100 баллов	Показывает высокий уровень сформированности компетенций, т.е.: <ul style="list-style-type: none"> - продемонстрирует глубокое и прочное усвоение материала; - исчерпывающе, четко, последовательно, грамотно и логически стройно излагает теоретический материал; - правильно формирует определения; - демонстрирует умения самостоятельной работы с нормативно-правовой литературой; - умеет делать выводы по излагаемому материалу.
«Хорошо» - 4 баллов	«Хорошо» - 15 - 17 баллов	«Хорошо» - 70 - 84 баллов	Показывает достаточный уровень сформированности компетенций, т.е.: <ul style="list-style-type: none"> - демонстрирует достаточно полное знание материала, основных теоретических положений; - достаточно последовательно, грамотно логически стройно излагает материал; - демонстрирует умения ориентироваться в нормальной литературе; - умеет делать достаточно обоснованные выводы по излагаемому материалу.
«Удовлетворительно» - 3 баллов	«Удовлетворительно» - 12 - 14 баллов	«Удовлетворительно» - 56 – 69 баллов	Показывает пороговый уровень сформированности компетенций, т.е.: <ul style="list-style-type: none"> - демонстрирует общее знание изучаемого материала; - испытывает серьезные затруднения при ответах на дополнительные вопросы; - знает основную рекомендуемую литературу; - умеет строить ответ в соответствии со структурой излагаемого материала.
«Неудовлетворительно» - 2 баллов	«Неудовлетворительно» - 1-11 баллов	«Неудовлетворительно» - 1-55 баллов	Ставится в случае: <ul style="list-style-type: none"> - незнания значительной части программного материала; - не владения понятийным аппаратом дисциплины; - допущения существенных ошибок при изложении учебного материала; - неумение строить ответ в соответствии со структурой излагаемого вопроса; - неумение делать выводы по излагаемому материалу.

3. Типовые контрольные задания, иные материалы и методические рекомендации, необходимые для оценки сформированности компетенций в процессе освоения ОПОП

3.1. Задания и вопросы для входного контроля

1. Опишите основные принципы объектно-ориентированного программирования (инкапсуляция, наследование, полиморфизм).
2. Чем примитивные типы отличаются от ссылочных типов в языках программирования?
3. Что такое массив? Как объявить и инициализировать массив в Python?
4. Объясните разницу между циклом for и while. Приведите примеры.
5. Что такое исключение? Как обрабатываются исключения в Python (try-except)?
6. Что такое класс и объект? Приведите пример определения класса в Python.
7. Что такое рекурсия? Напишите рекурсивную функцию для вычисления факториала.
8. Опишите структуру данных «стек». Какие операции с ней можно выполнять?
9. Что такое база данных? Для чего используются SQL-запросы?
10. Что такое клиент-серверная архитектура? Приведите пример взаимодействия клиента и сервера.

3.2. Оценочные средства и критерии сформированности компетенций

3.2.1. Аттестационная контрольная работа №1

1. Перечислите примитивные типы данных в Java. Каковы их размеры и диапазоны значений?
2. Что такое автоупаковка (autoboxing) и автораспаковка (unboxing)? Приведите пример.
3. Объясните разницу между операторами == и equals() при сравнении строк.
4. Как объявить и инициализировать одномерный и двумерный массив в Java?
5. Напишите фрагмент кода, использующий оператор switch для вывода названия дня недели по его номеру.
6. Какие виды циклов существуют в Java? Приведите пример использования цикла for-each для обхода массива.
7. Что такое приведение типов? Чем отличается явное приведение от неявного? Приведите примеры.
8. Как преобразовать строку в число (int, double) и наоборот?
9. Что такое метки (labels) в Java? Для чего они используются в циклах?
10. Напишите программу, которая вычисляет сумму элементов массива целых чисел.

3.2.2. Аттестационная контрольная работа №2

1. Опишите структуру класса в Java: поля, конструкторы, методы. Приведите пример.
2. Что такое интерфейс? Чем интерфейс отличается от абстрактного класса? (Java 8+)
3. Какие методы класса Object вы знаете? Для чего используются equals() и hashCode()?
4. Что такое обобщения (generics)? Напишите пример обобщённого класса Box<T>.
5. Как создать перечисление (enum) в Java? Можно ли добавлять поля и методы в enum?
6. Обработка исключений: ключевые слова try, catch, finally, throw, throws. Приведите пример с несколькими catch-блоками.
7. Какие классы для работы со строками существуют в Java (String, StringBuilder, StringBuffer)? В чём различия?
8. Что такое JavaDoc? Какие теги используются для документирования кода?
9. Напишите простой тест с использованием JUnit для проверки метода, складывающего два числа.
10. Что такое утверждения (assert) и как их включить при выполнении программы?

3.2.3. Аттестационная контрольная работа №3

1. Опишите основные шаги для подключения к базе данных из Java приложения с использованием JDBC.
2. Чем отличается Statement от PreparedStatement? Почему PreparedStatement предпочтительнее для защиты от SQL-инъекций?
3. Что такое Stream API? Приведите пример фильтрации и сбора элементов списка с помощью stream.
4. Для чего используется класс Optional? Как избежать NullPointerException с его помощью?
5. Как создать поток (thread) в Java? Опишите два способа: наследование от Thread и реализация Runnable.
6. Что такое синхронизация? Приведите пример использования ключевого слова synchronized для защиты общего ресурса.
7. Что такое Reflection API? Для каких задач оно применяется?
8. Что такое Dependency Injection? Назовите преимущества использования DI.
9. Напишите пример использования ExecutorService для пула потоков.
10. Как обрабатывать исключения в Stream API?

3.2.4. Список вопросов к зачету

1. Примитивные типы данных в Java: список, размеры, значения по умолчанию.
2. Преобразование типов: автоматическое и явное. Потеря данных при приведении.
3. Массивы в Java: объявление, инициализация, длина, копирование.
4. Операторы ветвления: if-else, switch. Примеры использования.
5. Циклы: for, while, do-while, for-each. Операторы break и continue.
6. Классы и объекты: определение класса, поля, методы, конструкторы.
7. Модификаторы доступа: private, default, protected, public.
8. Пакеты и импорт. Стандартные пакеты Java.
9. Наследование в Java: extends, super, переопределение методов.
10. Абстрактные классы и интерфейсы. Отличия, множественное наследование через интерфейсы.
11. Полиморфизм: связывание методов, виртуальные вызовы.
12. Класс Object: основные методы (toString, equals, hashCode, getClass).
13. Обобщения (generics): параметризованные классы и методы, ограничения.
14. Перечисления (enum): создание, методы, использование в switch.
15. Исключения: checked и unchecked exceptions, иерархия, обработка try-catch-finally, throws.
16. Строки: String, StringBuilder, StringBuffer. Основные операции.
17. Ввод-вывод: классы InputStream/OutputStream, Reader/Writer, работа с файлами.
18. Сериализация объектов: интерфейс Serializable, transient.
19. Коллекции: иерархия Collection и Map. List, Set, Queue. Основные реализации.
20. Сравнение объектов: Comparable и Comparator.
21. Многопоточность: создание потоков, состояния, приоритеты, демоны.
22. Синхронизация: synchronized, wait, notify, notifyAll. Блокировки.
23. Пакет java.util.concurrent: ExecutorService, Callable, Future, блокирующие очереди.
24. Stream API: создание стримов, промежуточные и терминальные операции, параллельные стримы.
25. Optional: создание, методы, использование для избежания null.
26. JDBC: драйверы, Connection, Statement, PreparedStatement, ResultSet.
27. Транзакции в JDBC: commit, rollback, savepoint, setAutoCommit.
28. Reflection API: получение информации о классе, создание экземпляров, вызов методов.
29. Аннотации: создание, встроенные аннотации, обработка аннотаций.
30. Dependency Injection: принцип, примеры внедрения зависимостей (конструктор, setter, интерфейс).

3.3. Вопросы по остаточным знаниям

1. Перечислите основные принципы ООП и объясните, как они реализованы в Java.
2. Чем интерфейс отличается от абстрактного класса в Java? (с учётом Java 8+)
3. Как работает сборка мусора в Java? Какие методы можно использовать для управления памятью?
4. Опишите различия между коллекциями ArrayList, LinkedList, HashSet и HashMap. Когда что применять?
5. Что такое многопоточность? Как синхронизировать доступ к общим данным в Java?
6. Для чего используется Stream API? Приведите пример конвейерной обработки данных.
7. Как выполнить подключение к базе данных из Java? Какие основные шаги JDBC?
8. Что такое обобщения (generics) и какие преимущества они дают?
9. Объясните механизм обработки исключений в Java. Чем checked исключения отличаются от unchecked?
10. Что такое рефлексия (Reflection) и где она применяется в реальных приложениях (фреймворки, тестирование)?

Зачеты и экзамены могут быть проведены в письменной форме, а также в письменной форме с устным дополнением ответа. Зачеты служат формой проверки качества выполнения студентами лабораторных работ, усвоения семестрового учебного материала по дисциплине (модулю), практических и семинарских занятий (при отсутствии экзамена по дисциплине).

По итогам зачета, соответствии с модульно – рейтинговой системой университета, выставляются баллы с последующим переходом по шкале баллы – оценки за зачет, выставляемый как по наименованию «зачтено», «не зачтено», так и дифференцированно т.е. с выставлением отметки по схеме – «отлично», «хорошо», «удовлетворительно» и «неудовлетворительно», определяемое решением Ученого совета университета и прописываемого в учебном плане.

Экзамен по дисциплине (модулю) служит для оценки работы студента в течении семестра (года, всего срока обучения и др.) и призван выявить уровень, качество и систематичность полученных им теоретических и практических знаний, приобретения навыков самостоятельной работы, развития творческого мышления, умения синтезировать полученные знания и применять их в решении практических задач. По итогам экзамена, в соответствии с модульно – рейтинговой системой университета выставляются баллы, с последующим переходом по шкале оценок на оценки: «отлично», «хорошо», «удовлетворительно» и «неудовлетворительно», свидетельствующие о приобретенных компетенциях или их отсутствии.

Критерии оценки уровня сформированности компетенций по результатам проведения зачета:

- оценка «зачтено»: обучающийся демонстрирует всестороннее, систематическое и глубокое знание материала, свободно выполняет задания, предусмотренные программой дисциплины, усвоивший основную и дополнительную литературу. Обучающийся выполняет задания, предусмотренные программой дисциплины, на уровне не ниже базового;

- оценка «не зачтено»: обучающийся демонстрирует незнание материала, не выполняет задания, предусмотренные программой дисциплины. Обучающийся не выполняет задания, предусмотренные программой дисциплины, на уровне ниже базового. Дальнейшее освоение ОПОП не возможно без дополнительного изучения материала и подготовки к зачету.

Критерии оценки уровня сформированности компетенций по результатам проведения дифференцированного зачёта (зачета с оценкой) / экзамена:

- оценка «**отлично**»: обучающийся дал полный, развернутый ответ на поставленный вопрос, проявил совокупность осознанных знаний об объекте, доказательно раскрыл основные положения темы. В ответе прослеживается четкая структура, логическая последовательность, отражающая сущность раскрываемых понятий, явлений. Обучающийся подкрепляет теоретический ответ практическими примерами. Ответ сформулирован научным языком, обоснована авторская позиция обучающегося. Могут быть допущены недочеты в определении понятий, исправленные студентом самостоятельно в процессе ответа или с помощью «наводящих» вопросов преподавателя. Обучающимся продемонстрирован высокий уровень владения компетенцией(-ями);

- оценка «**хорошо**»: обучающимся дан полный, развернутый ответ на поставленный вопрос, проявлено умение выделять существенные и несущественные признаки, причинно-следственные связи. Ответ четко структурирован, логичен, но есть недочеты в формулировании понятий, решении задач. При ответах на дополнительные вопросы допущены незначительные ошибки. Обучающимся продемонстрирован повышенный уровень владения компетенцией(-ями);

- оценка «**удовлетворительно**»: обучающимся дан неполный ответ на вопрос, логика и последовательность изложения имеют существенные нарушения. Допущены грубые ошибки при определении сущности раскрываемых понятий, явлений, нарушена логика ответа, не сделаны выводы. Речевое оформление требует коррекции. Обучающийся испытывает затруднение при ответе на дополнительные вопросы. Обучающимся продемонстрирован базовый уровень владения компетенцией(-ями);

- оценки «**неудовлетворительно**»: обучающийся испытывает значительные трудности в ответе на вопрос, допускает существенные ошибки, не владеет терминологией, не знает основных понятий, не может ответить на «наводящие» вопросы преподавателя. Обучающимся продемонстрирован низкий уровень владения компетенцией(-ями).