

Министерство науки и высшего образования Российской Федерации
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ (НИ ТГУ)

Механико-математический факультет

УТВЕРЖДАЮ:

Декан

 Л. В. Гензе

« 30 » 06 20 22 г.

Рабочая программа дисциплины

Программирование

по направлению подготовки

**01.03.01 Математика, 02.03.01 Математика и компьютерные науки,
01.03.03 Механика и математическое моделирование**

Направленность (профиль) подготовки :

**Основы научно-исследовательской деятельности в области математики, Основы
научно-исследовательской деятельности в области математики и компьютерных
наук, Основы научно-исследовательской деятельности в области механики и
математического моделирования**

Форма обучения

Очная

Квалификация

Бакалавр

Год приема

2022

Код дисциплины в учебном плане: Б1.О.2.18.01

СОГЛАСОВАНО:

Руководитель ОП

 Л. В. Гензе

Председатель УМК

 Е. А. Тарасов

Томск – 2022

1. Цель и планируемые результаты освоения дисциплины

Целью освоения дисциплины является формирование следующих компетенций:

ОПК-2 Способен находить или создавать, анализировать, реализовывать программно и использовать на практике современный математический аппарат, математические модели и алгоритмы, в том числе с применением современных вычислительных систем в научно-исследовательской и (или) опытно-конструкторской деятельности в различных областях техники, естествознания, экономики и управления.

ОПК-7 Способен разрабатывать алгоритмы и компьютерные программы, пригодные для практического применения.

Результатами освоения дисциплины являются следующие индикаторы достижения компетенций:

ИОПК 2.1 Использует методы построения и анализа математических моделей в задачах естествознания, технике, экономике и управления

ИОПК 2.2 Демонстрирует умение применять на практике математические модели и компьютерные технологии (в том числе с применением многопроцессорных систем) для решения различных задач в области профессиональной деятельности

ИОПК 2.3 Участвует в разработке математических моделей для решения задач естествознания, техники, экономики и управления под руководством более квалифицированного работника

ИОПК 7.1 Владеет навыками использования основных языков программирования для решения задач науки и техники

ИОПК 7.2 Демонстрирует умение отбора среди существующих математических методов, наиболее подходящих для решения конкретной прикладной задачи, в том числе с применением современных вычислительных систем

2. Задачи освоения дисциплины

- Сформировать у обучающихся систему понятий и представлений об основах программирования;

- Изучить основы программирования, освоить методы и процессы решения задач на современных языках программирования, этапы построения программ;

- Выработать навыки составления программ на современных языках программирования для решения различных задач.

3. Место дисциплины в структуре образовательной программы

Дисциплина относится к Блоку 1 «Дисциплина (модули)».

Дисциплина относится к обязательной части образовательной программы. Дисциплина входит в модуль Модуль: Компьютерные науки.

4. Семестр(ы) освоения и форма(ы) промежуточной аттестации по дисциплине

Первый семестр, экзамен

Второй семестр, зачет

Третий семестр, зачет

Четвертый семестр, экзамен

5. Входные требования для освоения дисциплины

Для успешного освоения дисциплины требуются компетенции, сформированные в ходе освоения образовательных программ предшествующего уровня образования.

6. Язык реализации

Русский

7. Объем дисциплины

Общая трудоемкость дисциплины составляет 13 з.е., 468 часов, из которых:

-лекции: 96 ч.

-практические занятия: 128 ч.

в том числе практическая подготовка: 128 ч.

Объем самостоятельной работы студента определен учебным планом.

8. Содержание дисциплины, структурированное по темам

Тема 1. Теоретические основы программирования и алгоритмизации. Язык программирования C++.

Краткое содержание темы: История развития вычислительной техники. Классы современных компьютеров. Архитектура фон Неймана. Принципиальное устройство компьютеров. Оперативная память. Процессор. Система элементарных команд процессора. Понятие алгоритма, его свойства. Способы представления алгоритмов. Представление алгоритмов с помощью блок-схем. Позиционные системы счисления. Структура программы на языке C (C++). Понятие типа данных. Стандартные скалярные типы данных. Переменные в языке C (C++), их описание и использование; область действия переменных. Выражения. Оператор присваивания. Операторы ввода-вывода с использованием стандартных устройств в языках C, C++. Форматированный ввод-вывод в языке C. Условный оператор. Оператор множественного выбора **switch**. Операторы цикла с пред-, и постусловием. Тестирование и отладка программ. Приемы, часто применяемые при отладке. Основные принципы структурного программирования. Теорема Бома-Джакопини. Метод проектирования программ «сверху вниз» - пошаговая детализация программ. Логические утверждения как характеристика множества допустимых значений переменных. Элементы теории Т. Хоара : определение частичной правильности последовательности операторов, аксиома правильности для оператора присваивания, теорема о достаточном признаке частичной правильности цикла с предусловием. понятия инварианта цикла и ограничивающей функции. Массивы как тип данных. Одномерные, двумерные массивы. Оператор цикла **for**. Указатели и их использование. Динамические переменные. Эквивалентность массивов и указателей. Динамические массивы. Понятие подпрограммы. Описание, вызов подпрограмм. Виды формальных параметров: входные, результаты, модифицируемые. Описание функции в языке C (C++). Формальные параметры функций. Вызов функции в языке C (C++). Действия компьютера при вызове функции. Соответствие формальных и фактических параметров при вызове функций. Глобальные параметры. Использование указателей и ссылок в качестве параметров функций. Использование массивов в качестве параметров функций. Указатели на функции. Прототипы функций. Заголовочные файлы и их подключение к проекту. Перегрузка функций. Формальные параметры функций по умолчанию.

Тема 2. Структуры данных, строки, функции, файлы на языке C++. Алгоритмы поиска, сортировки, хеширования, оценка их эффективности и сложности

Краткое содержание темы: **Структуры данных, строки, функции, файлы на языке C++.** Рекурсивные функции. Алгоритм поиска с возвратом на примере поиска пути в дорожной сети. Эффективность и сложность алгоритмов. Классификация алгоритмов по их сложности. Работа с информацией, расположенной в таблице (одномерном массиве). Постановка задач поиска, вставки и удаления элементов таблицы. Алгоритмы поиска, вставки и удаления в информации в неупорядоченном массиве. Оценка их эффективности. Алгоритм бинарного поиска в упорядоченном массиве; оценка его эффективности.

Алгоритмы вставки и удаления информации в упорядоченном массиве, оценка их эффективности. Постановка задачи сортировки одномерного массива. Алгоритмы сортировки: метод «пузырька», метод простого включения, метод прямого выбора, метод быстрой сортировки. Оценка их эффективности. Повышение эффективности стандартных методов сортировки. Хэширование как метод размещения данных в массиве. Коллизии и способы их разрешения. Алгоритмы поиска и вставки информации в массив при использовании хэширования. Преимущества и недостатки технологии хэширования.

Тема 3. Динамические структуры данных: стек, очередь, список, деревья. Объектно-ориентированное программирование

Краткое содержание темы: Алгоритм поиска с возвратом на примере поиска пути в дорожной сети. Стек как структура данных. Логическое описание. Варианты физической реализации стека: на базе массива и с помощью динамических переменных. Функции, реализующие операции со стеком. Очередь как структура данных. Логическое описание. Варианты реализации очереди с помощью массива и динамических переменных. Функции, реализующие операции с очередью. Линейный однонаправленный список (ЛОС), логическое описание. Варианты реализации списка с помощью массива и динамических переменных. Функции, реализующие операции с ЛОС. Деревья как структуры данных. Основные определения и способы реализации. Обходы деревьев. Деревья бинарного поиска. Алгоритмы поиска, вставки и удаления элементов в бинарном дереве поиска; оценка их эффективности. Начала объектно-ориентированного программирования (ООП). Классы как обобщение понятия структуры. Инкапсуляция – соединение переменных и методов работы с ними в одной структуре. Описание класса в языке C++. Конструкторы классов; конструкторы по умолчанию, конструкторы копирования. Шаблоны классов. Примеры создания классов для реализации стека, очереди и ЛОС. Использование принципа наследования при создании иерархии классов. Стандартные классы из библиотеки C++: строки, потоки, контейнеры. Понятие операционной системы, основные элементы операционной системы, принципы взаимодействия прикладных программ с операционными системами. Исключения как классы, обеспечивающие взаимодействие программы с операционной системой.

Тема 4 Базы данных.

Краткое содержание темы: Базы данных : необходимость в особой технологии, основные определения. Архитектура баз данных. Этапы проектирования базы данных. Инфологическая модель данных: понятия сущности, атрибутов, связи, ключа, E/R – диаграммы. Дatalogические модели данных. Типы баз данных: иерархическая, реляционная, сетевая. Варианты организации хранения базы данных во внешней памяти и связанные с ними алгоритмы поиска и обновления данных: хэширование, индексированные файлы, B – деревья. Реляционная модель данных: основные определения. Реляционная алгебра. Нормализация. Язык запросов SQL. Типы данных в SQL. Операторы CREATE, INSERT, DELETE, UPDATE, SELECT. Триггеры и хранимые процедуры. Встроенный SQL, взаимодействие SQL-операторов с программами, написанными на алгоритмическом языке высокого уровня. Динамические SQL-запросы. Проблемы безопасности БД. Основные методы защиты данных. Понятие транзакции. Пример простой базы данных, создание инфологической и реляционной модели данных, нормализация. СУБД MS Access. Работа с базами данных в MS Access.

Тема 5. Глобальная сеть, её информационные ресурсы и их структурирование.

Краткое содержание темы: Язык гипертекстовой разметки HTML. Введение в расширенный язык разметки XML. XML-схема. Создание XML-схем.

9. Текущий контроль по дисциплине

Текущий контроль по дисциплине проводится путем контроля посещаемости, проведения контрольных работ, тестов по лекционному материалу, выполнения

индивидуальных заданий и фиксируется в форме контрольной точки не менее одного раза в семестр.

Текущий контроль качества, объема и получения компетенций проводится во время консультации с преподавателем, а также на практических занятиях при выполнении индивидуальных практических занятий. Практические занятия по дисциплине «Программирование» призваны закрепить знания бакалавра по отдельным разделам курса, привить им навыки свободного владения технологиями программирования и развить алгоритмическое мышление. На практических занятиях обучающийся разрабатывает алгоритмы решения задач, составляет программы с использованием языка программирования С++ и тестирует программы выполненных заданий. По результатам выполнения индивидуального задания пишется отчет. В таблице 1 приведена система оценивания индивидуального задания.

Таблица 1. Система критериев при оценивании индивидуального задания

Критерии соответствия	Оценка
Содержание отчета и ответы по индивидуальному заданию являются полными. Студент правильно понимает терминологию. Демонстрирует умение понимать, доказательно и логически связно отвечать на вопросы.	зачтено
Неполное, логически противоречивое изложение отчета. Студент не понимает и неправильно использует терминологию. Не может доказательно и логически связно отвечать на вопросы. Отчет вообще не подготовлен к защите.	не зачтено

При ответе на вопросы при защите отчета по индивидуальному заданию оценивается полнота и точность ответа, логичность и аргументированность изложения материала, умения использовать в ответе фактический материал. Для выставления текущей успеваемости при контроле СРС рекомендуется использовать следующую таблицу (Таблица 1).

10. Порядок проведения и критерии оценивания промежуточной аттестации

Экзамен в первом семестре проводится в письменной форме по билетам. Билет содержит два теоретических вопроса и задачу. Продолжительность экзамена 1,5 часа. Экзамен в четвертом семестре проводится в устной форме по билетам. Экзаменационный билет состоит из двух частей. Продолжительность экзамена 1,5 часа.

Билет проверяет ИОПК 2.1, ИОПК 2.2, ИОПК 2.3. Ответы на вопросы билета даются путем выбора из списка предложенных.

Примерный перечень теоретических вопросов на экзамене:

1. История вычислительной техники (от абака до ЭВМ). Закон Мура. Классы современных компьютеров. Суперкомпьютеры. Кластер ТГУ Cyberia.
2. Архитектура ЭВМ фон Неймана. Основные устройства. ЭВМ с общей и распределенной памятью.
3. Основная память ЭВМ, ее структура. Представление информации в памяти и единицы измерения.
4. Системы счисления (позиционная и непозиционная). Перевод числа из одной системы счисления в другую.
5. Система элементарных команд процессора (операции, параметры операций).
6. Архитектура персонального компьютера. Системный блок и периферийные устройства.
7. Определение алгоритма и его свойства. Пример (алгоритм Евклида).
8. Способы записи алгоритмов.
9. Простейшие алгоритмы (блок-схемы): сложение 2-х чисел, поиск максимального из 2-х чисел, суммирование n чисел.
10. Блок-схемы алгоритмов с рекуррентными последовательностями: отрезки рядов, числа Фибоначчи.

11. Алгоритмический язык и его составляющие: алфавит, синтаксис, семантика. Компиляция и выполнение.
12. Способы описания синтаксиса языка программирования (БНФ и синтаксические диаграммы).
13. Основные понятия языка программирования: оператор, идентификатор, описание, функция.
14. Основные понятия языка программирования: константа, переменная. Обмен значениями.
15. Язык C/C++. История появления, достоинства и недостатки. Структура программы на C. Идентификаторы.
16. Базовые скалярные типы и операции над ними в ЯПВУ и в C. Примеры значений, операции, хранение в памяти.
17. Форматированный ввод/вывод на C.
18. Простой оператор присваивания на C. Арифметические операции. Математические функции. Преобразование типов.
19. Операции инкремента и декремента в C. Составные операторы присваивания в C.
20. Базовые управляющие конструкции (альтернатива) в C. Логические операции. Логический тип. Примеры.
21. Базовые управляющие конструкции (многозначное ветвление и цикл с предусловием) в C. Примеры. Проверка числа на простоту.
22. Базовые управляющие конструкции (цикл с постусловием и с параметром) в C. Примеры. Рекомендации по выбору цикла.
23. Тестирование и отладка программ. Тестирование как метод демонстрации ошибок в программе. Типы программных ошибок.
24. Методы отладки программ.
25. Операциональный подход программирования. Операторы условного и безусловного перехода. История появления структурного программирования.
26. Назначение структурного программирования. Требования к “хорошим” программам.
27. Принципы структурного программирования.
28. Структурная теорема Бома-Джакопини, ее следствия. Основное свойство “хороших” структур управления.
29. Методы структурного программирования. Пример.
30. Правила оформления текста программ в соответствии с принципами структурного программирования.
31. Теоретическое доказательство правильности программ.
32. Конструирование типов. Одномерные массивы. Описание и особенности работы с ними на C. Примеры.
33. Работа с массивами. Вычисление максимального и суммы элементов массива. Запись элементов массива в обратном порядке.
34. Представление массивов в памяти ЭВМ. Функция sizeof. Примеры. Эквивалентность массивов и указателей.
35. Многомерные массивы на C. Примеры. Вложенные циклы.
36. Поточковый ввод/вывод на C++. Структура программы на C++ с потоковым вводом/выводом.
37. Представление чисел в памяти ЭВМ с фиксированной и плавающей точкой. Арифметические операции для чисел с фиксированной и плавающей точкой. Отличие «компьютерных» чисел от классических математических.
38. Порядок. Сортировка массивов без использования дополнительных. Оценка сложности алгоритмов.
39. Поиск. Линейный и двоичный поиск элемента в массиве. Оценка сложности алгоритмов.
40. Статические и динамические структуры данных. Ссылочная реализация.

41. Указатели, ссылки, разыменованние указателей на С. Арифметика указателей.
42. Динамические переменные на С. Создание, использование и удаление динамических переменных.
43. Динамические массивы их создание, использование и удаление. Пример.
44. Строки на С. Объявление строк. Работа со строками в С. Примеры функций из `stdlib.h`.
45. Структуры. Объявление структурного типа и экземпляра структуры. Работа с полями экземпляра структуры. Указатель на экземпляр структуры. Операция «->».
46. Файлы на С. Объявление файловой переменной, открытие и закрытие файла, чтение и запись из файла. Функция `feof`.
47. Файлы на С++. Объявление файловой переменной, открытие и закрытие файла, чтение и запись из файла.

Примеры задач:

1. Написать программу на языке С «обращения» целочисленного массива: изменения порядка следования компонентов на противоположный. {1,2,5,6,4} -> {4,6,5,2,1}
2. Дано вещественное число X и целое N. Составить программу на языке С для вычисления X^N без использования математических функций. Значения X и N ввести с клавиатуры.
3. В одномерном целочисленном массиве определить минимальный элемент и его положение(!) (номера индекса). Составить программу на языке С. Массив вводится с клавиатуры.
4. Написать программу на языке С, подсчитывающую количество и сумму цифр натурального числа. Число вводится с клавиатуры.
5. Написать программу на языке С, определяющую количество смен знаков в целочисленном массиве, не содержащем «0»-х значений. Ввод значений массива осуществить с клавиатуры.
6. Дан одномерный целочисленный массив. Найти количество различных значений элементов этого массива. Например, для {0,0.1,0,2,1,0} должно быть напечатано 0 – 4, 1 – 2, 2 – 1.

Результаты экзамена определяются оценками «отлично», «хорошо», «удовлетворительно», «неудовлетворительно». Экзамен по курсу проводится устно. Экзаменационный билет состоит из двух теоретических вопросов и одной задачи. Б Максимальная возможная оценка 5 баллов. При ответе на вопрос оценивается полнота и точность ответа, логичность и аргументированность изложения материала, умения использовать в ответе фактический материал (Таблица 2).

Таблица 2

Оценка	Критерии соответствия
5	Правильно и развернуто изложен материал каждого вопроса и отчетов по индивидуальным заданиям соответствующего раздела. Студент полно, четко и логично излагает материал вопроса и защищаемый материал задания.
4	Правильно изложен материал каждого вопроса и отчетов по индивидуальным заданиям, но не весь материал изложен развернуто и логически структурировано.
3	В целом правильно изложен материал каждого вопроса и защищаемого отчетов по заданию, но изложение носит поверхностный характер и с нарушением логики изложения.
2	Материал ответа на каждый вопрос и защищаемых отчетов по заданиям представлен очень поверхностно и с нарушением логики изложения. Студент очень плохо владеет основными концепциями дисциплины. Допущены существенные терминологические и фактические ошибки.
1	Неверно изложен материал на вопросы билета, каждый отчет по индивидуальным заданиям написан с грубыми ошибками или отчеты вообще не подготовлены к защите.

Во втором семестре проводится зачет. Билет проверяет ИОПК 2.1, ИОПК 2.2, ИОПК 2.3. Ответы на вопросы билета даются путем выбора из списка предложенных.

Вопросы к зачету за второй семестр:

1. Эффективность и сложность алгоритмов.
2. Рекурсивные функции. Примеры. Правила, которые нужно соблюдать при написании рекурсивных функций.
3. Алгоритм поиска с возвратом на примере поиска пути в дорожной сети.
4. Задачи поиска, вставки и удаления информации в неупорядоченном массиве, вставка и удаление. Оценка сложности.
5. Задачи поиска, вставки и удаления информации в упорядоченном массиве. Оценка сложности.
6. Постановка задачи сортировки массива.
7. Сортировка массива методом «пузырька». Оценка сложности.
8. Сортировка массива методом простого включения. Оценка сложности.
9. Сортировка массива методом прямого выбора. Оценка сложности.
10. Быстрая сортировка массива. Оценка сложности.
11. Хэширование как метод размещения данных в массиве.

Зачет по курсу за второй семестр проводится устно. Экзаменационный билет состоит из двух теоретических вопросов. При ответе на вопрос оценивается полнота и точность ответа, логичность и аргументированность изложения материала, умения использовать в ответе фактический материал (Таблица 3).

Таблица 3. Система критериев при оценивании ответов на зачете

Критерии соответствия	Оценка
Содержание ответа являются полными. Студент правильно понимает терминологию. Демонстрирует умение понимать, доказательно и логически связно отвечать на вопросы.	зачтено
Неполное, логически противоречивое изложение ответов. Студент не понимает и неправильно использует терминологию. Не может доказательно и логически связно отвечать на вопросы.	не зачтено

Вопросы к зачету за третий семестр

1. Оператор описания типа. Примеры использования.
2. Понятие динамической структуры данных. Логический и физический уровни описания динамических структур.
3. Стек как динамическая структура. Логическое описание стека.
4. Физическая реализация стека с помощью массива.
5. Физическая реализация стека с помощью динамических переменных.
6. Очередь как динамическая структура. Логическое описание очереди.
7. Физическая реализация очереди с помощью динамических переменных.
8. Линейный однонаправленный список (ЛОС). Логическое описание ЛОС.
9. Физическая реализация ЛОС с помощью динамических переменных.
10. Деревья как динамическая структура данных. Основные определения, связанные с понятием дерева: корень, потомки, лист, степень дерева, глубина элемента и дерева в целом. Связь между количеством элементов в дереве, его степенью и глубиной.
11. Обходы деревьев и их разновидности.
12. Двоичные деревья. Обходы двоичных деревьев. Двоичные деревья поиска – определение.
13. Физическая реализация двоичных деревьев с помощью динамических переменных. Функции, с помощью которых осуществляются обходы двоичных деревьев.

14. Двоичные деревья поиска - алгоритмы поиска и вставки элементов; функции для реализации этих алгоритмов.
15. Двоичные деревья поиска – алгоритм удаления элемента (словесное описание или описание на уровне блок-схемы).
16. Основные принципы объектно-ориентированного программирования. Структура описания класса в языке C++. Уровни доступа к полям и методам объектов.
17. Конструкторы. Конструктор копирования.
18. Методы класса, их описание и использование.
19. Определение операций над объектами класса.
20. Шаблоны классов.
21. Понятие дружественных для класса функций и операций; их описание.
22. Принципы наследования в объектно-ориентированном программировании.
23. Понятие виртуальных методов класса.
24. Стандартный класс **string** в языке C++
25. Поточные классы в C++. Стандартные потоки **cin** и **cout**. Использование манипуляторов для форматирования данных при вводе-выводе.
26. Файловые потоки. Основные операции с файловыми потоками: открытие, закрытие, ввод-вывод.
27. Методы неформатированного чтения и записи в поток.

Зачет по курсу за третий семестр проводится устно. Экзаменационный билет состоит из двух теоретических вопросов. При ответе на вопрос оценивается полнота и точность ответа, логичность и аргументированность изложения материала, умения использовать в ответе фактический материал (Таблица 3). Билет проверяет ИОПК 2.1, ИОПК 2.2, ИОПК 2.3. Ответы на вопросы билета даются путем выбора из списка предложенных.

Экзамен в четвертом семестре проводится в устной форме по билетам. Экзаменационный билет состоит из двух частей. Продолжительность экзамена 1,5 часа.

Первая часть представляет собой тест из 3 вопросов, проверяющих ИОПК-7.1. Ответы на вопросы первой части даются путем выбора из списка предложенных.

Вторая часть содержит два вопроса, проверяющий ИОПК-7.2. Ответы на вопросы второй части дается в развернутой форме.

Вопросы к экзамену за четвертый семестр. Первая часть.

1. Основные определения теории баз данных: база данных (БД), система управления БД, администрация БД. Архитектура БД.
2. Модели данных. Инфологическая модель. Понятие сущности, атрибута, связи сущностей. «E/R – диаграммы». Понятие ключа.
3. Даталогические модели данных: иерархические, сетевые и реляционные БД.
4. Определение реляционной базы данных. Отношения, поля (домены) и записи (кортежи).
5. Хэширование как способ размещения данных во внешней памяти и выполнение операций с базами данных, для которых используется хэширование.
6. Размещение информации в базе данных во внешней памяти и выполнение основных операций с применением индексированных файлов.
7. Размещение информации в базе данных во внешней памяти и выполнение основных операций с применением B-деревьев.
8. Операции реляционной алгебры : проекция, выбор, объединение, разность.
9. Операции реляционной алгебры: соединение, пересечение, деление, декартово произведение.

10. Определение полной декомпозиции отношений. 5 нормальная форма отношения. Определение нормализации.
11. Функциональная и полная функциональная зависимость между полями. Теорема Хита.
12. Определение полной декомпозиции отношений. Первые 3 нормальных формы отношений и нормальная форма Бойса – Кодда (НФБК).
13. Определение полной декомпозиции отношений. 4-я и 5-я нормальная форма отношений.
14. Язык SQL. Типы данных в SQL.
15. Язык SQL. Создание таблиц, доменов и индексов в реляционной БД с помощью SQL. Оператор CREATE.
16. Язык SQL. Изменение структуры отношений в реляционной БД с помощью SQL.
17. Язык SQL. Вставка, удаление и изменение кортежей с помощью SQL. Операторы DELETE, INSERT, UPDATE.
18. Язык SQL. Оператор SELECT в языке SQL.
19. Реализация операций проекции и выбора реляционной алгебры с помощью SQL.
20. Реализация операции соединения реляционной алгебры с помощью SQL.
21. Трёхзначная логика, используемая при работе с БД.
22. Реализация ограничений целостности БД с помощью SQL: формулировка требований к полям и кортежам отношений.
23. Реализация ограничений целостности БД с помощью SQL; формулировка требований к отношениям – внешние ключи.
24. Понятие триггера.
25. Встроенный SQL, взаимодействие SQL-операторов с программами, написанными на алгоритмическом языке высокого уровня. Использование динамических запросов во встроенном SQL.
26. Использование курсора при работе с отношениями в реляционной БД.

Вопросы к экзамену за четвертый семестр. Вторая часть.

1. Индивиды в OWL DL. Назначение, определение и их функции онтологиях. Декларация элемента в XML-схеме. Примеры разных деклараций элементов. XML-схема. Назначение схемы. Простые примеры схемы и документа построенного с её помощью.
2. Декларация атрибута в XML-схеме. Примеры атрибутов разной структуры.
3. Язык гипертекстовой разметки HTML
4. Определение составного типа в XML-схеме. Примеры декларации составного типа разной структуры.
5. Графическое представление утверждений в RDF.
6. Декларация атрибута в XML-схеме. Примеры атрибутов разных структур декларации.
7. Пространство имен. Причины возникновения и краткое описание. Примеры использования пространств имен в XML-схеме.
8. Декларация элемента в XML-схеме. Примеры декларации элементов разной структуры.
9. Свойства subClassOf, subPropertyOf, domain и range и их интерпретация. Примеры из нескольких предметных областей.
10. Декларация атрибута в XML-схеме. Примеры атрибутов разной структуры.
11. Структура XML. Документы и схемы. Примеры схемы и документов, построенных с их помощью.
12. Индивиды в OWL DL. Назначение, определение и их функции онтологиях.

13. Определение классов в OWL DL. Назначение и способы построения классов. Примеры их создания.
14. Декларация элемента в XML-схеме. Примеры декларации элементов разной структуры.

Результаты экзамена определяются оценками «отлично», «хорошо», «удовлетворительно», «неудовлетворительно». Экзамен по курсу проводится устно. При ответе на вопрос оценивается полнота и точность ответа, логичность и аргументированность изложения материала, умения использовать в ответе фактический материал (Таблица 2).

11. Учебно-методическое обеспечение

а) Электронный учебный курс по дисциплине в электронном университете «Moodle» - <https://moodle.tsu.ru/course/view.php?id=8367> (1 курс 1 семестр)
<https://moodle.tsu.ru/course/view.php?id=13057> (1 курс 2 семестр)
<https://moodle.tsu.ru/course/view.php?id=8378> (2 курс 1 семестр)
<https://moodle.tsu.ru/course/view.php?id=8379> (2 курс 1 семестр)

б) Оценочные материалы текущего контроля и промежуточной аттестации по дисциплине.

в) План семинарских / практических занятий по дисциплине.

г) Методические указания по проведению лабораторных работ.

д) Методические указания по организации самостоятельной работы студентов.

В ходе реализации дисциплины используются классические образовательные технологии – лекции, практические занятия; самостоятельное изучение рекомендованной литературы и постепенное выполнение индивидуального задания; промежуточная аттестация в приема зачетов (2, 3 семестры) и экзаменов (1,4 семестры).

Самостоятельная работа включает в себя: теоретическое освоение лекционного курса, практическое выполнение контрольных заданий и индивидуальных заданий, подготовку к зачету и экзамену. Для выполнения самостоятельной работы обеспечивается доступ к информационным ресурсам курса:

- материалы лекций;

- список вопросов для самостоятельной проверки знаний и подготовки к экзамену.

- список литературы, включающий учебники и книги по изучаемым в курсе вопросам.

Все контрольные работы и индивидуальные задания подобраны так, чтобы максимально стимулировать психологическую установку студентов-математиков на формирование связи между математической теорией и ее практическим применением. Отчет по каждой лабораторной работе включает теоретическую часть, выполненное практическое задание и анализ полученных результатов.

Для успешного освоения материала студентам необходимо посещать занятия, а во время самостоятельной работы пользоваться основной и дополнительной литературой, базами данных и информационно-справочными системами, которые представлены в списке литературы. Самостоятельная работа студентов состоит в повторении материала с практических занятий и самостоятельного изучения дополнительных вопросов, более глубокого анализа темы с помощью литературы.

12. Перечень учебной литературы и ресурсов сети Интернет

а) основная литература:

1. Керниган Б., Ритчи Д. Язык программирования Си. СПб.: «Невский диалект», 2001.
2. Т. А. Павловская. С, С++. Программирование на языке высокого уровня. СПб.: ПИТЕР, 2002.
3. В. В. Подбельский Язык С++. М.: Финансы и статистика, 1999.
4. Т. А. Павловская, Ю. А. Щупак. С, С++. Структурное программирование: Практикум. СПб.:ПИТЕР, 2002.
5. Васильев А.Н. Программирование на С++ в примерах и задачах.
6. Конова Е., Поллак Г. Алгоритмы и программы. С++. СПб.: Издательство «Лань», 2017.384 с.
7. Ульман Дж. Д., Уидом Дж. Введение с системы баз данных. - М.: Издательство «Лори», 2000.
8. Ю.Б. Бакаревич, Н.В.Пушкина. Самоучитель MS Office 2016. – СПб., БХВ-Петербург, 2017, 418 с

б) дополнительная литература:

1. Вопросы и ответы по С и С++. Москва: «МикроАрт», 1997.
2. Брайан Оверленд. С++ без страха. М.; изд-во «Триумф», 2005.
3. Йодан Э. Структурное программирование и конструирование программ. М.:Мир, 1979.
4. Дейкстра. Дисциплина программирования. М.: Мир, 1978.
5. Вирт Н. Алгоритмы и структуры данных. М: Мир, 1985.
6. Мейер В., Бодуэн К. Методы программирования. М.:Мир,1982, Т.1,2.
7. Могилев А.В., Пак Н.И., Хеннер Е.К. Информатика. М.- Академия, 1999.
8. Дейт К. Дж. Введение в системы баз данных. – К.; М.; СПб.: Издательский дом «Вильямс», 1999
9. Задачник по программированию для математиков. Томск: Изд-во Том. ун-та, 2001.
10. Страуструп Б. Программирование. Принципы и практика использования С++. Москва:Вильямс , 2011 – 1238 стр.
11. Эдди С.Э. XML. Справочник, СПб., Питер, 1999, 480с.

в) ресурсы сети Интернет:

1. ЭБС Лань <http://e.lanbook.com/books>
2. Открытый университет Интуит.ру <http://intuit.ru>;
3. <http://www.intuit.ru/studies/courses/2260/156/info>
4. <http://www.intuit.ru/studies/courses/66/66/info>
5. Богословский Н.Н. Программирование на С++ Электронный ресурс: учебно-методический комплекс. Томск : [ИДО ТГУ] , 2015 URL: <http://moodle.tsu.ru/course/view.php?id=431>
6. Расширяемый язык разметки (XML) 1.0 (вторая редакция). <http://citforum.ru/internet/xmlspec/index.shtml>
7. Пространства имен в XML. <http://citforum.ru/internet/xnamesps/index.shtml>
8. Язык XML - практическое введение, Александр Печерский <http://citforum.ru/internet/xml/index.shtml>

13. Перечень информационных технологий

а) лицензионное и свободно распространяемое программное обеспечение:

– Microsoft Office Standart 2013 Russian: пакет программ. Включает приложения: MS Office Word, MS Office Excel, MS Office PowerPoint, MS Office On-eNote, MS Office Publisher, MS Outlook, MS Office Web Apps (Word Excel MS PowerPoint Outlook);

- интегрированная среда для работы на C/C++.
- публично доступные облачные технологии (Google Docs, Яндекс диск и т.п.).

б) информационные справочные системы:

- Электронный каталог Научной библиотеки ТГУ – <http://chamo.lib.tsu.ru/search/query?locale=ru&theme=system>
- Электронная библиотека (репозиторий) ТГУ – <http://vital.lib.tsu.ru/vital/access/manager/Index>
- ЭБС Лань – <http://e.lanbook.com/>
- ЭБС Консультант студента – <http://www.studentlibrary.ru/>
- Образовательная платформа Юрайт – <https://urait.ru/>
- ЭБС ZNANIUM.com – <https://znanium.com/>
- ЭБС IPRbooks – <http://www.iprbookshop.ru/>

14. Материально-техническое обеспечение

Аудитории для проведения занятий лекционного типа.

Аудитории для проведения занятий семинарского типа, индивидуальных и групповых консультаций, текущего контроля и промежуточной аттестации.

Помещения для самостоятельной работы, оснащенные компьютерной техникой и доступом к сети Интернет, в электронную информационно-образовательную среду и к информационным справочным системам.

15. Информация о разработчиках

Старченко Александр Васильевич, профессор, доктор физико-математических наук.

Гольдин Виктор Данилович, старший преподаватель

Зюзьков Валентин Михайлович, доцент, кандидат физико-математических наук

Лаева Валентина Ивановна, старший преподаватель

Данилкин Евгений Александрович, доцент, кандидат физико-математических наук

Шельмина Елена Александровна, доцент, кандидат физико-математических наук

Фазлиев Александр Зарипович, доцент, кандидат физико-математических наук

Михайлов Михаил Дмитриевич, старший преподаватель