

Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Гаскаев Сергей Валерьевич
Должность: Ректор
Дата подписания: 17.11.2025 16:13:13
Уникальный программный ключ:
04c19ed81bf98f3b6c773484b9a8788b52252



МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное
образовательное учреждение высшего образования
«Челябинский государственный университет» (ФГБОУ ВО «ЧелГУ»)

Фонд оценочных средств для промежуточной аттестации по дисциплине (модулю) «Программирование на языке Java (научный семинар)» по направлению подготовки 02.03.02 «Фундаментальная информатика и информационные технологии» направленности «Прикладное программирование и системы искусственного интеллекта» ФГБОУ ВО «ЧелГУ»

стр. 1

Фонд оценочных средств для промежуточной аттестации по дисциплине (модулю)
«Программирование на языке Java (научный семинар)»

Направление подготовки (специальность)
02.03.02 «Фундаментальная информатика и информационные технологии»

Направленность (профиль)
«Прикладное программирование и системы искусственного интеллекта»

Присваиваемая квалификация
Бакалавр

Форма обучения
Очная

Челябинск, 2025 г.



Содержание

1. Паспорт фонда оценочных средств	3
2. Перечень формируемых компетенций	4
3. Содержание оценочных средств по дисциплине	7
3.1. Виды оценочных средств	7
3.2. Содержание оценочных средств	8
4. Порядок проведения и критерии оценивания промежуточной аттестации	24
4.1. Порядок проведения промежуточной аттестации	24
4.2. Критерии оценивания промежуточной аттестации по видам оценочных средств	24
4.3. Результаты промежуточной аттестации и уровни сформированности компетенций.....	24



МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное
образовательное учреждение высшего образования
«Челябинский государственный университет» (ФГБОУ ВО «ЧелГУ»)

Фонд оценочных средств для промежуточной аттестации по дисциплине (модулю) «Программирование на языке Java (научный семинар)» по направлению подготовки 02.03.02 «Фундаментальная информатика и информационные технологии» направленности «Прикладное программирование и системы искусственного интеллекта» ФГБОУ ВО «ЧелГУ»

стр. 3

1. Паспорт фонда оценочных средств

Направление подготовки: 02.03.02 Фундаментальная информатика и информационные технологии.

Направленность (профиль): Прикладное программирование и системы искусственного интеллекта.

Дисциплина: Программирование на языке Java (научный семинар).

Семестры: 8.

Форма промежуточной аттестации: экзамен в 8 семестре.

Для оценивания результатов обучения используется балльно-рейтинговая система.



2. Перечень формируемых компетенций

Изучение дисциплины «Программирование на языке Java (научный семинар)» направлено на формирование компетенций, приведённых в Таблице 1.

Таблица 1. Результаты обучения по дисциплине.

Коды компетенции согласно ФГОС (ОПОП ВО)	Содержание компетенций согласно ФГОС (ОПОП ВО)	Индикаторы достижения компетенции согласно ОПОП	Перечень планируемых результатов обучения по дисциплине
УК-1	Способен осуществлять поиск, критический анализ и синтез информации, применять системный подход для решения поставленных задач	УК-1.1. Выполняет поиск информации, определяет критерии системного анализа поставленных задач. УК-1.2. Использует критический анализ, систематизацию и обобщение информации для решения поставленных задач.	Знать: методы и способы поиска информации, определения критериев системного анализа поставленных задач в области программирования на языке Java. Уметь: выполнять поиск информации, определять критерии системного анализа поставленных задач в области программирования на языке Java. Владеть: навыком критического анализа, систематизации и обобщения информации применительно к программированию на языке Java.
ПК-2	Способен к осуществлению интеграции программных модулей и компонент и проверки работоспособности программного продукта на основе международных и профессиональных стандартов информационных	ПК-2.1. Обладает знаниями о методах и средствах сборки модулей и компонент программного обеспечения, о разработке процедур для развертывания программного обеспечения, миграции и преобразования данных, о создании программных интерфейсов; о методах и механизмах оценки и анализа функционирования средств и систем информационных технологий; о международных и	Знать: синтаксис языка, основные возможности библиотек. Уметь: проектировать иерархию классов, читать и писать программный код на языке Java. Владеть: навыками программирования на языке Java.



Коды компетенции согласно ФГОС (ОПОП ВО)	Содержание компетенций согласно ФГОС (ОПОП ВО)	Индикаторы достижения компетенции согласно ОПОП	Перечень планируемых результатов обучения по дисциплине
	технологий, современных парадигм и методологий, инструментальных и вычислительных средств, методов и механизмов оценки и анализа функционирования средств и систем информационных технологий	профессиональных стандартах информационных технологий, о современных парадигмах и методологиях, инструментальных и вычислительных средствах. ПК-2.2. Демонстрирует умения: применять методы и средства сборки модулей и компонент программного обеспечения, разработки процедур для развертывания программного обеспечения, миграции и преобразования данных, создания программных интерфейсов; проводить проверку и оценку работоспособности программного продукта. ПК-2.3. Имеет практический опыт (навыки): сборки модулей и компонент программного обеспечения, разработки процедур для развертывания программного обеспечения, миграции и преобразования данных, создания программных интерфейсов; оценки работоспособности программного продукта.	
ПК-3	Способен к разработке требований и проектированию программного обеспечения на основе применения базовых математических знаний и информационных	ПК-3.1. Обладает знаниями о методах и средствах проектирования программного обеспечения, структур данных, баз данных, программных интерфейсов. ПК-3.2. Демонстрирует умения: разрабатывать требования к программному продукту, применять методы и средства проектирования программного	Знать: методы и приемы разработки программного обеспечения на платформе Java. Уметь: создавать простые приложения на основе современной ИТ-платформы с использованием библиотеки классов данной платформы и



Коды компетенции согласно ФГОС (ОПОП ВО)	Содержание компетенций согласно ФГОС (ОПОП ВО)	Индикаторы достижения компетенции согласно ОПОП	Перечень планируемых результатов обучения по дисциплине
	технологий при решении проектно-технических и прикладных задач	обеспечения, структур данных, баз данных, программных интерфейсов. ПК-3.3. Имеет практический опыт (навыки) проектирования программного обеспечения, структур данных, баз данных, программных интерфейсов.	разработкой собственных классов. Владеть: технологией разработки, тестирования и развертывания программного обеспечения на платформе Java.



3. Содержание оценочных средств по дисциплине

3.1. Виды оценочных средств

Таблица 2. Виды оценочных средств.

№ п/п	Код компетенции / планируемые результаты обучения	Контролируемые темы / разделы	Наименование оценочного средства для текущего контроля	Наименование оценочного средства на промежуточной аттестации
1	<p>УК-1 Знать: методы и способы поиска информации, определения критериев системного анализа поставленных задач в области программирования на языке Java. Уметь: выполнять поиск информации, определять критерии системного анализа поставленных задач в области программирования на языке Java. Владеть: навыком критического анализа, систематизации и обобщения информации применительно к программированию на языке Java.</p>	Технология Java и ее применение.	Комплект задач для лабораторных работ Проектное задание	Вопросы для экзамена, практические задания (для экзамена)
2	<p>ПК-2 Знать: синтаксис языка, основные возможности библиотек. Уметь: проектировать иерархию классов, читать и писать программный код на языке Java. Владеть: навыками программирования на языке Java.</p> <p>ПК-3 Знать: методы и приемы разработки программного обеспечения на платформе Java.</p>	Принципы объектно-ориентированного дизайна приложений.	Вопросы для экзамена, практические задания (для экзамена)	Вопросы для экзамена, практические задания (для экзамена)
3	<p>Уметь: создавать простые приложения на основе современной ИТ-платформы с использованием библиотеки классов данной платформы и разработкой собственных классов.</p>	Разработка веб-приложений и веб-сервисов на платформе Java	Вопросы для экзамена, практические задания (для экзамена)	Вопросы для экзамена, практические задания (для экзамена)



№ п/п	Код компетенции / планируемые результаты обучения	Контролируемые темы / разделы	Наименование оценочного средства для текущего контроля	Наименование оценочного средства на промежуточной аттестации
	Владеть: технологией разработки, тестирования и развертывания программного обеспечения на платформе Java.			

Типовые задания, критерии и показатели оценивания в рамках текущего контроля представлены в рабочей программе дисциплины (модуля). Полные комплекты оценочных средств и контрольно-измерительных материалов хранятся на кафедре.

3.2. Содержание оценочных средств

Промежуточная аттестация проводится в виде экзамена в 8 семестре.

Вопросы к экзамену:

1. История появления и развития платформы Java
2. Структура современной платформы Java
3. Возможности, предоставляемые современной платформой Java
4. Языки программирования платформы Java. Обзор
5. Средства разработки под платформу Java. Обзор
6. Приложение Java. Разработка и развертывание
7. Возможности технологий Java
8. Основные инструменты платформы Java
9. Архитектура JVM
10. Пакетная организация классов. Объявление и доступность
11. Пакетная организация классов. Выполнение
12. Пакетная организация классов. Указание зависимых классов
13. Язык Java. История создания, этапы и логика развития.
14. Язык Java. Принципы, лежащие в его основе.
15. Структура и синтаксис примитивной консольной программы Java
16. Соглашения о наименованиях
17. Документирование кода. Аннотации javadoc
18. Базовый Java. Система встроенных типов. Типобезопасность.
19. Базовый Java. Ссылочные типы и типы значений.
20. Базовый Java. Примитивные встроенные типы данных.
21. Базовый Java. Базовые управляющие конструкции. Синтаксис. Примеры использования.
22. Базовый Java. Базовые управляющие конструкции. Циклы. Рекомендации по проектированию.
23. Базовый Java. Аннотации
24. Базовый Java. Ссылочные типы. Массивы. Разновидности. Синтаксис. Примеры использования.
25. Базовый Java. Ссылочные типы. Строки. Синтаксис. Примеры использования.



26. Базовый Java. Консольный ввод вывод.
27. Базовый Java. Базовые типы данных. Иерархия.
28. Базовый Java. Ссылочные типы. Классы - оболочки.
29. Базовый Java. Ссылочные типы. Класс. Общие характеристики.
30. Базовый Java. Ссылочные типы. Класс. Структура.
31. Базовый Java. Ссылочные типы. Класс. Поля (класса, экземпляра), константы.
32. Базовый Java. Ссылочные типы. Класс. Ограничение доступа.
33. Базовый Java. Ссылочные типы. Класс. Конструкторы.
34. Базовый Java. Ссылочные типы. Класс. Методы.
35. Базовый Java. Ссылочные типы. Класс. Статические методы и поля.
36. Базовый Java. Ссылочные типы. Класс. Модификатор final.
37. Базовый Java. Ссылочные типы. Класс. Абстрактные методы.
38. Базовый Java. Ссылочные типы. Класс. Модификаторы native и synchronized.
39. Базовый Java. Ссылочные типы. Класс. Логические блоки.
40. Базовый Java. Ссылочные типы. Класс. Перегрузка методов.
41. Базовый Java. Ссылочные типы. Класс. Методы с переменным числом параметров.
42. Базовый Java. Ссылочные типы. Класс. Рекомендации по проектированию.
43. Базовый Java. Ссылочные типы. Перечисления.
44. Базовый Java. Ссылочные типы. Интерфейсы.
45. Базовый Java. Наследование и полиморфизм. Общее понятие.
46. Базовый Java. Наследование и полиморфизм. Позднее связывание.
47. Базовый Java. Наследование и полиморфизм. Ссылки super и this.
48. Базовый Java. Наследование и полиморфизм. Классы и методы final.
49. Базовый Java. Наследование и полиморфизм. Абстрактные классы.
50. Базовый Java. Наследование и полиморфизм. Класс Object.
51. Базовый Java. Наследование и полиморфизм. Клонирование.
52. Базовый Java. Наследование и полиморфизм. Рекомендации по проектированию иерархии классов.
53. Параметризация. Общее понятие.
54. Параметризация. Параметризация классов.
55. Параметризация. Джокеры (метасимволы).
56. Параметризация. Параметризация методов.
57. Параметризация. Параметризация интерфейсов.
58. Исключения. Общее понятие.
59. Исключения. Механизм обработки исключительных ситуаций.
60. Исключения. Иерархия исключений.
61. Исключения. Иерархия исключений. Ошибки.
62. Исключения. Проверяемые исключения.
63. Исключения. Непроверяемые исключения.
64. Исключения. Продвижение возбужденного исключения
65. Исключения. Способы обработки исключений. По месту возбуждения (пример)
66. Исключения. Способы обработки исключений. Обработка нескольких исключений
67. Исключения. Обработка исключений. Блок finally
68. Исключения. Возбуждение исключений пользователем. Оператор throw
69. Исключения. Обработка исключений и наследование
70. Исключения. Рекомендации по обработке исключений
71. Механизм утверждений (assertions).



72. Коллекции. Общие определения
73. Коллекции. Списки
74. Коллекции. Метасимвол в коллекциях
75. Коллекции. Интерфейс ListIterator
76. Коллекции. Интерфейс Comparator
77. Коллекции. Класс LinkedList и интерфейс Queue
78. Коллекции. Интерфейс Deque и класс ArrayDeque
79. Коллекции. Множества
80. Коллекции. Карты отображений
81. Коллекции. Унаследованные коллекции
82. Коллекции. Алгоритмы класса Collections

Задачи к экзаменационным билетам

Задача 1. На интервале $[2;n]$ найти натуральное число с максимальной суммой делителей.

Ввод

В первой строке ввода содержится число n ($2 \leq n \leq 1000$)

Вывод

В первой строке – максимальная сумма делителей.

Пример ввода

9

Пример вывода

15

Максимальная сумма делителей у числа 8 $1+2+4+8=15$.

Задача 2. Дана строка, изображающая арифметическое выражение вида " $\langle \text{цифра} \rangle \langle \text{знак} \rangle \langle \text{цифра} \rangle \langle \text{знак} \rangle \dots \langle \text{знак} \rangle \langle \text{цифра} \rangle$ ", где на месте знака операции находится символ "+" или "-" (например, "4+7-2-8"). Вывести значение данного выражения (целое число).

Ввод

В первой строке содержится последовательность символов, оканчивающаяся символом перехода на новую строку. Длина последовательности не превышает 256.

Вывод

Вывести одно целое число – значение выражения, записанного в исходной строке.

Пример ввода

1+2-3+4-5

Пример вывода

-1

Задача 3. Напишите программу, которая вычисляет значение по формуле:

$$\prod_{k=1}^n \left(1 + \frac{1}{k^k} \right)$$

В первой строке содержится натуральное число n ($1 \leq n \leq 100$).

Вывести результат с точностью $1e^{-6}$

Пример ввода

5



Пример вывода

2.603553

Задача 4. Напишите программу, которая вычисляет значение по формуле:

$$\frac{1}{a} + \frac{1}{a^2} + \frac{1}{a^4} + \dots + \frac{1}{a^{2^n}}$$

В первой строке содержатся натуральное число n ($1 \leq n \leq 10$) и действительное число a ($|a| < 5$, $a \neq 0$).

Вывести результат с точностью $10e-6$.

Пример ввода

10 -1.53

Пример вывода

-0.009509

Задача 5. Напишите программу, которая вычисляет значение выражения по формуле:

$$f(a) + 2f(a+h) + 2f(a+2h) + \dots + 2f(a+(n-1)h) + f(a+nh),$$

$$\text{где } f(x) = (x^2 + 1) \cdot \cos^2 x$$

В первой строке содержатся действительные числа a и h ($|a| < 25$, натуральное число n ($1 \leq n \leq 100$)).

Вывести результат с точностью $10e-6$.

Пример ввода

-14.568 0.5847 15

Пример вывода

1554.408554

Задача 6. Напишите программу, которая вычисляет значение по формуле:

$$\left(\frac{1}{\sin 1} \right) + \left(\frac{1}{\sin 1 + \sin 2} \right) + \dots + \left(\frac{1}{\sin 1 + \dots + \sin n} \right)$$

В первой строке содержится натуральное число n ($1 \leq n \leq 100$).

Вывести результат с точностью 10^{-6} .

Пример ввода

23

Пример вывода

3.403161

Задача 7. Пусть:



$$u_1 = u_2 = 0; \quad v_1 = v_2 = 1;$$

$$u_i = \frac{u_{i-1} - u_{i-2}v_{i-1} - v_{i-2}}{1 + u_{i-1}^2 + v_{i-1}^2}; \quad v_i = \frac{u_{i-1} - v_{i-1}}{|u_{i-2} + v_{i-1}| + 2}; \quad i = 3, 4, \dots$$

Дано натуральное n . Найти V_n .

В первой строке содержится целое число n ($3 \leq n \leq 50$).

Вывести результат с точностью $10e-6$.

Пример ввода

13

Пример вывода

-0.029583

Задача 8. Найти количество простых чисел, меньших n ($2 \leq n \leq 1000$).

Ввод

В первой строке ввода содержится число nn .

Вывод

Одно число – найденное количество простых чисел.

Пример ввода

8

Пример вывода

4

Задача 9. Дано число x ($0 < x < 1$). Среди всех простых несократимых дробей, заключённых между 0 и 1, знаменатель которых не превышает n ($2 \leq n \leq 1000$) найти дробь, наименее отличающуюся от x (дробь задаётся двумя натуральными числами – числителем и знаменателем).

Ввод

В первой строке ввода содержатся числа nn и xx .

Вывод

Числитель и знаменатель дроби.

Пример ввода

10 0.1234567

Пример вывода

1 8

Задача 10. Вычислить

$$\frac{(-1)^{\lfloor \lg 1 \rfloor}}{1} + \frac{(-1)^{\lfloor \lg 2 \rfloor}}{2} + \dots + \frac{(-1)^{\lfloor \lg n \rfloor}}{n}.$$

Ввод

В первой строке находится натуральное число n ($1 \leq n \leq 1000$).

Вывод

Вывести результат вычисления с шестью знаками после запятой.

Пример ввода



5

Пример вывода

2.283333

Задача 11. Дано целое число N и набор из N целых чисел, содержащий по крайней мере два нуля. Вывести сумму чисел из данного набора, расположенных между первым и последним нулем (если первый и последний нули идут подряд, то вывести 0).

Ввод

В первой строке содержится натуральное число N ($2 \leq N \leq 100$). Во второй строке находятся NN целых чисел, записанных через пробел.

Вывод

Вывести одно целое число – результат вычисления.

Пример ввода

6

1 0 2 3 4 0

Пример вывода

9

Задача 12. Пусть

$$a_i = \frac{i-1}{i+1} + \sin \frac{(i-1)^3}{i+1}$$

Найти число a_k такое, которое является наименьшим среди положительных из a_1, \dots, a_n .

Ввод

В первой строке содержится натуральное число n ($2 \leq n \leq 100$).

Вывод

Вывести найденное число с шестью знаками после запятой.

Пример ввода

5

Пример вывода

0.660528

Задача 13. Вычислите значение функции $f(x)$ в точке x , ($|x| < 1$) с помощью разложения ее в бесконечный ряд

$$\sum_{i=0}^{\infty} \frac{x^{2i+1}}{(2i+1)!}$$

с точностью $\text{eps}=10\text{e-}6$. Требуемая точность считается достигнутой, если найдена сумма некоторого количества первых слагаемых, а последующее слагаемое оказалось по модулю меньше, чем eps .

В первой строке ввода содержится вещественное число x ($|x| < 1$).

Вычислить результат с точностью $10\text{e-}6$.

Пример ввода



0.999999

Пример вывода

1.175200

Задача 14. Вычислите значение функции $f(x)$ в точке x , ($|x| < 1$) с помощью разложения ее в бесконечный ряд

$$\sum_{i=0}^{\infty} \frac{x^i}{i!}$$

с точностью $\text{eps}=10\text{e-}6$. Требуемая точность считается достигнутой, если найдена сумма некоторого количества первых слагаемых, а последующее слагаемое оказалось по модулю меньше, чем eps .

В первой строке ввода содержится вещественное число x ($|x| < 1$).

Вычислить результат с точностью $10\text{e-}6$.

Пример ввода

-0.1278

Пример вывода

0.880029

Задача 15. Ввести массив. Вывести значение, которое встречается в этом массиве чаще всего.

Первая строка ввода содержит одно целое число n ($1 \leq n \leq 100$) – количество элементов в массиве. Далее следует строка, содержащая n целых чисел в диапазоне от $-10\text{e-}9$ до $10\text{e-}9$ – элементы массива.

Вывести одно число – значение, которое встречается в этом массиве чаще всего. Если несколько значений встречаются одинаково часто, то вывести наибольшее из них.

Пример ввода

6

7 4 7 1 6 6

Пример вывода

7

Задача 16. Дана строка, содержащая полное имя файла, то есть имя диска, список каталогов (путь), собственно имя и расширение. Выделить из этой строки имя файла (без расширения).

Ввод

В первой строке находится исходная строка. Длина строки не превышает 256.

Вывод

Вывести имя файла в виде одной строки.

Пример ввода

C:\Documents\report.txt

Пример вывода

report



Задача 17. Группы символов, разделенные пробелами (одним или несколькими) и не содержащие пробелов внутри себя, будем называть словами. Известно, что исходная строка не содержит начальных и конечных пробелов. Подсчитать количество слов в данной строке.

Ввод

В первой строке содержится последовательность символов, оканчивающаяся символом перехода на новую строку. Длина последовательности не превышает 256.

Вывод

Вывести одно целое число – количество слов в строке.

Пример ввода

One, two, three.

Пример вывода

3

Задача 18. Напечатать в возрастающем порядке все трёхзначные числа, меньшие n ($103 \leq n \leq 1000$), цифры у которых различны. Операции / и % не использовать.

Ввод

В первой строке ввода содержится число nn .

Вывод

Найденные числа через пробел.

Пример ввода

110

Пример вывода

102 103 104 105 106 107 108 109

Задача 19. Дано натуральное число q ($0 < q \leq 1000$). Проверить, может ли оно являться периметром прямоугольного треугольника, стороны которого выражены натуральными числами.

Ввод

В первой строке ввода содержится число qq .

Вывод

Вывести сообщение "yes" – если может, "no" – в противном случае.

Пример ввода

7

Пример вывода

no

Задача 20. Дано натуральное число n . Вычислить:

$$\sum_{i=0}^n \sum_{j=i}^n \frac{(-1)^i j}{n}$$

Ввод

В первой строке ввода содержится число: n ($n \leq 1000$).

Вывод

В первой строке – результат вычислений с точностью до $10e-6$.

Пример ввода



3

Пример вывода

0.666667

Лабораторный практикум (разноплановые задачи)

Задание 1

0. Установить и настроить для использования JDK, NetBeans, MS VSCode, согласно инструкциям из папки java&progs

1. Задачи на отработку составления простых программ

Решить задачи из ipc.susu.ru

633, 549, 425, 489, 1102

2. Пакетная организация программы

Создать программу, печатающую звуки издаваемые животными зоопарка:

1. Классы животных Лев, Кошка, Утка, Ворона принадлежат пакетам zooapps.zoo.animals и zooapps.zoo.birds

- экспортируют всего один метод, возвращающий строку (с издаваемыми звуками)

2. Класс Зоопарк принадлежит пакету zooapps.zoo и экспортирует один метод, выводящий на консоль звуки, издаваемые животными

3. Наконец, класс Пускатель, принадлежащий пакету zooapps содержит точку входа в приложение и вызывает метод печати звуков всех животных

4. При программировании решения задачи необходимо следовать соглашению о наименованиях!!

5. Закомментируйте классы и методы с помощью javadoc комментариев и сгенерируйте по ним документацию (как для классов так и для пакетов)

.....

Крайне полезно решить эту задачу используя возможности, предоставляемые различными средами разработки (у нас это NetBeans, и VScode) =

то есть решать ее 2 раза (делая в различных средах независимо). Чтобы пакеты не пересекались, папки которые им соответствуют должны располагаться в различных корневых директориях.

Задание 2

Указание: решить как минимум 3 любых задания из ниже перечисленных, следуя рекомендациями по проектированию классов.

1. Создать классы, спецификации которых приведены ниже. Определить конструкторы и методы setТип(), getТип(), toString(). Определить дополнительно методы в классе, создающем массив объектов. Задать критерий выбора данных и вывести эти данные на консоль. В каждом классе, обладающем информацией, должно быть объявлено несколько конструкторов:

Train: Пункт назначения, Номер поезда, Время отправления, Число мест (общих, купе, плацкарт, люкс).

Создать массив объектов. Вывести:

а) список поездов, следующих до заданного пункта назначения;

б) список поездов, следующих до заданного пункта назначения и отправляющихся после заданного часа;



с) список поездов, отправляющихся до заданного пункта назначения и имеющих общие места.

2. Определить класс Точка на плоскости (в пространстве) и во времени.

Задать движение точки в определенном направлении. Создать методы по определению скорости и ускорения точки. Проверить для двух точек возможность пересечения траекторий. Определить расстояние между двумя точками в заданный момент времени.

3. Определить класс Вектор. Реализовать методы для вычисления модуля вектора, скалярного произведения, сложения, вычитания, умножения на константу. Объявить массив объектов. Написать метод, который для заданной пары векторов будет определять, являются ли они коллинеарными или ортогональными.

4. Определить класс Множество символов. Реализовать методы для определения принадлежности заданного элемента множеству; пересечения, объединения, разности двух множеств. Создать методы сложения, вычитания, умножения (пересечения), индексирования, присваивания. Создать массив объектов и передавать пары объектов в метод другого класса, который строит множество, состоящее из элементов, входящих только в одно из заданных множеств.

Задание 3

Создать приложение, удовлетворяющее требованиям, приведенным в задании.

Наследование применять только в тех заданиях, в которых это логически обосновано. Аргументировать принадлежность классу каждого создаваемого метода и корректно переопределить для каждого класса методы equals(), hashCode(), toString().

1. Создать объект класса Текст, используя классы Предложение, Слово.

Методы: дополнить текст, вывести на консоль текст, заголовок текста.

2. Создать объект класса Автомобиль, используя классы Колесо, Двигатель.

Методы: ехать, заправляться, менять колесо, вывести на консоль марку автомобиля.

3. Создать объект класса Наседка, используя классы Птица, Кукушка.

Методы: летать, петь, нести яйца, высиживать птенцов.

4. Создать объект класса Текстовый файл, используя классы Файл, Директория. Методы: создать, переименовать, вывести на консоль содержимое, дополнить, удалить.

Задание 4

Использование коллекций Java

1. Ввести строки из файла, записать в список. Вывести строки в файл в обратном порядке.

2. Реализовать класс Graph, представляющий собой неориентированный граф. В конструкторе класса передается количество вершин в графе.

Методы должны поддерживать быстрое добавление и удаление ребер.

3. Задан файл с текстом на английском языке. Выделить все различные слова.

Слова, отличающиеся только регистром букв, считать одинаковыми.

Использовать класс HashSet.

4. Не используя вспомогательных объектов, переставить отрицательные элементы данного списка в конец, а положительные — в начало списка.

Задание 5



1. Определить классы java для вычисления правой и центральной производной произвольной функции одного переменного в заданной точке (для которой эта производная существует в этой точке), с заданной точностью.

Основная программа ДОЛЖНА включать примеры использования этих методов, в качестве функций сделать определить статических метод класса, анонимный класс, и лямбда выражение(функцию)

2. Самостоятельно разработать интерфейс для реализации методов численного интегрирования функций одного переменного в заданном интервале:

На основе этого интерфейса реализовать классы с методами для численного интегрирования Монте-Карло, Прямоугольников, Трапеций.... Методика проверки (примеров) работоспособности аналогично предыдущему заданию.

Теорию описывающую данные методы брать из лекций или ВИКИПЕДИИ - там описываются конечные формулы, описывается последовательность действий и приводятся примеры.

3. Предусмотреть возбуждение и обработку следующих исключительных ситуациях:

В методах численного дифференцирования -

1. Считать ошибочной ситуацию, когда погрешность БОЛЬШЕ чем 0.1% от значения x0

2. Считать ошибочной ситуацию, 5 последних последовательных приближений ОТСУТСТВУЕТ сходимости к искомому значению (разность по модулю между текущим и предыдущим приближением НЕ УМЕНЬШАЕТСЯ в течении 5 последних итераций).

В методах численного интегрирования -

1. Считать ошибкой случай когда верхний предел интегрирования МЕНЬШЕ ЛИБО РАВЕН нижнему

2. Считать ошибочной ситуацию, 5 последних последовательных приближений ОТСУТСТВУЕТ сходимости к искомому значению (разность по модулю между текущим и предыдущим приближением НЕ УМЕНЬШАЕТСЯ в течении 10 последних итераций).

Определить по крайней мере для одного из методов дифференцирования и интегрирования СВОИ классы ПРОВЕРЯЕМЫХ исключений.

Предков для них выбрать исходя из СВОЕГО понимания задачи (и рекомендаций языка Джава), этот выбор ОБОСНОВАТЬ.

Обязательно! одну исключительную ситуацию обработать внутри численного метода, и еще одну дать возможность обработать ВЫШЕ по цепочке вызовов (методу которой вызывает разработанные вами подпрограммы)

Обязательно! Привести КОНКРЕТНЫЕ примеры в которых возникают вышеперечисленные исключительные ситуации.

Соглашение о наименованиях языка Java

- При выборе имени класса, поля, метода использовать цельные слова, полностью исключить сокращения. По возможности опускать предлоги и очевидные связующие слова. Аббревиатуры использовать только в том случае, когда они очевидны.

- Имя класса всегда пишется с большой буквы: **Coin, Developer**.

- Если имя класса состоит из двух и более слов, то второе и следующие слова пишутся слитно с предыдущим и начинаются с большой буквы: **AncientCoin, FrontendDeveloper**.

- Имя метода всегда пишется с маленькой буквы: **perform(), execute()**.



- Если имя метода состоит из двух и более слов, то второе и следующие слова пишутся слитно с предыдущим и начинаются с большой буквы: **performTask()**, **executeBaseAction()**.
- Имя поля класса, локальной переменной и параметра метода всегда пишутся с маленькой буквы: **weight**, **price**.
- Если имя поля класса, локальной переменной и параметра метода состоит из двух и более слов, то второе и следующие слова пишутся слитно с предыдущим и начинаются с большой буквы: **priceTicket**, **typeProject**.
- Константы и перечисления пишутся в верхнем регистре: **DISCOUNT**, **MAX_RANGE**.
- Все имена пакетов пишутся с маленькой буквы. Сокращения допустимы только в случае, если имя пакета слишком длинное: 10 или более символов. Использование цифр и других символов нежелательно.

Темы для индивидуальных проектов

1. Файловая система. В БД хранится информация о дереве каталогов файловой системы — каталоги, подкаталоги, файлы.

Для каталогов необходимо хранить:

- родительский каталог;
- название.

Для файлов необходимо хранить:

- родительский каталог;
- название;
- место, занимаемое на диске.

- Определить полный путь заданного файла (каталога).
- Подсчитать количество файлов в заданном каталоге, включая вложенные файлы и каталоги.
- Подсчитать место, занимаемое на диске содержимым заданного каталога.
- Найти в базе файлы по заданной маске с выдачей полного пути.
- Переместить файлы и подкаталоги из одного каталога в другой.
- Удалить файлы и каталоги заданного каталога.

2. Видеотека. В БД хранится информация о домашней видеотеке: фильмы, актеры, режиссеры.

Для фильмов необходимо хранить:

- название;
- имена актеров;
- дату выхода;
- страну, в которой выпущен фильм.

Для актеров и режиссеров необходимо хранить:

- ФИО;
- дату рождения.
- Найти все фильмы, вышедшие на экран в текущем и прошлом году.
- Вывести информацию об актерах, снимавшихся в заданном фильме.
- Вывести информацию об актерах, снимавшихся как минимум в N фильмах.
- Вывести информацию об актерах, которые были режиссерами хотя бы одного из фильмов.
- Удалить все фильмы, дата выхода которых была более заданного числа лет назад.

3. Расписание занятий. В БД хранится информация о преподавателях и проводимых ими занятиях.

Для предметов необходимо хранить:

- название;
- время проведения (день недели);



— аудитории, в которых проводятся занятия.

Для преподавателей необходимо хранить:

- ФИО;
- предметы, которые он ведет;
- количество пар в неделю по каждому предмету;
- количество студентов, занимающихся на каждой паре. JDBC

• Вывести информацию о преподавателях, работающих в заданный день недели в заданной аудитории.

- Вывести информацию о преподавателях, которые не ведут занятия в заданный день недели.
- Вывести дни недели, в которых проводится заданное количество занятий.
- Вывести дни недели, в которых занято заданное количество аудиторий.
- Перенести первые занятия заданных дней недели на последнее место.

4. Письма. В БД хранится информация о письмах и отправляющих их людях.

Для людей необходимо хранить:

- ФИО;
- дату рождения.

Для писем необходимо хранить:

- отправителя;
- получателя;
- тему письма;
- текст письма;
- дату отправки.
- Найти пользователя, длина писем которого наименьшая.
- Вывести информацию о пользователях, а также количестве полученных и отправленных ими письмах.

• Вывести информацию о пользователях, которые получили хотя бы одно сообщение с заданной темой.

- Вывести информацию о пользователях, которые не получали сообщения с заданной темой.
- Направить письмо заданного человека с заданной темой всем адресатам.

5. Сувениры. В БД хранится информация о сувенирах и их производителях.

Для сувениров необходимо хранить:

- название;
- реквизиты производителя;
- дату выпуска;
- цену.

Для производителей необходимо хранить:

- название;
- страну.
- Вывести информацию о сувенирах заданного производителя.
- Вывести информацию о сувенирах, произведенных в заданной стране.
- Вывести информацию о производителях, чьи цены на сувениры меньше заданной.
- Вывести информацию о производителях заданного сувенира, произведенного в заданном

году.

- Удалить заданного производителя и его сувениры.

6. Заказ. В БД хранится информация о заказах магазина и товарах в них.

Для заказа необходимо хранить:

- номер заказа;



- товары в заказе;
- дату поступления.

Для товаров в заказе необходимо хранить:

- товар;
- количество.

Для товара необходимо хранить:

- название;
- описание;
- цену.

- Вывести полную информацию о заданном заказе.

• Вывести номера заказов, сумма которых не превосходит заданную и количество различных товаров равно заданному.

- Вывести номера заказов, содержащих заданный товар.

• Вывести номера заказов, не содержащих заданный товар и поступивших в течение текущего дня.

- Сформировать новый заказ, состоящий из товаров, заказанных в текущий день.

- Удалить все заказы, в которых присутствует заданное количество заданного товара.

7. Продукция. В БД хранится информация о продукции компании.

Для продукции необходимо хранить:

- название;
- группу продукции (телефоны, телевизоры и др.);
- описание;
- дату выпуска;
- значения параметров.

Для групп продукции необходимо хранить:

- название;
- перечень групп параметров (размеры и др.).

Для групп параметров необходимо хранить:

- название;
- перечень параметров.

Для параметров необходимо хранить:

- название;
- единицу измерения.

- Вывести перечень параметров для заданной группы продукции.

- Вывести перечень продукции, не содержащий заданного параметра.

- Вывести информацию о продукции для заданной группы.

- Вывести информацию о продукции и всех ее параметрах со значениями JDBC

- Удалить из базы продукцию, содержащую заданные параметры.

- Переместить группу параметров из одной группы товаров в другую.

8. Погода. В БД хранится информация о погоде в различных регионах.

Для погоды необходимо хранить:

- регион;
- дату;
- температуру;
- осадки.

Для регионов необходимо хранить:

- название;



- площадь;
- тип жителей.

Для типов жителей необходимо хранить:

- название;
- язык общения.

- Вывести сведения о погоде в заданном регионе.
- Вывести даты, когда в заданном регионе шел снег и температура была ниже заданной отрицательной.
- Вывести информацию о погоде за прошедшую неделю в регионах, жители которых общаются на заданном языке.
- Вывести среднюю температуру за прошедшую неделю в регионах с площадью больше заданной.

9. Магазин часов. В БД хранится информация о часах, продающихся в магазине.

Для часов необходимо хранить:

- марку;
- тип (кварцевые, механические);
- цену;
- количество;
- реквизиты производителя.

Для производителей необходимо хранить:

- название;
- страна.
- Вывести марки заданного типа часов.
- Вывести информацию о механических часах, цена на которые не превышает заданную.
- Вывести марки часов, изготовленных в заданной стране.
- Вывести производителей, общая сумма часов которых в магазине не превышает заданную.

10. Города. В БД хранится информация о городах и их жителях.

Для городов необходимо хранить:

- название;
- год основания;
- площадь;
- количество населения для каждого типа жителей.

Для типов жителей необходимо хранить:

- город проживания;
- название;
- язык общения.
- Вывести информацию обо всех жителях заданного города, разговаривающих на заданном языке.
- Вывести информацию обо всех городах, в которых проживают жители выбранного типа.
- Вывести информацию о городе с заданным количеством населения и всех типах жителей, в нем проживающих.
- Вывести информацию о самом древнем типе жителей.

11. Планеты. В БД хранится информация о планетах, их спутниках и галактиках.

Для планет необходимо хранить:

- название;
- радиус;
- температуру ядра;



- наличие атмосферы;
- наличие жизни;
- спутники.

Для спутников необходимо хранить:

- название;
- радиус;
- расстояние до планеты.

Для галактик необходимо хранить:

- название;
- планеты.

• Вывести информацию обо всех планетах, на которых присутствует жизнь, и их спутниках в заданной галактике.

• Вывести информацию о планетах и их спутниках, имеющих наименьший радиус и наибольшее количество спутников.

• Вывести информацию о планете, галактике, в которой она находится, и ее спутниках, имеющей максимальное количество спутников, но с наименьшим общим объемом этих спутников.

• Найти галактику, сумма ядерных температур планет которой наибольшая.

12. Точки. В БД хранится некоторое конечное множество точек с их координатами.

• Вывести точку из множества, наиболее приближенную к заданной.

• Вывести точку из множества, наиболее удаленную от заданной.

• Вывести точки из множества, лежащие на одной прямой с заданной прямой.

13. Треугольники. В БД хранятся треугольники и координаты их точек на плоскости.

• Вывести треугольник, площадь которого наиболее приближена к заданной.

• Вывести треугольники, сумма площадей которых наиболее приближена к заданной.

• Вывести треугольники, которые помещаются в окружность заданного радиуса.

14. Словарь. В БД хранится англо-русский словарь, в котором для одного английского слова может быть указано несколько его значений и наоборот. Со стороны клиента вводятся последовательно английские (русские) слова. Для каждого из них вывести на консоль все русские (английские) значения слова.

15. Словари. В двух различных базах данных хранятся два словаря: русско-белорусский и белорусско-русский. Клиент вводит слово и выбирает язык.

Вывести перевод этого слова.

16. Стихотворения. В БД хранятся несколько стихотворений с указанием автора и года создания. Для хранения стихотворений использовать объекты типа `Blob`. Клиент выбирает автора и критерий поиска.

• В каком из стихотворений больше всего восклицательных предложений?

• В каком из стихотворений меньше всего повествовательных предложений?

• Есть ли среди стихотворений сонеты и сколько их?

17. Четырехугольники. В БД хранятся координаты вершин выпуклых четырехугольников на плоскости.

• Вывести координаты вершин параллелограммов.

• Вывести координаты вершин трапеций.

18. Треугольники. В БД хранятся координаты вершин треугольников на плоскости.

• Вывести все равнобедренные треугольники.

• Вывести все равносторонние треугольники.

• Вывести все прямоугольные треугольники.

• Вывести все тупоугольные треугольники с площадью больше заданной.



4. Порядок проведения и критерии оценивания промежуточной аттестации

4.1. Порядок проведения промежуточной аттестации

Схема подведения итогов промежуточной аттестации в форме экзамена:

- Студент выбирает билет, содержащий 2 теоретических вопроса и практическое задание;
- Решает практическое задание, отправляет решение на проверку автоматизированной проверочной системой, получает результат;
- Письменно отвечает на теоретические вопросы.
- Используя записи и результат проверки практического задания «сдает» экзаменационный билет преподавателю.

4.2. Критерии оценивания промежуточной аттестации по видам оценочных средств

Баллы, полученные за отдельные задания экзамена (теоретический вопрос, практическое задание) суммируются.

Теоретический вопрос 1 – 0-30 б.

Теоретический вопрос 2 – 0-30 б.

Практическое задание – 0-40 б.

Всего: 0-100 б.

При подведении итогов промежуточной аттестации учитываются результаты текущей аттестации.

Лабораторный практикум – 0-50 б.

Проектное задание – 0-50 б.

4.3. Результаты промежуточной аттестации и уровни сформированности компетенций

При подведении итогов промежуточной аттестации учитываются результаты текущего контроля.

0-49 баллов - неудовлетворительно (2);

50-69 баллов - удовлетворительно (3);

70-89 баллов - хорошо (4);

90-100 баллов - отлично (5).

Полученные за текущую аттестацию баллы могут либо повысить, либо понизить отметку за экзамен за счет вычисления среднего балла за экзамен и текущую аттестацию.

Особенности проведения процедуры оценивания результатов обучения инвалидов и лиц с ограниченными возможностями здоровья обозначены в рабочей программе дисциплины (модуля).

Уровни сформированности компетенций определяется следующим образом:

1. Продвинутый уровень сформированности компетенций соответствует оценке «отлично»:

Обучающийся владеет знаниями предмета в полном объеме учебной программы, достаточно глубоко осмысливает дисциплину; самостоятельно, в логической последовательности и исчерпывающе отвечает на все вопросы, подчеркивает при этом самое существенное, умеет анализировать, сравнивать, классифицировать, обобщать, конкретизировать и систематизировать изученный материал, выделять в нем главное; устанавливать причинно-следственные связи; четко формирует ответы.

2. Базовый уровень соответствует оценке «хорошо»:



Обучающийся владеет знаниями дисциплины почти в полном объеме программы (имеются пробелы знаний только в некоторых, особенно сложных разделах); самостоятельно и отчасти при наводящих вопросах дает полноценные ответы на вопросы; не всегда выделяет наиболее существенное, не допускает вместе с тем серьезных ошибок в ответах.

3. Пороговый уровень соответствует оценке «удовлетворительно»:

Обучающийся владеет основным объемом знаний по дисциплине; проявляет затруднения в самостоятельных ответах, оперирует неточными формулировками; в процессе ответов допускает ошибки по существу вопросов.

4. Низкий уровень соответствует оценке «неудовлетворительно»:

Обучающийся не освоил обязательного минимума знаний предмета, не способен ответить на вопросы даже при дополнительных наводящих вопросах экзаменатора.

