

Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Таскаев Сергей Валерьевич
Должность: Ректор
Дата подписания: 04.06.2026 09:21:33
Уникальный программный ключ:
891934b8c2cf7b6350cbe51cdda3096e877f61f7



МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное образовательное
учреждение высшего образования
«Челябинский государственный университет» (ФГБОУ ВО «ЧелГУ»)
Миасский филиал
Кафедра прикладной математики

Фонд оценочных средств по дисциплине «Программирование на Java»
по направлению подготовки 02.03.02 Фундаментальная информатика и информационные технологии, профиль
«Компьютерные науки» ФГБОУ ВО «ЧелГУ»

Версия документа - 1

стр. 1

Первый экземпляр _____

КОПИЯ № ____

Фонд оценочных средств для промежуточной аттестации

по дисциплине

Программирование на Java

Направление подготовки

02.03.02 Фундаментальная информатика и информационные технологии

Направленность (профиль)

Компьютерные науки

Присваиваемая квалификация
бакалавр

Форма обучения

очная

Миасс 2026 г.

**02.03.02 Фундаментальная информатика и информационные технологии,
Компьютерные науки, Программирование на Java, 2026, очная**

Фонд оценочных средств одобрен и рекомендован:

Проректор по учебной работе утверждено 27.02.26 А.А. Саламатов

Ученым советом Миасского филиала ФГБОУ ВО "ЧелГУ"

Протокол заседания № 8 от 24.02.2026

Председатель Ученого совета
Миасского филиала ФГБОУ ВО
"ЧелГУ"

согласовано

Т.В. Малькова

Заседанием кафедры прикладной математики

Протокол заседания № 6 от 30.01.2026

Заведующий кафедрой

согласовано

Е.В. Дутикова

Автор (составитель)

К.А. Лихачев

**Структура фонда оценочных средств для промежуточной аттестации по дисциплине
соответствует приказу ректора ФГБОУ ВО «ЧелГУ» от 27.09.2022 г. № 573-1 «Об
утверждении шаблонов документов».**



МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное образовательное
учреждение высшего образования
«Челябинский государственный университет» (ФГБОУ ВО «ЧелГУ»)
Миасский филиал
Кафедра прикладной математики

Фонд оценочных средств по дисциплине «Программирование на Java»
по направлению подготовки 02.03.02 Фундаментальная информатика и информационные технологии, профиль «Компьютерные науки» ФГБОУ ВО «ЧелГУ»

Версия документа - 1

стр. 3 из 29

Первый экземпляр _____

КОПИЯ № _____

Содержание

1. Паспорт фонда оценочных средств.....	4
2. Перечень формируемых компетенций.....	4
2.1. Компетенции, закреплённые за дисциплиной.....	4
3. Содержание оценочных средств по дисциплине.....	6
3.1 Виды оценочных средств.....	6
4. Порядок проведения и критерии оценивания промежуточной аттестации.....	25
4.1 Порядок проведения промежуточной аттестации.....	25
4.2. Критерии оценивания промежуточной аттестации по видам оценочных средств.....	32
4.3. Результаты промежуточной аттестации и уровни сформированности компетенций.....	35



МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное образовательное
учреждение высшего образования
«Челябинский государственный университет» (ФГБОУ ВО «ЧелГУ»)
Миасский филиал
Кафедра прикладной математики

Фонд оценочных средств по дисциплине «Программирование на Java»
по направлению подготовки 02.03.02 Фундаментальная информатика и информационные технологии, профиль «Компьютерные науки» ФГБОУ ВО «ЧелГУ»

Версия документа - 1

стр. 4 из 29

Первый экземпляр _____

КОПИЯ № _____

1. ПАСПОРТ ФОНДА ОЦЕНОЧНЫХ СРЕДСТВ

Направление подготовки: 02.03.02 Фундаментальная информатика и информационные технологии

Направленность (профиль): *Компьютерные науки*

Дисциплина: «Программирование на Java»

Семестры изучения: 6

Формы промежуточной аттестации: *зачет*

2. ПЕРЕЧЕНЬ ФОРМИРУЕМЫХ КОМПЕТЕНЦИЙ

2.1. Компетенции, закреплённые за дисциплиной

Изучение дисциплины «Программирование на Java» направлено на формирование следующих компетенций:

Коды компетенции (по ФГОС)	Содержание компетенций согласно ФГОС	Индикаторы достижения компетенций в соответствии с ОПОП	Перечень планируемых результатов обучения по дисциплине
1	2	3	4
ПК-2	Способен к разработке программного обеспечения, осуществлению интеграции программных модулей и компонент и проверке работоспособности программного обеспечения на основе международных и профессиональных стандартов в области информационных технологий	ПК-2.1 Демонстрирует знание основных принципов и технологий разработки программного обеспечения, методов и средств сборки модулей и компонент программного обеспечения; разработки процедур для развертывания программного обеспечения, методов и средств миграции и преобразования данных, методов создания пользовательских интерфейсов; средств программирования ПК-2.2 Демонстрирует умения разрабатывать программный код на языках программирования высокого и низкого уровня, осуществлять отладку программ, оформлять техническую документацию; использовать выбранную среду программирования для разработки процедур интеграции программных модулей, проводить оценку работоспособности программного обеспечения ПК-2.3 Имеет практический опыт разработки исходного кода, тестирования программного обеспечения, сборки модулей	<i>Знать</i> парадигмы объектно-ориентированного программирования и их особенности на языке Java, основы разработки программных интерфейсов на языке Java <i>Уметь</i> разрабатывать программные интерфейсы на языке Java, проектировать и разрабатывать абстрактные классы и суперклассы на языке Java <i>Владеть</i> навыками разработки программных интерфейсов на языке Java, разработки классов



МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное образовательное
учреждение высшего образования
«Челябинский государственный университет» (ФГБОУ ВО «ЧелГУ»)
Миасский филиал
Кафедра прикладной математики

Фонд оценочных средств по дисциплине «Программирование на Java»
по направлению подготовки 02.03.02 Фундаментальная информатика и информационные технологии, профиль «Компьютерные науки» ФГБОУ ВО «ЧелГУ»

Версия документа - 1

стр. 5 из 29

Первый экземпляр _____

КОПИЯ № _____

		и компонент программного обеспечения, разработки процедур для развертывания программного обеспечения, миграции и преобразования данных, создания программных интерфейсов	на языке Java, проектирования UML-диаграмм для классов на языке Java
--	--	--	--

3. СОДЕРЖАНИЕ ОЦЕНОЧНЫХ СРЕДСТВ ПО ДИСЦИПЛИНЕ

3.1 Виды оценочных средств

№ п/п	Контролируемые темы/разделы	Код компетенции/ планируемые результаты обучения	Наименование оценочного средства для текущего контроля	Наименование оценочного средства на промежуточной аттестации
1	Основы JAVA	ПК-2 <i>Знает</i> парадигмы объектно-ориентированного программирования и их особенности на языке Java, основы разработки программных интерфейсов на языке Java	Контрольная работа №1	Вопросы к зачету Типовые задания к зачету
2	Объектно ориентированный язык JAVA	ПК-2 <i>Умеет</i> разрабатывать программные интерфейсы на языке Java, проектировать и разрабатывать абстрактные классы и супер-классы на языке Java	Контрольная работа №2	Вопросы к экзамену Типовые задания к экзамену
3	Отличительные свойства языка	ПК-2 <i>Владеет</i> навыками разработки программных интерфейсов на языке Java, разработки классов на языке Java, проектирования UML-диаграмм для классов на языке Java	Контрольная работа №3	Вопросы к экзамену Типовые задания к экзамену

Типовые задания, контрольные работы, критерии и показатели оценивания в рамках текущего контроля представлены в рабочей программе по дисциплине. Полные комплекты оценочных средств и контрольно-измерительных материалов хранятся на кафедре и являются учебно-методическими материалами ограниченного (конфиденциального) пользования.



МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное образовательное
учреждение высшего образования
«Челябинский государственный университет» (ФГБОУ ВО «ЧелГУ»)
Миасский филиал
Кафедра прикладной математики

Фонд оценочных средств по дисциплине «Программирование на Java»
по направлению подготовки 02.03.02 Фундаментальная информатика и информационные технологии, профиль «Компьютерные науки» ФГБОУ ВО «ЧелГУ»

Версия документа - 1

стр. 6 из 29

Первый экземпляр _____

КОПИЯ № _____

3.2 Содержание оценочных средств для текущей аттестации

Тестовые задания по дисциплине «Программирование на Java»

Блок 1. Вопросы открытого типа (1–10)

№	Вопрос
1	Дайте определение объектно-ориентированного программирования в Java. Перечислите три основных принципа ООП и приведите краткий пример кода, демонстрирующий каждый из них.
2	В чём принципиальная разница между абстрактным классом (abstract class) и интерфейсом (interface) в Java? В каких архитектурных ситуациях целесообразнее использовать каждый из них?
3	Опишите механизм сборки мусора (Garbage Collection) в JVM. Как организована куча (Heap) и какие поколения объектов существуют?
4	Дайте сравнительную характеристику коллекций List, Set и Map. Приведите по одной конкретной реализации каждой из них и укажите их временную сложность базовых операций.
5	Что такое дженерики (Generics) в Java? Объясните механизм стирания типов (Type Erasure) и приведите пример использования ограниченного wildcard (<? extends T> или <? super T>).
6	Опишите назначение и синтаксис лямбда-выражений в Java 8. Что такое функциональный интерфейс? Приведите пример использования встроенного интерфейса java.util.function.Predicate.
7	Опишите архитектуру JDBC и основные этапы выполнения SQL-запроса из Java-приложения. Чем PreparedStatement отличается от Statement и почему его использование предпочтительнее?
8	В чём суть Stream API, появившегося в Java 8? Объясните разницу между промежуточными (intermediate) и терминальными (terminal) операциями, а также концепцию ленивых вычислений.
9	Опишите разницу между перегрузкой методов (Method Overloading) и переопределением методов (Method Overriding). Как JVM определяет, какой метод вызывать во время выполнения программы (динамическое связывание)?
10	Опишите систему обработки исключений в Java. В чём отличие проверяемых (checked) и непроверяемых (unchecked) исключений? Как работает конструкция try-with-resources и какие требования она предъявляет к объектам?

Блок 2. Вопросы закрытого типа (11–20)

№	Вопрос	Варианты ответов
11	Какое ключевое слово в Java запрещает наследование от класса и переопределение метода?	а) static; б) final; в) abstract; г) sealed
12	Какая коллекция сохраняет порядок вставки и допускает дубликаты элементов, обеспечивая	а) HashSet; б) LinkedList; в) ArrayList; г) TreeMap



МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное образовательное
учреждение высшего образования
«Челябинский государственный университет» (ФГБОУ ВО «ЧелГУ»)
Миасский филиал
Кафедра прикладной математики

Фонд оценочных средств по дисциплине «Программирование на Java»
по направлению подготовки 02.03.02 Фундаментальная информатика и информационные технологии, профиль «Компьютерные науки» ФГБОУ ВО «ЧелГУ»

Версия документа - 1

стр. 7 из 29

Первый экземпляр _____

КОПИЯ № _____

	доступ по индексу за $O(1)$?	
13	Что гарантирует аннотация <code>@Override</code> при компиляции Java-кода?	а) Метод будет выполнен в отдельном потоке; б) Компилятор проверит, действительно ли метод переопределяет метод родительского класса; в) Метод станет недоступным для внешнего кода; г) Метод будет автоматически помечен как <code>final</code>
14	Для корректной работы конструкции <code>try-with-resources</code> объект должен реализовывать интерфейс:	а) <code>Serializable</code> ; б) <code>Comparable</code> ; в) <code>AutoCloseable</code> ; г) <code>Runnable</code>
15	Какой модификатор доступа позволяет обращаться к члену класса внутри того же пакета и в классах-наследниках, даже если они находятся в других пакетах?	а) <code>public</code> ; б) <code>private</code> ; в) <code>protected</code> ; г) пакетный (<code>default</code>)
16	В чём основное отличие <code>HashMap</code> от <code>TreeMap</code> ?	а) <code>HashMap</code> сохраняет порядок вставки, а <code>TreeMap</code> – нет; б) <code>HashMap</code> работает на основе хеширования, а <code>TreeMap</code> поддерживает элементы в отсортированном виде; в) <code>HashMap</code> потокобезопасна, а <code>TreeMap</code> – нет; г) <code>TreeMap</code> допускает <code>null</code> в качестве ключа, а <code>HashMap</code> – нет
17	Какой компонент JDBC используется для получения результатов выборки после выполнения запроса <code>SELECT</code> ?	а) <code>Connection</code> ; б) <code>Statement</code> ; в) <code>ResultSet</code> ; г) <code>DriverManager</code>
18	К какому типу относится исключение <code>NullPointerException</code> в Java?	а) Проверяемое (<code>Checked</code>); б) Непроверяемое (<code>Unchecked/Runtime Exception</code>); в) Ошибка (<code>Error</code>); г) Пользовательское исключение
19	Что произойдёт при выполнении кода <code>String s1 = "Java"; String s2 = new String("Java"); System.out.println(s1 == s2);</code> ?	а) <code>true</code> ; б) <code>false</code> ; в) Ошибка компиляции; г) Исключение во время выполнения
20	Какая операция в <code>Stream API</code> является терминальной?	а) <code>.filter()</code> ; б) <code>.map()</code> ; в) <code>.collect()</code> ; г) <code>.sorted()</code>

Блок 3. Вопросы на соответствие (21–25)

№	Задание
21	Установите соответствие между модификаторами доступа в Java и их областью видимости: А) <code>public</code> Б) <code>private</code> В) <code>protected</code>



МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное образовательное
учреждение высшего образования
«Челябинский государственный университет» (ФГБОУ ВО «ЧелГУ»)
Миасский филиал
Кафедра прикладной математики

Фонд оценочных средств по дисциплине «Программирование на Java»
по направлению подготовки 02.03.02 Фундаментальная информатика и информационные технологии, профиль «Компьютерные науки» ФГБОУ ВО «ЧелГУ»

Версия документа - 1

стр. 8 из 29

Первый экземпляр _____

КОПИЯ № _____

	<p>Г) Пакетный (по умолчанию)</p> <ol style="list-style-type: none">1) Доступ только внутри объявляющего класса2) Доступ из любого класса проекта и внешних библиотек3) Доступ внутри класса, в подклассах и в пределах одного пакета4) Доступ только внутри классов одного пакета
2 2	<p>Установите соответствие между интерфейсом коллекции и её основной характеристикой:</p> <p>А) List Б) Set В) Map Г) Queue</p> <ol style="list-style-type: none">1) Хранение пар «ключ–значение», уникальные ключи2) Упорядоченная последовательность, допускает дубликаты, доступ по индексу3) Неупорядоченная коллекция уникальных элементов4) Структура данных, работающая по принципу FIFO (первый вошёл – первый вышел)
2 3	<p>Установите соответствие между типом операции Stream API и примером метода:</p> <p>А) Промежуточная операция фильтрации Б) Промежуточная операция преобразования В) Терминальная операция сбора Г) Терминальная операция проверки</p> <ol style="list-style-type: none">1) <code>.collect(Collectors.toList())</code>2) <code>.filter(x -> x > 0)</code>3) <code>.anyMatch(x -> x % 2 == 0)</code>4) <code>.map(String::toUpperCase)</code>
2 4	<p>Установите соответствие между принципом ООП и механизмом его реализации в Java:</p> <p>А) Инкапсуляция Б) Наследование В) Полиморфизм Г) Абстракция</p> <ol style="list-style-type: none">1) Ключевое слово <code>extends</code>, получение полей и методов суперкласса2) Скрытие полей через <code>private</code> + геттеры/сеттеры3) Переопределение (<code>@Override</code>) и динамическое связывание методов4) Использование <code>abstract class</code> или <code>interface</code> для описания контракта без полной реализации
2 5	<p>Установите соответствие между компонентом JDBC и его функцией:</p> <p>А) <code>DriverManager</code> Б) <code>Connection</code> В) <code>PreparedStatement</code> Г) <code>ResultSet</code></p> <ol style="list-style-type: none">1) Управление пулом соединений и поиск подходящего драйвера БД2) Сессия взаимодействия с конкретной базой данных, транзакции3) Предкомпилированный SQL-запрос с поддержкой параметров-заполнителей (?)



МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное образовательное
учреждение высшего образования
«Челябинский государственный университет» (ФГБОУ ВО «ЧелГУ»)
Миасский филиал
Кафедра прикладной математики

Фонд оценочных средств по дисциплине «Программирование на Java»
по направлению подготовки 02.03.02 Фундаментальная информатика и информационные технологии, профиль «Компьютерные науки» ФГБОУ ВО «ЧелГУ»

Версия документа - 1

стр. 9 из 29

Первый экземпляр _____

КОПИЯ № _____

4) Курсор для построчного чтения данных, возвращаемых запросом SELECT

Ключи к тесту и критерии оценивания

№ задания	Верный ответ	Критерии оценивания
1	ООП – парадигма, основанная на объектах. Принципы: инкапсуляция (скрытие состояния, пример: private поля + методы), наследование (extends), полиморфизм (единый интерфейс, разная реализация, пример: переопределение методов).	2 балла: точное определение + 3 принципа с примерами. 1 балл: определение + 1–2 принципа. 0 баллов: ответ неверен/отсутствует.
2	Абстрактный класс: может иметь поля, конструкторы, реализацию методов, один родитель. Интерфейс: только контракты (до Java 8), множественное наследование, статические/дефолтные методы. Интерфейс – для контракта/API, абстрактный класс – для общей логики родственных классов.	3 балла: чёткое различие + аргументация выбора. 2 балла: различие без аргументации. 1 балл: только одно понятие. 0 баллов: неверно.
3	GC автоматически освобождает память недостижимых объектов. Куча делится на Young Generation (Eden, Survivor), Old/Tenured Generation, Metaspace. Алгоритмы: Stop-the-World, Mark-Sweep, G1, ZGC.	2 балла: описание GC + поколения + примеры алгоритмов. 1 балл: только описание или только поколения. 0 баллов: неверно.
4	List (упорядоченный, дубликаты): ArrayList (O(1) доступ, O(n) вставка), LinkedList (O(n) доступ, O(1) вставка). Set (уникальные): HashSet (O(1)). Map (ключ-значение): HashMap (O(1)).	3 балла: сравнение всех 3 + реализации + сложность. 2 балла: сравнение + реализации. 1 балл: частичное понимание. 0 баллов: неверно.
5	Дженерики обеспечивают type-safety на этапе компиляции. Type Erasure: замена параметризованных типов на их верхнюю границу (Object/Comparable и т.д.) во время компиляции. Wildcard <? extends T> – чтение, <? super T> – запись.	2 балла: определение + стирание типов + wildcard. 1 балл: только определение или стирание. 0 баллов: неверно.
6	Лямбда – анонимная функция. Функциональный интерфейс – интерфейс с одним абстрактным методом. @FunctionalInterface. Пример: Predicate<String> p = s -> s.length() > 5; p.test("Java");	2 балла: синтаксис + понятие ФИ + пример. 1 балл: только синтаксис или только понятие. 0 баллов: неверно.
7	Этапы: загрузка драйвера → DriverManager.getConnection() → создание Statement/PreparedStatement → executeQuery()/executeUpdate() → обработка ResultSet → close(). PreparedStatement защищает от SQL-инъекций, кэширует план выполнения.	3 балла: этапы + отличие PreparedStatement. 2 балла: этапы без отличий. 1 балл: частичное описание. 0 баллов: неверно.



МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное образовательное
учреждение высшего образования
«Челябинский государственный университет» (ФГБОУ ВО «ЧелГУ»)
Миасский филиал
Кафедра прикладной математики

Фонд оценочных средств по дисциплине «Программирование на Java»
по направлению подготовки 02.03.02 Фундаментальная информатика и информационные технологии, профиль «Компьютерные науки» ФГБОУ ВО «ЧелГУ»

Версия документа - 1

стр. 10 из 29

Первый экземпляр _____

КОПИЯ № _____

8	Stream API – функциональная обработка коллекций. Промежуточные (ленивые, возвращают Stream): filter, map, sorted. Терминальные (запускают конвейер, возвращают результат/void): collect, forEach, count.	2 балла: суть + различие типов + ленивость. 1 балл: только суть или только типы. 0 баллов: неверно.
9	Перегрузка: same name, different signature, compile-time. Переопределение: same signature, runtime polymorphism, @Override. Динамическое связывание: JVM определяет тип объекта во время выполнения и вызывает переопределённый метод.	2 балла: чёткое различие + объяснение связывания. 1 балл: только различие. 0 баллов: неверно.
10	Checked: компилятор требует try-catch/throws (IOException). Unchecked: RuntimeException/Error (NullPointerException). try-with-resources автоматически вызывает close() для объектов, реализующих AutoCloseable.	3 балла: типы исключений + механизм try-with-resources + требования. 2 балла: типы + механизм. 1 балл: только типы. 0 баллов: неверно.
11	б) final	1 балл: верный выбор. 0 баллов: неверный выбор.
12	в) ArrayList	1 балл: верный выбор. 0 баллов: неверный выбор.
13	б) Компилятор проверит, действительно ли метод переопределяет метод родительского класса	1 балл: верный выбор. 0 баллов: неверный выбор.
14	в) AutoCloseable	1 балл: верный выбор. 0 баллов: неверный выбор.
15	в) protected	1 балл: верный выбор. 0 баллов: неверный выбор.
16	б) HashMap работает на основе хеширования, а TreeMap поддерживает элементы в отсортированном виде	1 балл: верный выбор. 0 баллов: неверный выбор.
17	в) ResultSet	1 балл: верный выбор. 0 баллов: неверный выбор.
18	б) Непроверяемое (Unchecked/Runtime Exception)	1 балл: верный выбор. 0 баллов: неверный выбор.
19	б) false (сравнение ссылок, разные объекты в памяти)	1 балл: верный выбор. 0 баллов: неверный выбор.
20	в) .collect()	1 балл: верный выбор. 0 баллов: неверный выбор.
21	А–2, Б–1, В–3, Г–4	2 балла: все 4 соответствия верны. 1 балл: 1 ошибка. 0 баллов: ≥2 ошибок.
22	А–2, Б–3, В–1, Г–4	2 балла: все 4 соответствия верны. 1 балл: 1 ошибка. 0 баллов: ≥2 ошибок.
23	А–2, Б–4, В–1, Г–3	2 балла: все 4 соответствия



МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное образовательное
учреждение высшего образования
«Челябинский государственный университет» (ФГБОУ ВО «ЧелГУ»)
Миасский филиал
Кафедра прикладной математики

Фонд оценочных средств по дисциплине «Программирование на Java»
по направлению подготовки 02.03.02 Фундаментальная информатика и информационные технологии, профиль «Компьютерные науки» ФГБОУ ВО «ЧелГУ»

Версия документа - 1

стр. 11 из 29

Первый экземпляр _____

КОПИЯ № _____

		верны. 1 балл: 1 ошибка. 0 баллов: ≥ 2 ошибок.
24	A–2, B–1, B–3, Г–4	2 балла: все 4 соответствия верны. 1 балл: 1 ошибка. 0 баллов: ≥ 2 ошибок.
25	A–1, B–2, B–3, Г–4	2 балла: все 4 соответствия верны. 1 балл: 1 ошибка. 0 баллов: ≥ 2 ошибок.

Итоговая шкала оценивания

Набрано баллов	Процент выполнения	Оценка / Результат зачёта	Уровень сформированности ПК-2
38–42	90–100%	Зачтено (продвинутый)	Свободно владеет синтаксисом Java, ООП-архитектурой, Collections API, Stream API и JDBC; способен проектировать масштабируемые модули и работать с БД
29–37	69–89%	Зачтено (базовый)	Уверенно применяет основные конструкции языка, понимает принципы ООП, generics и обработку исключений; решает типовые задачи разработки
21–28	50–68%	Зачтено (пороговый)	Знает базовый синтаксис и стандартные классы Java; способен выполнять простые скрипты по образцу, допускает неточности в сложных темах (дженерики, Stream API)
0–20	<50%	Не зачтено	Компетенции не сформированы, материал требует повторного изучения

Примеры контрольной работы

Номер задания	Вариант 1
4	<p>Дан массив целых чисел и еще одно целое число. Удалите все вхождения этого числа из массива</p> <p>Решение:</p> <pre>public static void main(String[] args) { int test_array[] = {0,1,2,2,3,0,4,2}; System.out.println(Arrays.toString(removeElement(test_array, 3))); }</pre> <pre>public static int[] removeElement(int[] nums, int val) { int offset = 0;</pre>



	<pre>for(int i = 0; i< nums.length; i++){ if(nums[i] == val){ offset++; } else{ nums[i - offset] = nums[i]; } } return Arrays.copyOf(nums, nums.length - offset); }</pre>
7	<p>Найти корень уравнения $\cos(x^5) + x^4 - 345.3 * x - 23 = 0$ на отрезке $[0;10]$ с точностью по x не хуже, чем $0,001$. Известно, что на промежутке корень единственный.</p> <p>Решение:</p> <pre>public static double func(double x){ return Math.cos(Math.pow(x, 5)) + Math.pow(x, 4) - 345.3 * x - 23; } public static double solve(double start, double end){ if(end - start <= 0.001){ return start; } double x = start + (end - start) / 2; if(func(start) * func(x) > 0){ return solve(x, end); } else { return solve(start, x); } } public static void main(String[] args) { System.out.println(solve(0, 10)); }</pre>
12	<p>Напишите класс BaseConverter для конвертации из градусов по Цельсию в Кельвины, Фаренгейты, и так далее. У метода должен быть метод convert, который и делает конвертацию.</p> <p>Решение:</p>



```
interface Converter {
    double getConvertedValue(double baseValue);
}
class CelsiusConverter implements Converter {
    @Override
    public double getConvertedValue(double baseValue) {
        return baseValue;
    }
}
class KelvinConverter implements Converter {
    @Override
    public double getConvertedValue(double baseValue) {
        return baseValue + 273.15;
    }
}
class FahrenheitConverter implements Converter {
    @Override
    public double getConvertedValue(double baseValue) {
        return 1.8 * baseValue + 32;
    }
}
public class Main {
    public static void main(String[] args) {
        double temperature = 23.5;
        System.out.println("t = " +
            new CelsiusConverter().getConvertedValue(temperature));
        System.out.println("t = " +
            new KelvinConverter().getConvertedValue(temperature));
        System.out.println("t = " +
            new
            FahrenheitConverter().getConvertedValue(temperature));
    }
}
```

Контрольная работа по Golang

1 . Привести примеры на Golang условных операторов

1.1 Оператор if

if command == " выполнение " { // если command равен "выполнение"



МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное образовательное
учреждение высшего образования
«Челябинский государственный университет» (ФГБОУ ВО «ЧелГУ»)
Миасский филиал
Кафедра прикладной математики

Фонд оценочных средств по дисциплине «Программирование на Java»
по направлению подготовки 02.03.02 Фундаментальная информатика и информационные технологии, профиль «Компьютерные науки» ФГБОУ ВО «ЧелГУ»

Версия документа - 1

стр. 14 из 29

Первый экземпляр _____

КОПИЯ № _____

```
    fmt.Println("выполнение 2.")  
  } else if command == "выолнение 3" { // в противном случае, если command равен "вы-  
полнение 3"
```

1.2 Оператор switch

```
switch command { // Сравнивает case с command  
case "выполение 1":  
    fmt.Println("выполение 2 ")  
case "выполнеие 3", "выполенение 4": // Запятая разделяет список возможных значе-  
ний  
    fmt.Println("выполнеие 5")  
case "прочитать знак":  
  
default:  
    fmt.Println("Выполнеие 6.")
```

2. Привести пример оператора цикла на Golang

Оператор for

```
for [инициализация счетчика]; [условие]; [изменение счетчика]{  
    // действия  
  
for i := 1; i < 10; i++){  
    fmt.Println(i * i)
```

3. Оператор структура

```
type Employee struct {  
    Name string  
    Age int  
    Designation string  
    Salary int  
}
```

Контрольная работа по Java

1. Привести пример массива

Оператор for

```
for(i = 0; i < 10; i++)  
    iNumbers[i] = i;
```



МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное образовательное
учреждение высшего образования
«Челябинский государственный университет» (ФГБОУ ВО «ЧелГУ»)
Миасский филиал
Кафедра прикладной математики

Фонд оценочных средств по дисциплине «Программирование на Java»
по направлению подготовки 02.03.02 Фундаментальная информатика и информационные технологии, профиль «Компьютерные науки» ФГБОУ ВО «ЧелГУ»

Версия документа - 1

стр. 15 из 29

Первый экземпляр _____

КОПИЯ № _____

2. Привести пример многомерного массива

```
Er [ ] [ ]
```

```
for(i = 0; i < 2; i++)  
  for(j = 0; j < 3; j++)  
  {  
    System.out.println(Er[i][j]);  
  }
```

3. Привести пример класса

Class

```
public class Air {  
    private String name, id, flight;  
  
    public Air(String name, String id, String flight) {  
        this.name = name;  
        this.id = id;  
        this.flight = flight;  
    }  
  
    // getters/setters  
}
```

Пример документации JAVA на английском языке

1.1 Java Client API Overview The Java Client API provides the following capabilities: • Insert, update, or remove documents and document metadata, either individually or in batches. For details, see “Single Document Operations” on page 36, “Synchronous Multi-Documen Operations” on page 70, or “Asynchronous Multi-Documen Operations” on page 92. • Query documents, lexicons, and semantic data. For details, see “Searching” on page 144. • Extract data from MarkLogic as tables. For details, see “Optic Java API for Relational Operations” on page 218. • Persist, retrieve, and query Java objects in stored in MarkLogic. For details, see “POJO Data Binding Interface” on page 226. • Configure persistent and dynamic query options. For details, see “Query Options” on page 190. • Apply transformations to new content and search results. For details, see “Content Transformations” on page 282. • Extend the Java API to expose custom capabilities you install on MarkLogic Server. For details, see “Extending the Java API” on page 288. MarkLogic Server Introduction to the Java API MarkLogic 10—May, 2019 Java Application Developer’s Guide—Page 13 When working with the Java API, you first



МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное образовательное
учреждение высшего образования
«Челябинский государственный университет» (ФГБОУ ВО «ЧелГУ»)
Миасский филиал
Кафедра прикладной математики

Фонд оценочных средств по дисциплине «Программирование на Java»
по направлению подготовки 02.03.02 Фундаментальная информатика и информационные технологии, профиль «Компьютерные науки» ФГБОУ ВО «ЧелГУ»

Версия документа - 1

стр. 16 из 29

Первый экземпляр _____

КОПИЯ № _____

create a manager for the type of document or operation you want to perform on the database (for instance, a `JSONDocumentManager` to write and read JSON documents or a `QueryManager` to search the database). To write or read the content for a database operation, you use standard Java APIs such as `InputStream`, `DOM`, `StAX`, `JAXB`, and `Transformer` as well as Open Source APIs such as `JDOM` and `Jackson`. The Java API provides a handle (a kind of adapter) as a uniform interface for content representation. As a result, you can use APIs as different as `InputStream` and `DOM` to provide content for one `read()` or `write()` method. In addition, you can extend the Java API so you can use the existing `read()` or `write()` methods with new APIs that provide useful representations for your content. This chapter covers a number of basic architecture aspects of the Java API, including fundamental structures such as database clients, managers, and handles used in almost every program you will write with it. Before starting to code, you need to understand these structures and the concepts behind them. The MarkLogic Java Client API is built on top of the MarkLogic REST API. The REST API, in turn, is built using XQuery that is evaluated against an HTTP App Server. For this reason, you need a REST API instance on MarkLogic Server to use the Java API. A suitable REST API instance on port 8000 is pre-configured when you install MarkLogic Server. You can also create your own on another port. For details, see “Choose a REST API Instance” on page 15.

Критерии оценивания текущей аттестации


Критерии оценивания контрольных работ:

«отлично»

- 1) студент легко ориентируется в содержании учебного материала, свободно пользуется понятийным аппаратом, обладает умением связывать теорию с практикой, высказывать и обосновывать свои суждения;
- 1) знает и правильно применяет формулы;
- 2) знает и правильно применяет нормативные документы;
- 3) решение задачи записано понятно, аккуратно, последовательно;
- 4) записан правильный ответ

«хорошо»

- 1) студент демонстрирует полное освоение теоретического материала, владеет понятийным аппаратом, ориентируется в изученном материале, осознанно применяет знания для решения практических задач, грамотно излагает свою позицию;

 <p>МИНОБРНАУКИ РОССИИ Федеральное государственное бюджетное образовательное учреждение высшего образования «Челябинский государственный университет» (ФГБОУ ВО «ЧелГУ») Миасский филиал Кафедра прикладной математики</p>			
<p>Фонд оценочных средств по дисциплине «Программирование на Java» по направлению подготовки 02.03.02 Фундаментальная информатика и информационные технологии, профиль «Компьютерные науки» ФГБОУ ВО «ЧелГУ»</p>			
Версия документа - 1	стр. 17 из 29	Первый экземпляр _____	КОПИЯ № _____

- 2) знает и применяет формулы и нормативные документы, но допускает небольшие неточности;
- 3) решение задачи записано, но не приведены формулы, с помощью которых были проведены расчеты;
- 4) записан правильный ответ

«удовлетворительно»

- 1) студент демонстрирует неполное освоение теоретического материала, плохо владеет понятийным аппаратом, плохо ориентируется в изученном материале, неуверенно излагает свою позицию;
- 2) знает отдельные формулы и нормативные документы, но допускает значительные неточности в их применении;
- 3) решение задачи записано неверно, но не приведены формулы, с помощью которых были проведены расчеты;
- 4) записан правильный ответ

«неудовлетворительно»

- 1) студент имеет разрозненные, бессистемные знания, не умеет выделять главное и второстепенное, допускает ошибки в определении понятий, искажающие их смысл;
- 2) беспорядочно и неуверенно излагает материал, не может применять знания для решения практических задач;
- 3) решение задачи записано неверно либо отсутствует;
- 4) записан неправильный ответ либо не записан ответ

Работа, по результатам проверки которой выставлена оценка **«неудовлетворительно»**, возвращается студенту на доработку. Студент не может быть допущен до сдачи зачета до тех пор, пока не представит исправленную работу.

Критерии оценивания понимания текста

“5” – «отлично» (96-100 баллов) «зачтено» – 100 – 75% понимания основного содержания текста, студент умеет свободно (почти свободно) и аргументировано высказываться, обстоятельно излагать содержание прочитанного, логически выстраивать свое сообщение, развивать отдельные положения и делать соответствующие выводы, изложение оформлено правильно грамматически и лексически.



МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное образовательное
учреждение высшего образования
«Челябинский государственный университет» (ФГБОУ ВО «ЧелГУ»)
Миасский филиал
Кафедра прикладной математики

Фонд оценочных средств по дисциплине «Программирование на Java»
по направлению подготовки 02.03.02 Фундаментальная информатика и информационные технологии, профиль «Компьютерные науки» ФГБОУ ВО «ЧелГУ»

Версия документа - 1

стр. 18 из 29

Первый экземпляр _____

КОПИЯ № _____

“4” – «хорошо» (76-95 баллов) «зачтено» – 75% понимания основного содержания текста, студент может кратко излагать содержание прочитанного обосновать и объяснить свои взгляды, в изложении допускаются 2-3 незначимые грамматические или лексические ошибки.

“3” – «удовлетворительно» (60-75 балл) «зачтено» – 75 – 50% понимания основного содержания текста, студент может использовать простые фразы и предложения, но недостаточно понятно и обстоятельно излагать содержание прочитанного, в изложении 2-3 грамматические или лексические ошибки.

"2" – «неудовлетворительно» (0-60 балл) «не зачтено» – менее 50% понимания основного содержания текста, искажение содержания, превышение количества грамматических и лексических ошибок, студент владеет недостаточным словарным запасом, затрудняется в изложении прочитанного.

4. ПОРЯДОК ПРОВЕДЕНИЯ И КРИТЕРИИ ОЦЕНИВАНИЯ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ

4.1. Порядок проведения и содержание оценочных средств промежуточной аттестации

Промежуточная аттестация проводится в форме зачёта в 6 семестре и экзамена в 7 семестре.

Порядок проведения зачёта

Зачет проводится в форме письменно-устного опроса. На письменный ответ студента отводится 40 минут, затем ответ проверяется преподавателем, при необходимости могут быть заданы уточняющие вопросы.

Оценочные средства для промежуточной аттестации представлены базой вопросов к зачёту и типовыми заданиями.

Вопросы для зачета

1. Базовый синтаксис программирования на языке Java.
2. JAR - понятие, свойства, назначение.
3. JDK, JRE.
4. Java. Переменные и типы данных.
5. Оператор присваивания, логические выражения.
6. Массивы и их реализация в языке программирования Java
7. Логические и математические операции.
8. Операторы управления. Синтаксис, структура, модификации.
9. Операторы циклов. Оператор цикла с условием. Циклы «до» или цикл с постусловием



МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное образовательное
учреждение высшего образования
«Челябинский государственный университет» (ФГБОУ ВО «ЧелГУ»)
Миасский филиал
Кафедра прикладной математики

Фонд оценочных средств по дисциплине «Программирование на Java»
по направлению подготовки 02.03.02 Фундаментальная информатика и информационные технологии, профиль «Компьютерные науки» ФГБОУ ВО «ЧелГУ»

Версия документа - 1

стр. 19 из 29

Первый экземпляр _____

КОПИЯ № _____

10. Структура циклов.
11. Функции. Синтаксис, обязательные и необязательные параметры.
12. Алгоритмы сортировки массива.
13. Принципы ООП (Объектно - ориентированного программирования).
14. Объекты, свойства и методы, события.
15. Программирование клиентского приложения. Обработка форм. События.
16. Операторы управления потоком языка GO
17. Методы и интерфейсы языка GO
18. Пояснить "Многопоточность" языка GO
19. Объекты и классы
20. NetBeans
21. Идентификаторы
22. Переменные
23. Инструкции присваивания и выражения присваивания
24. Именованные константы
25. Расширенные операторы присваивания
26. Инструкции if
27. Тип данных boolean
28. Основы Go: цикл for
29. Основы Go: срезы
30. Основы Go:методы
31. Основы Go: каналы
32. Основы Go: Go-процедуры (goroutines)
33. Основы Go:замыкания (closures)

Перечень вопросов к экзамену

1. Стандартная библиотека коллекций языка Java. Интерфейсы, реализации и алгоритмы коллекций. Структура библиотеки коллекций. Коллекции, множества и списки. Использование реализаций интерфейсов коллекций. Карты (maps) в библиотеке коллекций. Использование различных реализаций карт. Итераторы карт и коллекций. Стандартные алгоритмы библиотеки для работы с коллекциями и массивами.
2. Методы рефакторинга для преобразования структуры программы на языке Java. Рефакторинг типов. Рефакторинг иерархии наследования. Перемещение методов по иерархии наследования и между классами. Использование среды Eclipse для рефакторинга программы на языке Java.



МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное образовательное
учреждение высшего образования
«Челябинский государственный университет» (ФГБОУ ВО «ЧелГУ»)
Миасский филиал
Кафедра прикладной математики

Фонд оценочных средств по дисциплине «Программирование на Java»
по направлению подготовки 02.03.02 Фундаментальная информатика и информационные технологии, профиль «Компьютерные науки» ФГБОУ ВО «ЧелГУ»

Версия документа - 1

стр. 20 из 29

Первый экземпляр _____

КОПИЯ № _____

3. Родовые типы в языке Java. Назначение родовых типов. Не ковариантность родовых типов. Родовой тип wildcard. Родовые методы. Ограниченные родовые типы.

4. Потoki байтового вывода языка Java. Назначение и возможности классов OutputStream, ByteArrayOutputStream, FileOutputStream, PipedOutputStream, FilterOutputStream, BufferedOutputStream, DataOutputStream, PrintStream. Потoki символьного вывода языка Java.

5. Потoki ввода языка Java. Назначение и возможности классов InputStream, ByteArrayInputStream, FileInputStream, PipedInputStream, FilterInputStream, BufferedInputStream, DataInputStream. Потoki символьного ввода языка Java. Чтение данных из потока с помощью класса Scanner.

6. Java DataBase Connectivity (JDBC) - стандарт взаимодействия Java-приложений с различными СУБД. Подключение драйверов СУБД. Выполнение SQL команд. Управление транзакциями. 7. Использование JDBC для извлечения информации о структуре (метаданных) базы данных. Визуализация структуры базы данных.

8. Интернационализация программ в языке Java. Файлы текстовых ресурсов. Выбор языка пользователя для выдачи текстовых сообщений.

9. Лямбда выражения в языке Java8. Замена анонимных классов лямбда выражениями. Функциональные интерфейсы. Синтаксис Lambda-выражений. Примеры функциональных интерфейсов из пакета java.util.function.

10. Потoki в языке Java8. Определение потоков. Отличие операций потоков от операций коллекций. Обработка данных потока в конвейере. Методы для порождения потоков. Методы для преобразования потоков в конвейере. Преобразования в конце конвейера.

11. Модули в Java 9. Назначение модулей. Синтаксис описания модулей. Зависимость от модулей. Экспорт пакетов модуля. Открытость модуля.

Механизм Reflection языка Java. Получение класса объекта в оперативной памяти. Получение класса объекта с использованием URL. Метаданные для классов. Метаданные для примитивов и массивов. Использование конструкторов класса для загрузки драйверов. Вызов методов. Доступ к полям.

12. Класс Shell. Конструктор, стили, события, характерные методы класса Shell. Класс Composite. Конструктор, стили, события, характерные методы класса Composite. Класс Canvas. Конструктор, стили, события, характерные методы класса Canvas.

13. Структурирование интерфейса пользователя с помощью классов TabFolder и TabItem. Конструктор, стили, события, характерные методы классов TabFolder и TabItem.



14. Рисование графических элементов с помощью класса Graphics Context (GC) библиотеки Standard Widget Toolkit. Рисование линий, фигур, изображений графических файлов, курсоров. Задание их атрибутов представления на экране.

15. Обработка событий в библиотеке Standard Widget Toolkit. Интерфейс обработчика события PaintListener для перерисовки изображений на экране.

16. Интерфейс обработчика клавиатуры мыши MouseListener, вращения колеса MouseWheelListener, перемещения мыши MouseMoveListener.

17. Списки библиотеки SWT в интерфейсе пользователя. Инициализация списков. Слушатели событий. Определение выбранного элемента списка.

18. Кнопки библиотеки SWT в интерфейсе пользователя. Радио-кнопки. Инициализация кнопок. Слушатели событий. Определение выбора пользователя.

Примеры типовых заданий для зачёта и экзамена

1	Привести примеры данных	<pre>package main import "fmt" func main() { fmt.Println("go" + "lang") fmt.Println("1+1 =", 1+1) fmt.Println("7.0/3.0 =", 7.0/3.0) } \$ go run values.go</pre>
2	Привести пример констант и числовых типов данных	<pre>package main import "fmt" import "math" const s string = "constant" func main() { fmt.Println(s) const n = 500000000 const d = 3e20 / n fmt.Println(d) fmt.Println(int64(d)) fmt.Println(math.Sin(n)) } \$ go run constant.go</pre>



		<code>constant 6e+11 600000000000 -0.28470407323754 404</code>
3	<p>Привести примеры Срезов (Slices)</p> <p>Установка и чтение значений происходит также как в массиве.</p> <p>Срезы можно копировать.</p> <p>Можно объявить и инициализировать переменную для среза в одну строку.</p> <p>Срезы могут быть составлены в многомерные структуры данных.</p>	<pre>package main import "fmt" func main() { s := make([]string, 3) fmt.Println("emp:", s) s[0] = "a" s[1] = "b" s[2] = "c" fmt.Println("set:", s) fmt.Println("get:", s[2]) fmt.Println("len:", len(s)) s = append(s, "d") s = append(s, "e", "f") fmt.Println("apd:", s) c := make([]string, len(s)) copy(c, s) fmt.Println("cpy:", c) l := s[2:5] fmt.Println("sl1:", l) l = s[:5] fmt.Println("sl2:", l) l = s[2:] fmt.Println("sl3:", l) t := []string{"g", "h", "i"} fmt.Println("dcl:", t) twoD := make([][]int, 3) for i := 0; i < 3; i++ { innerLen := i + 1 twoD[i] = make([]int, innerLen) } for j := 0; j < innerLen; j++ { twoD[i][j] = i + j } } fmt.Println("2d: ", twoD) }</pre>



		<pre>\$ go run slices.go emp: [] set: [a b c] get: c len: 3 apd: [a b c d e f] cpy: [a b c d e f] sl1: [c d e] sl2: [a b c d e] sl3: [c d e f] dcl: [g h i] 2d: [[0] [1 2] [2 3 4]]</pre>
4	Привести пример направления каналов (Channel Directions)	<pre>package main import "fmt" func ping(pings chan<- string, msg string) { pings <- msg } func pong(pings <-chan string, pongs chan<- string) { msg := <-pings pongs <- msg } func main() { pings := make(chan string, 1) pongs := make(chan string, 1) ping(pings, "passed message") pong(pings, pongs) fmt.Println(<-pongs) } \$ go run channel-directions.go passed message</pre> <p>Функция ping принимает канал только для отправки значений. Функция pong принимает один канал для приёма (pings) и второй для отправки (pongs).</p>
5	Привести пример Тайм-ауты (Timeouts)	<pre>package main import "time" import "fmt" func main() { c1 := make(chan string, 1) go func() { time.Sleep(time.Second * 2) c1 <- "result 1" }() select {</pre> <p>Здесь внешний вызов, который возвращает результат в канал c1 спустя 2 сек. select выполняет вари-</p>



	<p>ант с тайм-аутом. <code>res := <- c1</code> ожидает результата и <code><-Time.After</code></p> <p>Если указать длинный тайм-аут в 3 сек., приём из <code>c2</code> будет успешен</p>	<pre>case res := <-c1: fmt.Println(res) case <-time.After(time.Second * 1): fmt.Println("timeout 1") } c2 := make(chan string, 1) go func() { time.Sleep(time.Second * 2) c2 <- "result 2" }() select { case res := <-c2: fmt.Println(res) case <-time.After(time.Second * 3): fmt.Println("timeout 2") } \$ go run timeout- s.go · timeout 1 result 2</pre>
6	<p>Привести пример выполнения Мьютекса (Mutexes)</p> <p>Пусть состояние (<code>state</code>) будет картой. Этот <code>mutex</code> будет синхронизировать доступ к <code>state</code>. Для сравнения подхода на основе мьютексов с другим, который увидим позже, <code>ops</code> будет считать количество операций над состоянием.</p> <p>Запускаем 100 горутин для выполнения повторяющихся чтений состояния.</p> <p>Для каждой операции чтения выбираем ключ для доступа, закрываем (<code>Lock()</code>)</p>	<pre>package main import ("fmt" "math/rand" "runtime" "sync" "sync/atomic" "time") func main() { var state = make(map[int]int) var mutex = &sync.Mutex{} var ops int64 = 0 for r := 0; r < 100; r++ { go func() { total := 0 for { key := rand.Intn(5) mu- tex.Lock()</pre>



	<p>этот <code>mutex</code> для получения эксклюзивного доступа к <code>state</code>, читаем значение для выбранного ключа, <code>Unlock()</code> мьютекс и увеличиваем счётчик <code>ops</code>.</p> <p>Для гарантии того, что горутина не остановит планировщик, явно выполним после каждой операции <code>runtime.Gosched()</code>. Такое выполнение производится автоматически, например, каждой операцией канала или при блокирующих вызовах подобных <code>time.Sleep</code>, но в нашем случае нужно выполнить его вручную.</p> <p>Запускаем 10 горутин для имитации записи. Используем тот же подход, что и для имитации чтения.</p> <p>Дадим время в 1 сек. для работы 10 горутин с <code>state</code> и <code>mutex</code>.</p> <p>Отчёт об окончательном числе операций.</p> <p>Фиксируем окончательно состояние <code>state</code> и показываем его.</p> <p>Программа покажет, что мы выполнили 3</p>	<pre>total += state[key] mu- tex.Unlock() atomic.AddInt64(&ops, 1) runtime.Gosched() } }() } for w := 0; w < 10; w++ { go func() { for { key := rand.Intn(5) val := rand.Intn(100) mu- tex.Lock() state[key] = val mu- atom- ic.AddInt64(&ops, 1) run- time.Gosched() } }() } time.Sleep(time.Second) opsFinal := atom- ic.LoadInt64(&ops) fmt.Println("ops:", opsFinal) mutex.Lock() fmt.Println("state:", state) mutex.Unlock() } \$ go run mutex- es.go</pre>
--	--	--



МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное образовательное
учреждение высшего образования
«Челябинский государственный университет» (ФГБОУ ВО «ЧелГУ»)
Миасский филиал
Кафедра прикладной математики

Фонд оценочных средств по дисциплине «Программирование на Java»
по направлению подготовки 02.03.02 Фундаментальная информатика и информационные технологии, профиль «Компьютерные науки» ФГБОУ ВО «ЧелГУ»

Версия документа - 1

стр. 26 из 29

Первый экземпляр _____

КОПИЯ № _____


		500 000 операций с помощью <code>mutex</code> -синхронизации <code>state</code> .	ops: 3598302 state: map[1:38 4:98 2:23 3:85 0:44]
7	JAVA Ввести случайную загрузку мощности от 1 до 100 %	<pre>public class Test { // объявляется класс Test public static void main(String[] args) { // так включаются программы, // просто запомнить int i = (int) (Math.random() * 100 + 1); System.out.println("Загрузка мощности:" + i + " %"); } }</pre>	
8	Вывести разность чисел считанных с экрана	<pre>import java.util.Scanner; // импортируем класс Scanner public class Main { // объявляется класс Main public static void main(String[] args) { // так включаются программы, // просто запомнить Scanner sc = new Scanner(System.in); System.out.println("Введите число:"); int a = sc.nextInt(); System.out.println("Введите число:"); int b = sc.nextInt(); sc.close(); System.out.println("Разность:" + (a - b)); } }</pre>	
9	Ввод математических формул	<pre>import java.util.Scanner; public class MultMain { private static double getValue(Scanner in) { if (!in.hasNextDouble()) { System.out.println("Wrong arguments, x,n double : java MultMain 1.2 5"); System.exit(0); } return in.nextDouble(); // если число будет неверным вылетит исключение } }</pre>	



```
public static void main(String[] args) {
    Scanner in;
    double x = 1.21;
    double n = 5;
    double result = 1;

    if (args == null || args.length < 2) {
        System.out.println("Program call format (x,n double): java Mult-
Main [x] [n]");
        System.out.println("Default values: x = "+x+", n = "+ n);
    } else {
        in = new Scanner(args[0] + " " + args[1]);
        x = getValue(in);
        n = (int) getValue(in);
    }

    x = Math.pow(x, n);
    for (int i = 0; i < 10; i++) {
        for (int j = 0; j < (int) n; j++) {
            result = result * (Math.sin(x + Math.pow(j, 4)) + 1);
        }
    }
    System.out.printf("x = %e n = %e result=%e\n", x, n, result);
    System.out.printf("x = %f n = %f result=%f\n", x, n, result);
}
}
```

 МИНОБРНАУКИ РОССИИ Федеральное государственное бюджетное образовательное учреждение высшего образования «Челябинский государственный университет» (ФГБОУ ВО «ЧелГУ») Миасский филиал Кафедра прикладной математики			
Фонд оценочных средств по дисциплине «Программирование на Java» по направлению подготовки 02.03.02 Фундаментальная информатика и информационные технологии, профиль «Компьютерные науки» ФГБОУ ВО «ЧелГУ»			
Версия документа - 1	стр. 28 из 29	Первый экземпляр _____	КОПИЯ № _____

4.2. Критерии оценивания компетенций в ходе промежуточной аттестации

Код компетенции	Планируемые результаты	Критерии оценивания	
		Зачтено	Не зачтено
ПК-2	Знает парадигмы объектно-ориентированного программирования и их особенности на языке Java, основы разработки программных интерфейсов на языке Java	Знает парадигмы объектно-ориентированного программирования и их особенности на языке Java, основы разработки программных интерфейсов на языке Java	Не знает парадигмы объектно-ориентированного программирования и их особенности на языке Java, основы разработки программных интерфейсов на языке Java
	Умеет разрабатывать программные интерфейсы на языке Java, проектировать и разрабатывать абстрактные классы и супер-классы на языке Java	Умеет разрабатывать программные интерфейсы на языке Java, проектировать и разрабатывать абстрактные классы и супер-классы на языке Java	Не умеет разрабатывать программные интерфейсы на языке Java, проектировать и разрабатывать абстрактные классы и супер-классы на языке Java
	Владеет навыками разработки программных интерфейсов на языке Java, разработки классов на языке Java, проектирования UML-диаграмм для классов на языке Java	Владеет навыками разработки программных интерфейсов на языке Java, разработки классов на языке Java, проектирования UML-диаграмм для классов на языке Java	Не владеет навыками разработки программных интерфейсов на языке Java, разработки классов на языке Java, проектирования UML-диаграмм для классов на языке Java

Критерии оценивания промежуточной аттестации Критерии оценивания зачета

Письменно-устный ответ студента по вопросам дисциплины оценивается положительно с выставлением оценки «**зачтено**» в следующем случае:

– студент обнаруживает знание и понимание основных положений учебного материала, возможно, допускает неточности и несущественные ошибки в определении понятий, формулировке положений, не допускает или допускает незначительные ошибки в решении задач.

Оценка «**не зачтено**» за письменный и письменно-устный ответ студента по вопросам дисциплины выставляется в случаях, когда:



МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное образовательное
учреждение высшего образования
«Челябинский государственный университет» (ФГБОУ ВО «ЧелГУ»)
Миасский филиал
Кафедра прикладной математики

Фонд оценочных средств по дисциплине «Программирование на Java»
по направлению подготовки 02.03.02 Фундаментальная информатика и информационные технологии, профиль «Компьютерные науки» ФГБОУ ВО «ЧелГУ»

Версия документа - 1

стр. 29 из 29

Первый экземпляр _____

КОПИЯ № _____

– студент имеет разрозненные, бессистемные знания: не умеет выделять главное и второстепенное; допускает ошибки в определении понятий, формулировке теоретических положений, искажает их смысл; беспорядочно и неуверенно излагает материал;

– не умеет соединять теоретические положения с практикой; не умеет применять знания для обоснования и объяснения фактов.

4.3 Результаты промежуточной аттестации и уровни сформированности компетенций

Уровень освоения компетенций	Оценка
Продвинутый	зачтено
Базовый	зачтено
Пороговый	зачтено
компетенции не сформированы	не зачтено

Уровни формирования компетенций:

1. Пороговый уровень:

- предполагает формирование компетенций на начальном уровне: знание основ пакетов прикладных программ;
- студент способен давать ответы на теоретические вопросы дисциплины на удовлетворительном уровне, читать и понимать документацию программ на английском языке.

2. Базовый уровень:

- предполагает формирование компетенций на более высоком уровне: формируется комплексное знание особенностей и применения пакетов прикладных программ;
- студент способен давать развернутые ответы на теоретические вопросы дисциплины; способен решать практические задания с использованием документации программ на английском языке.

3. Продвинутый уровень:

- предполагает формирование компетенций на высоком уровне, использует полученные знания и умения при изучении смежных дисциплин, обнаруживает готовность к самостоятельной профессиональной деятельности;
- студент способен аргументировать собственную точку зрения, формулировать собственные выводы на основе применения усвоенных компетенций.