



**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ**
федеральное государственное бюджетное образовательное учреждение
высшего образования
«ИРКУТСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

Институт математики и информационных технологий
Кафедра информационных технологий



Рабочая программа дисциплины (модуля)

Б1.О.12 Информатика

Направление подготовки информационные технологии	02.03.02	Фундаментальная	информатика	и
Направленность (профиль) подготовки программная инженерия		Фундаментальная	информатика	и
Квалификация выпускника		бакалавр		
Форма обучения		очная		

Иркутск 2026 г.

ЦЕЛИ И ЗАДАЧИ ДИСЦИПЛИНЫ

Цель

Формирование у студентов базовых знаний в области информатики и программирования.

Задачи

Формирование знаний, умений и навыков студента по разделам «Введение в программирование», «Представление и обработка чисел в компьютере», формирование базовых практических умений и навыков программирования на языке Java.

МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОПОП ВО

2.1. Учебная дисциплина (модуль) относится к обязательной части программы и изучается на первом курсе.

2.2. Для изучения данной учебной дисциплины (модуля) необходимы знания, умения и навыки, сформированные: не предусмотрены.

2.3. Перечень последующих учебных дисциплин, для которых необходимы знания, умения и навыки, формируемые данной учебной дисциплиной: Веб-технологии, Программирование, Введение в информационный поиск, Проектирование информационных систем.

ТРЕБОВАНИЯ К РЕЗУЛЬТАТАМ ОСВОЕНИЯ ДИСЦИПЛИНЫ

Процесс освоения дисциплины направлен на формирование компетенций (элементов следующих компетенций) в соответствии с ФГОС ВО по соответствующему направлению подготовки.

Перечень планируемых результатов обучения по дисциплине (модулю), соотнесенных с индикаторами достижения компетенций

Компетенция	Индикаторы компетенций	Результаты обучения
УК-1 Способен осуществлять поиск, критический анализ и синтез информации, применять системный подход для решения поставленных задач	ИДК УК1.1 Анализирует задачу, выделяя ее базовые составляющие	Знает принципы анализа поставленной задачи Умеет формализовать поставленную задачу Владеет навыками анализа поставленной задачи
	ИДК УК1.2 Определяет и ранжирует информацию, требуемую для решения поставленной задачи	Знает источники получения официальной документации по языку программирования Java Умеет пользоваться документацией Владеет принципами работы с документацией
	ИДК УК1.3 Осуществляет поиск информации для решения поставленной задачи по различным типам запросов	Знает источники получения официальной документации по языку программирования Java Умеет пользоваться документацией

		Владеет принципами работы с документацией
	ИДК УК1.4 При обработке информации отличает факты от мнений, интерпретаций, оценок, формирует собственные мнения и суждения, аргументирует свои выводы, в том числе с применением философского понятийного аппарата	Знает площадки публикации наиболее часто возникающих вопросов, связанных с программированием. Умеет искать нужную информацию на этих площадках Владеет способностью оценивать полученные сведения и применять в своей работе
ОПК-3 Способен к разработке алгоритмических и программных решений в области системного и прикладного программирования, математических, информационных и имитационных моделей, созданию информационных ресурсов глобальных сетей, образовательного контента, прикладных баз данных, тестов и средств тестирования систем и средств на соответствие стандартам и исходным требованиям	ИДК ОПК3.1 Знает основные языки программирования и типы баз данных, операционные системы и оболочки, современные программные среды разработки информационных систем и технологий	Знает принципы структурного и модульного программирования, основные приемы алгоритмизации и программирования на языке высокого уровня, принципы разработки программ, принципы автономной отладки и тестирования простых программ, базовые конструкции языка программирования Java, способы представления примитивных типов данных и алгоритмы их обработки Умеет формализовать поставленную задачу, составлять и оформлять программы на языке программирования Java, тестировать и отлаживать программы в современных интегрированных средах разработки, применять полученные знания к различным предметным областям. Владеет навыками разработки, отладки и тестирования программ в интегрированной среде разработки.
	ИДК ОПК3.2 Применяет языки программирования и современные программные среды разработки информационных систем и технологий для автоматизации бизнес-процессов, решения прикладных задач раз-	Знает принципы структурного и модульного программирования, основные приемы алгоритмизации и программирования на языке высокого уровня, принципы разработки программ, принципы автономной отладки и тестирования простых программ, базовые конструкции

	<p>личных классов, создания информационных ресурсов глобальных сетей, ведения баз данных и информационных хранилищ</p>	<p>языка программирования Java, способы представления примитивных типов данных и алгоритмы их обработки Умеет формализовать поставленную задачу, составлять и оформлять программы на языке программирования Java, тестировать и отлаживать программы в современных интегрированных средах разработки, применять полученные знания к различным предметным областям. Владеет навыками разработки, отладки и тестирования программ в интегрированной среде разработки.</p>
	<p>ИДК опкз.3 Способен выполнять задачи программирования, отладки и тестирования прототипов программных средств и информационных систем</p>	<p>Знает принципы структурного и модульного программирования, основные приемы алгоритмизации и программирования на языке высокого уровня, принципы разработки программ, принципы автономной отладки и тестирования простых программ, базовые конструкции языка программирования Java, способы представления примитивных типов данных и алгоритмы их обработки. Умеет формализовать поставленную задачу, составлять и оформлять программы на языке программирования Java, тестировать и отлаживать программы в современных интегрированных средах разработки, применять полученные знания к различным предметным областям. Владеет навыками разработки, отладки и тестирования программ в интегрированной среде разработки.</p>
<p>УК-2 Способен определять круг задач в рамках поставленной цели и выбирать оптимальные способы их решения, исходя из действующих</p>	<p>ИДК УК2.1 Формулирует цели, задач, значимости, ожидаемых результатов</p>	<p>Знает способы описания ожидаемых результатов программы Умеет формулировать ожи-</p>

правовых норм, имеющихся ресурсов и ограничений	проекта	даемые результаты Владеет навыками формулирования ожидаемых результатов
	ИДК УК2.2 В рамках поставленных задач определяет имеющиеся ресурсы и ограничения, действующие правовые нормы	Знает способы взаимодействия с файловой системой Умеет реализовывать чтение данных из файлов Владеет навыками реализации чтения данных из файлов
	ИДК УК2.3 Разрабатывает план реализации проекта	Знает принципы, используемые в ИОС DOMIC для указания времени окончания проекта Умеет распределять время на выполнения учебных заданий Владеет навыками планирования выполнения учебных заданий
	ИДК УК2.4 Осуществляет контроль реализации проекта	Знает способы проверки правильности выполненных задач Умеет проверять выполненные задачи на соответствие формулировке Владеет навыками тестирования написанных программ
	ИДК УК2.5 Проводит оценку эффективности реализации проекта и разработку плана действий по его корректировке	Знает принципы оценки эффективности написанной программы Умеет оценивать эффективность реализованной программы Владеет навыками оценки эффективности реализованной программы
ОПК-2 Способен применять компьютерные/суперкомпьютерные методы, современное программное обеспечение, в том числе отечественного происхождения, для решения задач профессиональной деятельности	ИДК ОПК2.1 Понимает базовые принципы и устройство современных информационных технологий и программных средств	Знает принципы структурного и модульного программирования, основные приемы алгоритмизации и программирования на языке высокого уровня, принципы разработки программ, принципы автономной отладки и тестирования простых программ, базовые конструкции языка программирования Java, способы представления примитивных типов данных и алгоритмы их обработки.

		<p>Умеет формализовать поставленную задачу, составлять и оформлять программы на языке программирования Java, тестировать и отлаживать программы в современных интегрированных средах разработки, применять полученные знания к различным предметным областям.</p> <p>Владеет навыками разработки, отладки и тестирования программ в интегрированной среде разработки.</p>
	<p>ИДК опк2.2 Способен применять современное программное обеспечение (в том числе отечественного производства) для решения задач профессиональной деятельности</p>	<p>Знает принципы структурного и модульного программирования, основные приемы алгоритмизации и программирования на языке высокого уровня, принципы разработки программ, принципы автономной отладки и тестирования простых программ, базовые конструкции языка программирования Java, способы представления примитивных типов данных и алгоритмы их обработки.</p> <p>Умеет формализовать поставленную задачу, составлять и оформлять программы на языке программирования Java, тестировать и отлаживать программы в современных интегрированных средах разработки, применять полученные знания к различным предметным областям.</p> <p>Владеет навыками разработки, отладки и тестирования программ в интегрированной среде разработки.</p>
	<p>ИДК опк2.3 Способен применять суперкомпьютерные методы для решения задач профессиональной деятельности</p>	<p>Знает принципы структурного и модульного программирования, основные приемы алгоритмизации и программирования на языке высокого уровня, принципы разработки программ, принципы автономной отладки и</p>

		<p>тестирования простых программ, базовые конструкции языка программирования Java, способы представления примитивных типов данных и алгоритмы их обработки. Умеет формализовать поставленную задачу, составлять и оформлять программы на языке программирования Java, тестировать и отлаживать программы в современных интегрированных средах разработки, применять полученные знания к различным предметным областям.</p> <p>Владеет навыками разработки, отладки и тестирования программ в интегрированной среде разработки.</p>
--	--	--

СОДЕРЖАНИЕ И СТРУКТУРА ДИСЦИПЛИНЫ

Объем дисциплины составляет 5 зачетных единиц, 180 часов, в том числе 35 часов на контроль, практическая подготовка 180.
 Форма промежуточной аттестации: 1 семестр – экзамен.

4.1. СОДЕРЖАНИЕ ДИСЦИПЛИНЫ, СТРУКТУРИРОВАННОЕ ПО ТЕМАМ, С УКАЗАНИЕМ ВИДОВ УЧЕБНЫХ ЗАНЯТИЙ И ОТВЕДЕННОГО НА НИХ КОЛИЧЕСТВА АКАДЕМИЧЕСКИХ ЧАСОВ

№ п/п	Раздел дисциплины/темы	Семестр	Виды учебной работы, включая самостоятельную работу обучающихся и трудоемкость (в часах)				Формы текущего контроля успеваемости
			Контактная работа преподавателя с обучающимися			Самостоятельная работа + контроль	
			Лекции	Семинарские (практические занятия)	Контроль обучения		
1	Алгоритмы и программы	1	2			6,5	
2	Знакомство с Java. Среда разработки	1	2	4	0,5	6,5	лаб.
3	Форматный вывод	1	1	2	0,5	6,5	лаб.
4	Целые типы данных	1	2	2	0,5	6,5	лаб.
5	Типы данных с плавающей точкой. Методы класса Math	1	2	2	0,5	6,5	лаб.
6	Логический тип данных	1	1	2	0,5	6,5	лаб.
7	Условный оператор	1	2	2	0,5	6,5	лаб.
8	Оператор выбора	1	1	2	0,5	10,5	лаб.
9	Циклические конструкции. Операторы перехода	1	6	4	0,5	8,5	лаб.
10	Статические методы	1	3	4	1	8,5	лаб.
11	Работа с файлами	1	1	2	1	6,5	лаб.
12	Ошибки и отладка	1	1		0,5	10,5	лаб.
13	Представление числовых данных и их обработка	1	6	4	2	5,5	контр.

14	Преобразование примитивных типов	1	1		0,5	6,5	
15	Битовые операции	1	1	4	1	6,5	лаб.
16	Одномерные массивы	1	2			6,5	
Итого часов		Од но мер ны е мас си вы	34	34	10	102	

4.2. ПЛАН ВНЕАУДИТОРНОЙ САМОСТОЯТЕЛЬНОЙ РАБОТЫ ОБУЧАЮЩИХСЯ ПО ДИСЦИПЛИНЕ

Се- мestr	Название раздела, темы	Самостоятельная работа обучающихся			Оценочное сред- ство	Учебно-методическое обеспечение самостоятельной работы
		Вид самостоятель- ной работы	Сроки выпол- нения	Затраты времени (час.)		
1	Тема 2. Знакомство с Java. Среда разработки	Выполнение прак- тической работы	30.09	6,5	Проверка домаш- ней работы	Материалы курса на платформе ИОС DOMIC
1	Тема 3. Форматный вывод	Выполнение прак- тической работы	07.10	6,5	Проверка домаш- ней работы	Материалы курса на платформе ИОС DOMIC
1	Тема 4. Целые типы данных	Выполнение прак- тической работы	07.10	6,5	Проверка домаш- ней работы	Материалы курса на платформе ИОС DOMIC
1	Тема 5. Типы данных с плавающей точкой. Методы класса Math	Выполнение прак- тической работы	14.10	6,5	Проверка домаш- ней работы	Материалы курса на платформе ИОС DOMIC
1	Тема 6. Логический тип данных	Выполнение прак- тической работы	21.10	6,5	Домашняя работа	Литература из п. 5 и материалы курса на платформе ИОС DOMIC
1	Тема 7. Условный оператор	Выполнение прак- тической работы	28.10	6,5	Домашняя работа	Литература из п. 5 и материалы курса на платформе ИОС DOMIC
1	Тема 8. Оператор выбора	Выполнение прак- тической работы	04.11	6,5	Домашняя работа	Литература из п. 5 и материалы курса на платформе ИОС DOMIC

Се- местр	Название раздела, темы	Самостоятельная работа обучающихся			Оценочное сред- ство	Учебно-методическое обеспечение самостоятельной работы
		Вид самостоятель- ной работы	Сроки выпол- нения	Затраты времени (час.)		
1	Тема 9. Циклические конструкции. Операторы перехода	Выполнение прак- тической работы	25.11	10,5	Домашняя работа	Литература из п. 5 и материалы курса на платформе ИОС DOMIC
1	Тема 10. Статические методы	Выполнение прак- тической работы	02.12	8,5	Домашняя работа	Литература из п. 5 и материалы курса на платформе ИОС DOMIC
1	Тема 11. Работа с файлами	Выполнение прак- тической работы	09.12	8,5	Домашняя работа	Литература из п. 5 и материалы курса на платформе ИОС DOMIC
1	Тема 12. Ошибки и отладка	Выполнение прак- тической работы	16.12	6,5	Домашняя работа	Литература из п. 5 и материалы курса на платформе ИОС DOMIC
1	Тема 13. Представление числовых данных и их обработка	Работа с книгой	16.12	10,5	Домашняя работа	Литература из п. 5 и материалы курса на платформе ИОС DOMIC
1	Тема 14. Преобразование прими- тивных типов	Выполнение прак- тической работы	26.12	5,5	Домашняя работа	Литература из п. 5 и материалы курса на платформе ИОС DOMIC
1	Тема 15. Битовые операции	Выполнение прак- тической работы	26.12	6,5	Домашняя работа	Литература из п. 5 и материалы курса на платформе ИОС DOMIC
Общая трудоемкость самостоятельной работы по дисциплине (час)				102		
Из них объем самостоятельной работы с использованием электронного обучения и дистанционных образовательных технологий (час)				102		

4.3. СОДЕРЖАНИЕ УЧЕБНОГО МАТЕРИАЛА

1. Тема 1. Алгоритмы и программы

Алгоритм, свойства алгоритма, программа, команда, способы записи алгоритма, основные качества программ, правила написания программ-ного кода, этапы разработки программы.

Тема 2. Знакомство с языком программирования Java. Среда разработки

Язык Java, байт-код, виртуальная машина Java, простейшая программа на Java, компиляция и выполнение программы, оператор вывод, пе-ременная, типы переменных, имена переменных, ключевые и зарезервированные слова Java, объявление переменной, оператор присваива-ния, сокращенная запись арифметических операций, ручная прокрутка программы, порядок выполнения операций, ввод данных с клавиату-ры. JDK, IDE – интегрированная среда разработки, NetBeans IDE, установка, создание приложения, создание класса, компиляция и запуск, отладка программы, методы отладки.

Тема 3. Форматный вывод

Форматный вывод, форматная строка, спецификаторы формата, форматирование целых чисел, форматирование вещественных чисел, спецификатор минимальной ширины поля, указание точности, флаги формата, выравнивание вывода, использование индекса аргументов.

Тема 4. Целые типы данных

Целые типы данных, объявление переменных целого типа, операции над величинами целого типа, особенности деления в Java, определение цифр числа, целочисленные константы.

Тема 5. Вещественные типы данных. Методы класса Math

Типы с плавающей точкой, константы с плавающей точкой, класс Math, константы класса Math, прямые трансцендентные функции, обратные трансцендентные функции, гиперболические функции, экспоненциальные функции, функции округления, прочие функции, псевдослучайные числа,

Тема 6. Логический тип данных

Логический тип данных, логические операции, порядок выполнения

Тема 7. Условный оператор.

Разветвляющиеся алгоритмы, условный оператор, сложные условия.

Тема 8. Оператор выбора

Оператор выбора, выполнение оператора выбора, вложенные операторы выбора.

Тема 9. Циклические конструкции. Операторы перехода

Цикл, цикл с известным количеством шагов, цикл с уменьшением циклической переменной, изменение шага циклической переменной, цикл с условием, цикл с постусловием, операторы break, continue, return. Применение циклов.

Тема 10. Статические методы

Статические методы, возвращающие и не возвращающие значение, сигнатура метода, описание метода, глобальные и локальные переменные, формальные и фактические параметры, особенности методов, возвращающих значение, логические методы.

Тема 11. Работа с файлами

Класс Scanner, основы сканирования, примеры чтения данных, установка разделителей, класс FileWriter.

Тема 12. Ошибки и отладка

Виды ошибок, средства отладки программы.

Тема 13. Представление числовых данных и их обработка

Системы счисления, перевод чисел из одной системы счисления в другую, перевод дробных чисел из одной системы счисления в другую, понятие экономичности системы счисления, перевод чисел между системами счисления 2-8-16, преобразование нормализованных чисел.

Кодирование и обработка в компьютере целых чисел без знака, кодирование и обработка в компьютере целых чисел со знаком, кодирование и обработка в компьютере вещественных чисел.

Тема 14. Преобразование примитивных типов

Строго типизированный язык, преобразование типов, неявное преобразование типов, явное преобразование типов, виды приведений, тождественное преобразование, преобразование примитивных типов, расширение простого типа, искажения при расширениях, сужение простого типа, сужение дробного типа до целочисленного.

Тема 15. Битовые операции

Битовые операции, побитовые операции в Java, побитовые логические операции, инверсия, операция &, маска, обнуление битов, проверка битов, операция |, установка битов в 1, операция ^, инверсия битов, шифровка, битовые сдвиги, логический сдвиг, циклически сдвиг, арифметический сдвиг, автоматическое повышение типов.

Тема 16. Одномерные массивы

Одномерный массив, объявление массива, ввод с клавиатуры, поэлементные операции, максимальный элемент массива, заполнение массива случайными числами, реверс массива, циклические сдвиг.

4.3.1. Перечень семинарских, практических занятий и лабораторных работ

№ п/н	№ раздела и темы	Наименование семинаров, практических и лабораторных работ	Трудоемкость (час.)		Оценочные средства	Формируемые компетенции (индикаторы)*
			Всего часов	Из них практическая подготовка		
1	2	3	4	5	6	7
1	2	Тема 2. Знакомство с Java. Среда разработки	4	4	Лабораторная работа	УК-1, ОПК-3, УК-2, ОПК-2
1	3	Тема 3. Форматный вывод	2	2	Лабораторная работа	УК-1, ОПК-3, ОПК-2
1	4	Тема 4. Целые типы данных	2	2	Лабораторная работа	УК-1, ОПК-3, ОПК-2
1	5	Тема 5. Типы данных с плавающей точкой. Методы класса Math	2	2	Лабораторная работа	УК-1, ОПК-3, ОПК-2
1	6	Тема 6. Логический тип данных	2	2	Лабораторная работа	УК-1, ОПК-3, ОПК-2
1	7	Тема 7. Условный оператор	2	2	Лабораторная работа	УК-1, ОПК-3, ОПК-2
1	8	Тема 8. Оператор выбора	2	2	Лабораторная работа	УК-1, ОПК-3, ОПК-2
1	9	Тема 9. Циклические конструкции. Операторы перехода	4	4	Лабораторная работа	УК-1, ОПК-3, УК-2, ОПК-2
1	10	Тема 10. Статические методы	4	4	Лабораторная работа	УК-1, ОПК-3, УК-2, ОПК-2

1	11	Тема 11. Работа с файлами	2	2	Лабораторная работа	УК-1, ОПК-3, УК-2, ОПК-2
1	13	Тема 13. Представление числовых данных и их обработка	4	4	Лабораторная работа	УК-1, ОПК-3, УК-2, ОПК-2
1	15	Тема 15. Битовые операции	2	2	Лабораторная работа	УК-1, ОПК-3, ОПК-2

4.3.2. Перечень тем (вопросов), выносимых на самостоятельное изучение студентами в рамках самостоятельной работы (СР)
Не предусмотрено

4.4. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ПО ОРГАНИЗАЦИИ САМОСТОЯТЕЛЬНОЙ РАБОТЫ СТУДЕНТОВ

Во время изучения дисциплины студент посещает лекции, практические занятия, выполняет лабораторные задания, готовится к тестам, зачетам и экзаменам. Для каждого вида деятельности необходимо правильно организовать самостоятельную работу.

Лекции. В высшем учебном заведении лекция является важной формой учебного процесса. На лекции студенты получают глубокие и разносторонние знания. Лекция способствует развитию творческих способностей, формирует идейную убежденность, позволяет устанавливать связь учебного материала с производством, новейшими научными достижениями. Лекция требует три вида деятельности: подготовку к лекции, работу на лекции и работу после лекции.

После прослушивания лекции студент должен проработать и осмыслить полученный материал. На каждый пример, приведенный на лекции, желательно, (если это возможно) привести свой. Материал, изложенный в лекции, можно просмотреть в других источниках.

В процессе лекционного занятия студент должен выделять важные моменты, выводы, анализировать основные положения. Недостаточно только «слушать» лекцию. Возможности памяти человека не универсальны. Как бы внимательно студент ни слушал лекцию, большая часть информации вскоре после восприятия будет забыта. Чтобы восстановить лекционный материал, его нужно повторить, а для этого лекцию необходимо конспектировать. Конспект лекций должен быть в отдельной тетради, в которой не должно быть ничего, кроме лекции. Не надо стремиться подробно слово в слово записывать всю лекцию. Конспектируйте только самое важное в рассматриваемой теме: ключевые слова и их значения, примеры использования конструкций, что старается выделить лектор, на чем акцентирует внимание студентов.

Тетрадь для конспекта лекций также требует особого внимания. Ее нужно сделать удобной, практичной и полезной, ведь именно она является основным информативным источником при подготовке к различным отчетным занятиям, зачетам, экзаменам. Конечно, оформление лекционной тетради – это дело вкуса. Но целесообразно отделить поля, где студент мог бы изложить свои мысли, вопросы, появившиеся в ходе лекции. Полезно одну из страниц оставлять свободной. Она потребуется потом, при самостоятельной подготовке. Сюда можно будет занести дополнительную информацию по данной теме, полученную из других источников: рисунки, схемы, примеры кода и т.д.

Лабораторное занятие. Лабораторные занятия по решению задач существенно дополняют лекции. В процессе анализа и решения задач студенты расширяют и углубляют знания, полученные из лекционного курса и учебников, приобретают умение применять общие закономерности к конкретным случаям.

Необходимо, чтобы студенты готовили теоретический материал, т.к. именно невыполнение этого требования приводит к неудаче при решении задач.

Несмотря на различие в видах задач, их решение можно проводить по следующему общему плану (некоторые пункты плана могут выпадать в некоторых конкретных случаях): а) прочесть внимательно условие задачи; б) посмотреть, все ли термины в условиях задачи известны и понятны (если что-то неясно, следует обратиться к учебнику, просмотреть решения предыдущих задач, посоветоваться с преподавателем); в) произвести анализ задачи, (нужно чётко понимать, в чем будет заключаться решение задачи); г) решить задачу; д) протестировать полученное решение на данных из примеров к задаче, а также на дополнительных данных.

Если задача не решена или «не решается», то необходимо еще раз вернуться к пунктам а) и б). Сколько раз нужно возвращаться к этим пунктам? Практика показывает, что не более десяти раз. Если и после этого задача «не решается», то можно попытаться найти решение этой или похожей задачи в различных источниках.

Домашнее задание. При выполнении домашнего задания необходимо просмотреть текст лекции, разобраться с новыми определениями, посмотреть задания, которые были выполнены на лабораторной работе и применить полученные знания для выполнения домашней работы.

Тест. В первую очередь постарайтесь узнать чего ждать от теста, какие примерно там будут задания. Если вам доступны образцы теста (как, например, при сдаче ЕГЭ), необходимо этим воспользоваться и ежедневно тренироваться.

Не оставляйте все на самый последний момент. Если будете постоянно готовиться к тесту, вы наверняка улучшите свои знания. Для этого составьте план на каждый день, чтобы правильно распределять свое время.

Делайте небольшие перерывы во время учебы. В промежутках можно дать себе небольшую физическую нагрузку. Мозг лучше всего работает, когда умственный труд сменяется физическим. Прогуляйтесь, побегайте, поиграйте в баскетбол, попинайте мяч – помимо стимуляции умственной деятельности, это снимет стресс.

Отдых и контроль над волнением — одни из главных составляющих успеха при подготовке к тесту. Часто ошибки совершаются только из-за стресса, который мешает сконцентрироваться и собраться. Чтобы быть отдохнувшим и расслабленным, соблюдайте составленный режим и старайтесь высыпаться.

Экзамен. На экзамене оцениваются: 1) понимание и степень усвоения теории; 2) методическая подготовка; 3) знание фактического материала; 4) знакомство с обязательной литературой; 5) умение приложить теорию к практике, решать практические задачи и т. д.; 6) логика, структура и стиль ответа, умение защищать выдвигаемые положения. Но значение экзаменов не ограничивается проверкой знаний. Являясь естественным завершением работы студента, они способствуют обобщению и закреплению знаний и умений, приведению их в строгую систему, а также устранению возникших в процессе занятий пробелов.

Студенты готовятся к экзаменам по-разному. Одни из них прорабатывают лишь некоторые вопросы, выбранные наугад, другие стремятся запомнить весь материал подряд, не вникая глубоко в его суть. Работа при этом концентрируется на одном стремлении – сдать экзамен. Недостатки такой системы очевидны. Очевидно также, что подготовка не должна ограничиваться чтением лекционных записей. Первоначальные необработанные конспекты студента содержат факты, определения, выводы, сделанные преподавателем, но в них, как правило, слабо просматривается связующая идея курса, так как студент, записывая каждую лекцию в отдельности, редко способен сразу и достаточно точно уловить общую направляющую мысль. Поэтому конспект требует дополнительной обработки на основе использования учебников и рекомендованной литературы.

Существенные недостатки имеет и такой способ подготовки к экзаменам, как беглый просмотр всего материала. Он эффективен только на некоторых этапах планирования и закрепляющего повторения. Более надежный и целесообразный путь – это тщательная систематизация материала при вдумчивом повторении, установлении внутрипредметных связей, увязке различных тем и разделов, закреплении путем решения задач.

Перед экзаменом назначается консультация. Цель ее – дать ответы на вопросы, возникшие в ходе самостоятельной подготовки. Хотелось бы обратить особое внимание на важность предэкзаменационных консультаций. Здесь студент имеет полную возможность получить ответ на все неясные ему вопросы. А для этого он должен проработать до консультации весь курс. Кроме того, преподаватель будет отвечать на вопросы других студентов, что будет для вас повторением и закреплением знаний. И еще очень важное обстоятельство: лектор на консультации, как правило, обращает внимание на те разделы, по которым на предыдущих экзаменах ответы были неудовлетворительными, а также фиксирует внимание на наиболее трудных разделах курса. Некоторые студенты не приходят на консультации либо потому, что считают, что у них нет вопросов к лектору, либо полагают, что у них и так мало времени и лучше самому почитать материал по конспекту или в учебнике. Это глубокое заблуждение. Никакая другая работа не сможет принести столь значительного эффекта накануне экзамена, как консультация преподавателя.

Подготовку к экзамену следует начинать с первого дня изучения дисциплины. Как правило, на лекциях подчеркиваются наиболее важные и трудные вопросы или разделы курса, требующие внимательного изучения и обдумывания. Нужно эти вопросы выделить и обязательно постараться разобраться в них, не дожидаясь экзамена, проработать их, готовясь к лабораторным занятиям, попробовать самостоятельно решить несколько типовых задач. И если, несмотря на это, часть материала осталась неувоенной, ни в коем случае нельзя успокаиваться, надеясь на то, что это не попадется на экзамене. Факты говорят об обратном: если те или другие вопросы курса не вошли в экзаменационный билет, преподаватель может их задать (и часто задает) в виде дополнительных вопросов. Точно такое же отношение должно быть выработано к вопросам и задачам, перечисленным в экзаменационной программе, выдаваемой студентам еще до экзамена. Обычно эти же вопросы и аналогичные задачи содержатся в экзаменационных билетах. Не следует оставлять без внимания ни одного раздела курса; если не удалось в чем-то разобраться самому, нужно обратиться к товарищам; если и это не помогло выяснить какой-либо вопрос до конца, нужно обязательно задать этот вопрос преподавателю на предэкзаменационной консультации. Чрезвычайно важно приучить себя к умению самостоятельно мыслить, учиться думать, понимать суть дела. Очень полезно после проработки каждого раздела восстановить в памяти содержание изученного материала, кратко записав это на листе бумаги. Если этого не сделать, то большая часть материала останется не понятой, а лишь формально заученной, и при первом же вопросе экзаменатора студент убедится в том, насколько поверхностно он усвоил материал.

4.5. ПРИМЕРНАЯ ТЕМАТИКА КУРСОВЫХ РАБОТ (ПРОЕКТОВ)

Не предусмотрено.

УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)

а) перечень литературы

1. Кудинов Ю. И. Основы современной информатики: Учебное пособие. 2-е изд., испр. / Ю. И. Кудинов, Ф.Ф. Пащенко – СПб: Издательство «Лань», 2018. – 256 с. – ISBN: 978-5-8114-0918-1. – (ЭБС «Лань»)+
2. Теоретические основы информатики : учебное пособие / Б. Е. Стариченко. - 2-е изд., испр. и доп. - М. : Горячая линия-Телеком, 2004. - 312 с. - ISBN 5-93517-090-6. (18 экз.)+

Дополнительная:

1. Парфилова Н. И., Пруцков А. В., Пылькин А. Н., Трусов Б. Г. Информатика и программирование. Основы информатики : учебник для студ. учреждений высш. проф. образования. – Издат. центр "Академия", 2012. – ISBN: 978-5-7695-4144-1. – 256 с. – (ЭБС «БиблиоТех»). +
2. Кудинов Ю. И. Практикум по основам современной информатики: Учебное пособие / Ю. И. Кудинов, Ф.Ф. Пащенко, А.Ю. Келина – СПб: Издательство «Лань», 2011. – 352 с. – ISBN: 978-5-8114-1152-8. – (ЭБС «Лань»)+

б) периодические издания

в) список авторских методических разработок:

Материалы курса, опубликованные в ИОС «DOMIC».

г) базы данных, информационно-справочные и поисковые системы _____

JDK 11 Documentation. URL: <https://docs.oracle.com/en/java/javase/11/>

МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

6.1. УЧЕБНО-ЛАБОРАТОРНОЕ ОБОРУДОВАНИЕ:

Для проведения лекционных занятий необходима аудитория с презентационным оборудованием, для проведения лабораторных занятий необходима аудитория на 25-30 рабочих мест (в зависимости от численности учебной группы), оборудованная доской, презентационной техникой, компьютерами.

6.2. ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ:

1. Комплект разработчика приложений Java Platform (JDK) 11, Standard Edition (распространяется бесплатно);

2. Интегрированная среда разработки NetBeans IDE 12 (распространяется бесплатно, LGPLv2.1, GPLv2 with Classpatch exception).

6.3. ТЕХНИЧЕСКИЕ И ЭЛЕКТРОННЫЕ СРЕДСТВА:

ИОС EDUCA, DOMIC, презентационное оборудование, персональный компьютер с возможностью демонстрации презентаций в формате ppt.

ОБРАЗОВАТЕЛЬНЫЕ ТЕХНОЛОГИИ

При реализации данного курса используются следующие образовательные технологии: технологии традиционного обучения, технологии проблемного обучения, технологии контекстного обучения, интерактивные технологии, технологии дистанционного обучения.

ОЦЕНОЧНЫЕ МАТЕРИАЛЫ ДЛЯ ТЕКУЩЕГО КОНТРОЛЯ И ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ

8.1. ОЦЕНОЧНЫЕ МАТЕРИАЛЫ ДЛЯ ВХОДНОГО КОНТРОЛЯ

Не предусмотрены.

8.2. ОЦЕНОЧНЫЕ МАТЕРИАЛЫ ТЕКУЩЕГО КОНТРОЛЯ

Разноуровневые задания к лабораторному практикуму. Каждое задание оценивается по 100-бальной шкале. Баллы за каждое задание представлены в таблице.

Учебная единица	Баллы
java_intro / Основы языка Java. Задача 2 (Задания для самостоятельного выполнения)	0.5
java_intro / Основы языка Java. Задача 3 (Задания для самостоятельного выполнения)	0.5
java_printf / Форматный вывод. Задача 1 (Задания для самостоятельного выполнения)	0.5
java_int / Целые типы данных. Задача 3 (Задания для самостоятельного выполнения)	1
java_int / Целые типы данных. Задача 4 (Задания для самостоятельного выполнения)	1
java_int / Целые типы данных. Задача 5 (Задания для самостоятельного выполнения)	1
java_int / Целые типы данных. Задача 6 (Задания для самостоятельного выполнения)	1
java_int / Целые типы данных. Задача 8 (Задания для самостоятельного выполнения)	1.5
java_float / Типы данных с плавающей точкой. Задача 2 (Задания для самостоятельного выполнения)	0.5
java_float / Типы данных с плавающей точкой. Задача 4 (Задания для самостоятельного выполнения)	1
java_float / Типы данных с плавающей точкой. Задача 5 (Задания для самостоятельного выполнения)	1
java_float / Типы данных с плавающей точкой. Задача 6 (Задания для самостоятельного выполнения)	1
java_float / Типы данных с плавающей точкой. Задача 7 (Задания для самостоятельного выполнения)	1
java_boolean / Логический тип данных. Задача 3 (Задания для самостоятельного выполнения)	0.5
java_boolean / Логический тип данных. Задача 4 (Задания для самостоятельного выполнения)	0.5
java_boolean / Логический тип данных. Задача 5 (Задания для самостоятельного выполнения)	0.5
java_boolean / Логический тип данных. Задача 7 (Задания для самостоятельного выполнения)	0.5
java_boolean / Логический тип данных. Задача 8 (Задания для самостоятельного выполнения)	1
java_boolean / Логический тип данных. Задача 9 (Задания для самостоятельного выполнения)	1
java_if / Условный оператор. Задача 2 (Задания для самостоятельного выполнения)	0.5
java_if / Условный оператор. Задача 3 (Задания для самостоятельного выполнения)	0.5
java_if / Условный оператор. Задача 5 (Задания для самостоятельного выполнения)	0.5
java_if / Условный оператор. Задача 6 (Задания для самостоятельного выполнения)	0.5
java_if / Условный оператор. Задача 7 (Задания для самостоятельного выполнения)	1
java_if / Условный оператор. Задача 8 (Задания для самостоятельного выполнения)	1
java_if / Условный оператор. Задача 9 (Задания для самостоятельного выполнения)	1
java_switch / Оператор выбора. Задача 3 (Задания для самостоятельного выполнения)	1
java_switch / Оператор выбора. Задача 4 (Задания для самостоятельного выполнения)	1
java_switch / Оператор выбора. Задача 5 (Задания для самостоятельного выполнения)	1
java_switch / Оператор выбора. Задача 6 (Задания для самостоятельного выполнения)	1

java_cycle / Циклические конструкции. Задача 2 (Задания для самостоятельного выполнения)	0.5
java_cycle / Циклические конструкции. Задача 3 (Задания для самостоятельного выполнения)	0.5
java_cycle / Циклические конструкции. Задача 5 (Задания для самостоятельного выполнения)	0.5
java_cycle / Циклические конструкции. Задача 6 (Задания для самостоятельного выполнения)	1
java_cycle / Циклические конструкции. Задача 7 (Задания для самостоятельного выполнения)	1
java_cycle / Циклические конструкции. Задача 8 (Задания для самостоятельного выполнения)	1.5
java_cycle / Циклические конструкции. Задача 9 (Задания для самостоятельного выполнения)	2
java_cycle / Циклические конструкции. Задача 10 (Задания для самостоятельного выполнения)	1.5
java_cycle / Циклические конструкции. Задача 11 (Задания для самостоятельного выполнения)	2
java_cycle / Циклические конструкции. Задача 12 (Задания для самостоятельного выполнения)	1.5
java_static_met / Статические методы. Задача 1 (Задания для самостоятельного выполнения)	1.5
java_static_met / Статические методы. Задача 2 (Задания для самостоятельного выполнения)	0.5
java_static_met / Статические методы. Задача 3 (Задания для самостоятельного выполнения)	0.5
java_static_met / Статические методы. Задача 4 (Задания для самостоятельного выполнения)	0.5
java_static_met / Статические методы. Задача 5 (Задания для самостоятельного выполнения)	1
java_static_met / Статические методы. Задача 6 (Задания для самостоятельного выполнения)	1.5
java_static_met / Статические методы. Задача 7 (Задания для самостоятельного выполнения)	2
java_files / Работа с файлами. Задача 1 (Задания для самостоятельного выполнения)	2.5
java_files / Работа с файлами. Задача 2 (Задания для самостоятельного выполнения)	1.5
present_digit / Сумма целых чисел (Задания для самостоятельного выполнения)	0.5
present_digit / Произведение целых чисел (Задания для самостоятельного выполнения)	0.5
present_digit / Наибольшее целое число (Задания для самостоятельного выполнения)	0.5
present_digit / Дополнение числа (Задания для самостоятельного выполнения)	0.5
present_digit / Двоичный дополнительный код (Задания для самостоятельного выполнения)	0.5
present_digit / Значение в двоичной кодировке (Задания для самостоятельного выполнения)	1
present_digit / Сумма вещественных чисел (Задания для самостоятельного выполнения)	1
present_digit / Произведение вещественных чисел (Задания для самостоятельного выполнения)	1

полнения)	
present_digit / Кодирование вещественных чисел (Задания для самостоятельного выполнения)	1.5
java_bitwise_op / Битовые операции. Задача 1 (Задания для самостоятельного выполнения)	1.5
java_bitwise_op / Битовые операции. Задача 2 (Задания для самостоятельного выполнения)	1.5
java_bitwise_op / Битовые операции. Задача 3 (Задания для самостоятельного выполнения)	2

1. Лабораторная работа по теме «Циклические конструкции»

Циклические конструкции

Задача 9. Вычисления с заданной точностью

1. Напишите программу нахождения суммы ряда **из вашего варианта** с заданной точностью ϵ . Вам необходимо использовать рекуррентные соотношения при вычислении очередного члена ряда. Вычисление заканчивается, если модуль очередного слагаемого становится меньше заданного значения точности (ϵ), причем для этих рядов (при $|x| < 1$) абсолютная величина суммы всех отброшенных членов ряда будет меньше ϵ .

🔔 **Внимание!** При вычислении суммы ряда не нужно пользоваться контрольной формулой, она нужна только для самопроверки.

📌 **Рекомендация:** При вычислении очередного элемента ряда используйте рекуррентные соотношения, т.е. очередной элемент последовательности a_n вычисляйте через предыдущий элемент a_{n-1} . Не используйте метод возведения в степень `Math.pow()`.

📌 **Совет:** Выведите для отладки каждый вычисленный элемент последовательности, произведите вычисления элементов вручную и сравните с результатом работы программы.

2. Для оценки правильности расчетов выполните вычисление по контрольной формуле **из вашего варианта**.
3. Вычислите погрешность вычислений (модуль разности между суммой (полученной в п.1) и значением контрольной формулы (полученным в п.2)).

🔔 **Внимание!** Убедитесь, что полученная погрешность меньше заданной точности. Если это не так, то скорее всего вы неверно вычислили сумму ряда.

4. Выведите количество слагаемых ряда, которые нужно просуммировать для достижения необходимой точности.

🔔 **Внимание!** При выводе задействуйте максимальное количество десятичных разрядов в типе `double`, т.е. 16.

Примеры работы программы:

Для ряда

$$1 + \frac{x}{1!} + \frac{x^2}{2!} + \frac{x^3}{3!} + \frac{x^4}{4!} + \dots$$

и контрольной формулы e^x программа должна работать следующим образом

<p>Введите точность вычислений: 0,1 Введите значение x: 0,8</p> <p>1. Сумма ряда с точностью: 2,205333333333334 2. Значение контрольной формулы: 2,2255409284924680 3. Погрешность: 0,0202075951591345 4. Количество слагаемых: 4</p>
<p>Введите точность вычислений: 0,0001 Введите значение x: 0,8</p> <p>1. Сумма ряда с точностью: 2,2255363657142860 2. Значение контрольной формулы: 2,2255409284924680 3. Погрешность: 0,0000045627781819 4. Количество слагаемых: 8</p>
<p>Введите точность вычислений: 0,000000000001 Введите значение x: 0,8</p> <p>1. Сумма ряда с точностью: 2,2255409284924395 2. Значение контрольной формулы: 2,2255409284924680 3. Погрешность: 0,000000000000284 4. Количество слагаемых: 15</p>

2. Лабораторная работа по теме «Статические методы»

Статические методы

Задача 1. ASCII-графика

Напишите статический метод, зависящий от одного целочисленного аргумента, который выводит на экран изображение выбранной размерности по вашему варианту. Необходимо точно воспроизвести вывод, согласно вашему варианту в соответствии с указанной размерностью. На первом изображении в примерах приведена фигура размерности 3. В ней различные вложенные фигуры внутри основного рисунка имеют высоту 3. Вложенные фигуры обладают тем свойством, что их высота определяет их ширину.

Для решения задачи потребуется использовать вложенные циклы с инструкциями `print` и `println`.

Внимание! Вы должны использовать статические методы для структурирования своего решения. Старайтесь **избегать избыточности кода**, структурируйте свою программу.

Внимание! Поскольку символ `\` является специальным символом, с которого начинаются различные управляющие последовательности, например, `\n` (переход на новую строку) или `\t` (символ табуляции), то, для того, чтобы вывести символ `\` в консоль, вам нужно его экранировать, т.е. заменить его на последовательность `\\`.

```
System.out.println("\\"); // на экране будет выведена строка "\\"
```

Совет. Давайте вспомогательным методам «говорящие» имена. В соответствии с рекомендациями по оформлению кода имена методов должны представлять собой глаголы или глагольные словосочетания.

Совет. Для вывода фигуры вам потребуется многократно выводить одинаковые символы заданное количество раз, например, вывести символ `.` 5 раз, чтобы получить строку `.....` или строку `=|` 8 раз, чтобы получить строку `=|=|=|=|=|=|=|`.

Для решения этой подзадачи логично реализовать отдельный метод, который будет зависеть от двух формальных параметров: строки символов `str` и количества повторов `n`, метод должен выводить указанную строку символов заданное количество раз. Сигнатура этого метода представлена в примере

```
public static void f(String str, int n) {  
    // реализация метода: нужно n раз вывести строку str.  
}
```

Теперь для того, чтобы вывести строки `.....` и `=|=|=|=|=|=|=|` достаточно сделать следующие вызовы

```
f(" ", 5);  
f("=", 8);
```

Добавьте комментарий к каждому отдельному методу, в комментарии опишите что делает данный метод.

3. Контрольная работа «Представление числовых данных и их обработка»

1. При беззнаковой двоичной кодировке и 2-байтном двоичном слове вычислить сумму чисел. Записать значение регистра переполнения. Корректен ли будет результат вычислений?
2. При беззнаковой двоичной кодировке и 2-байтном двоичном слове вычислить произведение чисел. Записать значение регистра переполнения. Корректен ли будет результат вычислений?
3. При n-байтовом кодировании целых чисел со знаком, какое наибольшее число Z_{max} можно закодировать? Поясните ответ.
4. Пусть для представления числа используется k десятичных разрядов. Построить дополнение числа n в десятичной системе счисления.
5. Пусть для представления числа используется k двоичных разрядов. Построить дополнительный код (ДК) числа n
6. Выполнить вычитание в двоичной кодировке (перевести оба числа в двоичный код и сложить)
7. Вычислите сумму вещественных чисел X_1 и X_2 и запишите результат, если для записи мантииссы отводится n разрядов. Сравните полученный результат с точным результатом вычисления. Какова погрешность вычисления?
8. Вычислите произведение вещественных чисел X_1 и X_2 и запишите результат, если для записи мантииссы отводится n разрядов. Сравните полученный результат с точным результатом вычисления. Какова погрешность вычисления?
9. Установить распределение разрядов двоичного представления числа вещественного типа, если для записи отводится n бит, а максимальное значение десятичного порядка m. Какова точность обработки таких чисел?

8.3. ОЦЕНОЧНЫЕ МАТЕРИАЛЫ ПРОМЕЖУТОЧНОГО КОНТРОЛЯ

- баллы за работу в семестре (выполнение лабораторных работ) – 65 баллов
- баллы за экзаменационный тест – 30 баллов
- баллы за посещаемость – 5 баллов

Критерии итоговой оценки за курс:

- 80 и более – отлично
- от 70 до 79 – хорошо
- от 60 до 69 и – удовлетворительно
- менее 60 – неудовлетворительно

В случае, если студент не набрал необходимое количество баллов, проводится собеседование, на котором ему дается дополнительная задача и вопросы из списка. Ответ на каждый вопрос и задачу оценивается из 5 баллов. Студенту случайным образом дается нужное количество вопросов, но не более 4.

Если студент в течение семестра набрал менее 40 баллов, тогда на промежуточной аттестации ему выставляется неудовлетворительно.

Список вопросов для экзаменационного теста и собеседования

1. Понятие алгоритма, программы, команды. Свойства алгоритма. Основные качества программ. Правила написания программного кода.

2. Понятие алгоритма, программы, команды. Способы записи алгоритма. Этапы разработки программ.

3. Языки программирования низкого и высокого уровня. Язык программирования Java, его особенности. Виртуальная машина Java и байт-код. Сборка мусора. Отладка программы. Методы отладки. Виды ошибок в программе.

4. Понятие переменной. Имена переменных. Ключевые и зарезервированные слова. Простые типы данных. Объявление переменной. Оператор присваивания. Порядок выполнения операций.

5. Целые типы данных. Понятие ширины типа. Операции над величинами целого типа. Сокращенная запись арифметических операций. Особенность деления в Java. Определение цифр числа. Способы задания целочисленных констант. Целочисленные константы типа long. Примеры.

6. Обмен значений переменных. Инкремент и декремент. Префиксная и постфиксная формы. Понятие реперполнения. Примеры.

7. Типы данных с плавающей точкой. Константы с плавающей точкой. Константы с плавающей точкой типа float. Понятие бесконечности. Значение NaN, особенности его обработки. Точность вычислений. Константы класса Math. Методы класса Math: прямые трансцендентные, обратные трансцендентные, экспоненциальные. Примеры.

8. Типы данных с плавающей точкой. Константы с плавающей точкой. Константы с плавающей точкой типа float. Понятие бесконечности. Значение NaN, особенности его обработки. Точность вычислений. Константы класса Math. Методы класса Math: функции округления. Генерация псевдослучайных чисел. Генерация целых и вещественных чисел в заданном интервале. Примеры.

9. Форматный вывод. Форматная строка. Понятие и виды спецификаторов формата. Форматирование целых чисел. Форматирование вещественных чисел. Указание минимальной ширины поля. Примеры.

10. Форматный вывод. Форматная строка. Понятие и виды спецификаторов формата. Указание точности. Флаги формата. Использование индекса аргументов. Примеры.
11. Логический тип данных. Логические операции. Замыкающие логические операции и их особенности. Применение замыкающих логических операций. Законы де Моргана.
12. Разветвляющиеся алгоритмы. Полная и неполная формы ветвления. Условный оператор и его особенности. Пример программы с блок-схемой. Условные оператор и логические переменные. Тернарный оператор. Примеры.
13. Сложные условия. Порядок выполнения логических операций. Пример программы с блок-схемой.
14. Оператор выбора. Порядок выполнения оператора выбора. Особенности оператора выбора. Вложенные операторы выбора.
15. Цикл с известным числом шагов и его особенности. Пример программы с блок-схемой. Приемы накопления суммы и произведения, комбинация приемов. Рекуррентные соотношения. Переменные-флаги. Переменные-счетчики.
16. Цикл с условием и его особенности. Пример программы с блок-схемой. Вычисления с заданной точностью. Цикл с постусловием и его особенности.
17. Вложенные циклы. Примеры. Операторы перехода и их особенности: break, continue, return. Варианты использования оператора break. Использование break в качестве оператора безусловного перехода.
18. Применение циклов. Аналитическое и точное решение. Численные методы, их достоинства и недостатки. Метод дихотомии для решения уравнения и его программная реализация. Особенности метода.
19. Применение циклов. Аналитическое и точное решение. Численные методы, их достоинства и недостатки. Численные методы вычисления площади криволинейной трапеции. Программная реализация одного из методов.
20. Применение циклов. Аналитическое и точное решение. Численные методы, их достоинства и недостатки. Численный метод вычисления длины кривой и его программная реализация.
21. Статические методы. Понятие процедурной декомпозиции. Примеры методов. Виды методов. Определение статического метода: составные части метода. Вызов статического метода. Статические методы, не возвращающие значения. Примеры.
22. Статические методы. Формальные и фактические параметры. Статические методы, возвращающие значения и их особенности. Логические методы. Примеры.
23. Использование класса Scanner для чтения данных из консоли, файла и строки. Понятие лексемы. Процедура чтения данных из файла. Методы класса Scanner. Примеры.
24. Класс PrintWriter. Методы класса PrintWriter. Примеры.
25. Преобразование типов. Виды преобразования типов. Условия выполнения неявного преобразования типов. Условия выполнения явного преобразования типов. Общая форма явного преобразования. Примеры. Виды приведений типов.
26. Преобразование типов. Тожественное преобразование, расширение простого типа. Возможные расширения простых типов. Искажения при расширении простого типа. Примеры.
27. Преобразование типов. Сужения простого типа. Возможные виды сужения простого типа. Алгоритм сужения дробного типа до целочисленного. Примеры.
28. Битовые операции. Битовые логические операции. Битовые маски и их использование. Примеры.
29. Битовые операции. Битовые сдвиги: логический, циклический, арифметический. Примеры.
30. Системы счисления. Классификация с/с.
31. Перевод целых чисел из одной с/с в другую. Преобразование $Z_p \rightarrow Z_1 \rightarrow Z_q$. Пример.
32. Перевод целых чисел из одной с/с в другую. Преобразование $Z_p \rightarrow Z_{10}$. Пример.

33. Перевод целых чисел из одной с/с в другую. Преобразование $Z_{10} \rightarrow Z_q$. Пример.
34. Перевод дробных чисел из одной с/с в другую. Преобразование $0, Y_{10} \rightarrow 0, Y_q$.
Пример.
35. Перевод дробных чисел из одной с/с в другую. Преобразование $0, Y_r \rightarrow 0, Y_{10}$.
Пример.
36. Понятие экономичности с/с. Наиболее экономичная с/с.
37. Перевод чисел между с/с $2 \leftrightarrow 8 \leftrightarrow 16$. Пример.
38. Кодирование и обработка в компьютере целых чисел без знака. Понятие наибольшего целого числа. Понятие регистра переноса. Примеры.
39. Кодирование и обработка в компьютере целых чисел со знаком. Прямой и дополнительный код. Дополнение числа. Дополнительный код двоичного числа. Примеры.
40. Проверка и анализ корректности операций сложения/вычитания целых чисел со знаком. Примеры.
41. Кодирование и обработка в компьютере вещественных чисел. Понятие машинного нуля. Особенности обработки.
42. Сложение и умножение нормализованных чисел. Примеры.

Примеры дополнительных задач для собеседования на промежуточной аттестации:

1. Пример дополнительных заданий на экзамене:

Дан фрагмент кода, нужно определить значения переменных, например

Запишите значения переменных x , y и что будет выведено на экран

```
double x = 2. / 0;
double y = -1 / 0.;
System.out.print(x + y);
```

Запишите значение переменной b и что будет выведено на экран

```
long m = -130;
byte b = (byte)-m;
System.out.print("b" + b);
```

Запишите значения переменных b и c и что будет выведено на экран

```
int a = -125;
int b = (a >> 2);
int c = (a << 2);
System.out.println((byte)(b + c));
```

Запишите значения переменных b и c и что будет выведено на экран

```
int a = -10;
int b = (a >>> 2);
int c = (a << 2);
System.out.println((byte) (b + c));
```

Запишите значения переменных b и c и что будет выведено на экран

```
int a = -10;
```

```
int b = ~256;  
int c = a ^ b;  
System.out.println((byte) c);
```

Разработчики:



(подпись)

доцент

(занимаемая должность)

Зинченко А.С.

(Ф.И.О.)

Программа составлена в соответствии с требованиями ФГОС ВО по направлению подготовки 02.03.02 «Фундаментальная информатика и информационные технологии» (уровень бакалавриата), утвержденный приказом Министерства образования и науки Российской Федерации от 23 августа 2017 г. N 808, зарегистрированный в Минюсте России «14» сентября 2017 г. № 48185 с изменениями и дополнениями с изменениями и дополнениями от: 26 ноября 2020 г., 8 февраля 2021 г.