|  |
| --- |
| МИРЭА_ЭМБЛЕМА_приказ |
| МИНОБРНАУКИ РОССИИ |
| Федеральное государственное бюджетное образовательное учреждение высшего образования  **«Московский технологический университет»**  **МИРЭА** |

|  |  |  |  |
| --- | --- | --- | --- |
|  | **УТВЕРЖДАЮ**  Директор  Института информационных технологий  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_А.С. Зуев  «\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2017 г. |  | **СОГЛАСОВАНО**  Учебно-методический совет  Институтакибернетики  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Т.С. Хачлаев  «\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2016 г. |

**РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ**

**Б1.Б.16 «Структура и алгоритмы обработки данных»**

Направление подготовки

**09.03.04 «Программная инженерия»**

Профиль подготовки

**«Корпоративные информационные системы»**

Квалификация выпускника

**Бакалавр**

Форма обучения

**Очная**

Москва 2017

1. **Цели освоения дисциплины**

Дисциплина «Структура и алгоритмы обработки данных» имеет своей целью способствовать формированию у обучающихся общепрофессионалных ОПК-4, и профессиональных компетенций ПК-1в соответствии с требованиями ФГОС ВО по направлению подготовки бакалавров 09.03.04 «Программная инженерия» с учетом специфики профиля подготовки – «Корпоративные информационные системы».

1. **Место дисциплины в структуре основной образовательной программы**

Дисциплина «Структура и алгоритмы обработки данных» является дисциплиной базовой части блока «Дисциплины» учебного плана направления подготовки бакалавров 09.03.04 «Программная инженерия» с профилем подготовки – «Корпоративные информационные системы».

Общая трудоемкость дисциплины составляет 8 зачетных единиц (288 акад. часа.).

Для освоения дисциплины «Структура и алгоритмы обработки данных» обучающиеся должны обладать знаниями, умениями и навыками, полученными в результате формирования и развития компетенций в следующих дисциплинах и практиках:

**ОПК-4** способностью осуществлять поиск, хранение, обработку и анализ информации из различных источников и баз данных, представлять ее в требуемом формате с использованием информационных, компьютерных и сетевых технологий

* Анализ сложности алгоритмов (2 семестр)

**ПК-1** (готовность применять основные методы и инструменты разработки программного обеспечения):

* Программирование (1 семестр)
* Объектно-ориентированное программирование (3 семестр)
* Введение в программную инженерию (1 семестр)
* Анализ сложности алгоритмов (2 семестр)

Освоение дисциплины «Структура и алгоритмы обработки данных» является необходимым для изучения последующих дисциплин в рамках дальнейшего формирования и развития следующих компетенций:

**ОПК-4** способностью осуществлять поиск, хранение, обработку и анализ информации из различных источников и баз данных, представлять ее в требуемом формате с использованием информационных, компьютерных и сетевых технологий

* Web-программирование (4 семестр)
* Базы данных (5 семестр)
* Основы сетевых технологий (5,6 семестр)
* Архитектура вычислительных систем (7 семестр)
* Функциональное и логическое программирование
* Хранилища данных (7 семестр)
* Распределенные базы данных (7исеместр)
* Интеллектуальный анализ данных (7 семестр)
* Практика по получению профессиональных умений и опыта профессиональной деятельности (2, 3, 4, 5, 6, 7 семестры)
* Преддипломная практика (8 семестр)
* Государственная итоговая аттестация (8 семестр)

**ПК-1** (готовность применять основные методы и инструменты разработки программного обеспечения):

* Объектно-ориентированное программирование (3 семестр)
* Структура и алгоритмы обработки данных (3, 4 семестр)
* Web-программирование ( 4 семестр)
* Системы искусственного интеллекта (5,6семестр)
* Компьютерная графика (5 семестр)
* Системный анализ (6 семестр)
* Технологии программирования (6 семестр)
* Параллельное программирование (6 семестр)
* Технологии разработки программного обеспечения (6,7 семестр)
* Современные технологии программирования (7 семестр)
* Проектирование информационных систем (7, 8семестр)
* Практика по получению профессиональных умений и опыта профессиональной деятельности(3,4,5,6,7,8 семестр)
* Преддипломная практика (8 семестр)
* Государственная итоговая аттестация (8 семестр)

1. **Планируемые результаты обучения по дисциплине, соотнесенные с планируемыми результатами освоения программы бакалавриата (компетенциями выпускников)**

|  |  |
| --- | --- |
| Формируемые компетенции  (код и название компетенции, уровень освоения – при наличии в карте компетенции) | Планируемые результаты обучения по дисциплине (модулю), характеризующие этапы формирования компетенций |
| **ОПК-4** способность осуществлять поиск, хранение, обработку и анализ информации из различных источников и баз данных, представление информации в требуемом формате с использованием информационных, компьютерных и сетевых технологий | **Знать** базовые структуры и технологии создания не стандартных структур хранения данных, используемых для хранения, оптимизации поиска данных в больших наборах данных: линейные списки, бинарное дерево поиска, сбалансированное бинарное дерево поиска, хеш таблица, граф, файл. Алгоритмы поиска данных, алгоритмы модификации структур данных Возможности стандартных библиотек языков высшего уровня. |
| **Уметь** проводить анализ данных задачи и выбор Структура (базовой или разработанной пользователем) данных для оптимизации алгоритмов решения задач, связанных с хранением, поиском данных |
| **Владеть** навыками практической реализации структур данных в задачах обработки больших наборов данных с применением объектно-ориентированного программирования; навыками реализации структур данных с использованием класса – шаблон; навыками обработки файлов данных с использованием потокового ввода и вывода; навыками применения программных шаблонов и библиотек в задачах обработки больших наборов данных. |
| **ПК-1** (готовность применять основные методы и инструменты разработки программного обеспечения) | **Знать** методику реализации рекурсивных алгоритмов, метод «разделяй и властвуй», метод ветвей и границ, метод динамического программирования; знать технологию использования элементов программной библиотеки. |
| **Уметь** применять методику рекурсивных алгоритмов, методы «разделяй и властвуй», метод ветвей и границ, метод динамического программирования при разработке программного обеспечения |
| **Владеть** практическими навыками применения основных базовых типов данных, типов данных созданных пользователем, шаблонов STL при разработке программного обеспечения обработки больших наборов данных в среде Visual Studio |

1. **Содержание дисциплины**

Общая трудоемкость дисциплины составляет 8 зачетных единиц (288 акад. часа.).

4.1. Распределение объема и содержания дисциплины (модуля) по разделам, семестрам, видам учебной работы и формам контроля

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| № раздела | Семестр | Неделя семестра | Объем (в акад. час.) | | | | | | | Формы текущего контроля успеваемости *(по неделям семестра)*  Формы промежуточной аттестации *(по семестрам)* |
| Всего | Контактная работа (по видам учебных занятий) | | | | СР | Контроль |
| Всего | ЛК | ЛБ | ПР |
| 1 | 4 | 1 | 9 | 4 | 2 |  | 2 | 5 |  | Практическое задание |
| 2 | 4 | 2 | 6 | 4 | 2 |  | 2 | 2 |  | Устное собеседование |
| 2 | 4 | 3-4 | 13 | 8 | 4 |  | 4 | 5 |  | Практическое задание |
| 3 | 4 | 5-6 | 13 | 8 | 4 |  | 4 | 5 |  | Практическое задание |
| 3 | 4 | 6 | 8 | 2 |  | 2 |  | 6 |  | Защита лабораторной работы 1 |
| 4 | 4 | 7-8 | 10 | 8 | 4 |  | 4 | 2 |  | Устное собеседование |
| 4 | 4 | 9-10 | 13 | 8 | 4 |  | 4 | 5 |  | Практическое задание |
| 5 | 4 | 10 | 8 | 2 |  | 2 |  | 6 |  | Защита лабораторной работы 2 |
| 5 | 4 | 11-12 | 13 | 8 | 4 |  | 4 | 5 |  | Практическое задание |
| 6 |  | 13-14 | 13 | 8 | 4 |  | 4 | 5 |  | Практическое задание |
| 6 |  | 14 | 8 | 2 |  | 2 |  | 6 |  | Защита лабораторной работы 3 |
| 7 |  | 15 | 9 | 4 | 2 |  | 2 | 5 |  | Практическое задание |
| 7 |  | 16 | 12 | 6 | 2 | 2 | 2 | 6 |  | Защита лабораторной работы 4 |
| По материалам 3 семестра | | | 135 | 68 | 32 | 8 | 32 | 63 | 9 | Защита курсового проекта |
| По материалам всего 3 семестра | | | 144 | 68 | 32 | 8 | 32 | 63 | 36 | Экзамен |
| *Всего в 3 семестре:* | | | *180* | *68* | *32* | *8* | *32* | *63* | *45* |  |
| 8 | 3 | 1 | 2 | 2 | 2 |  |  |  |  | Устное собеседование |
| 8 | 3 | 2 | 8 | 4 | 2 |  | 2 | 4 |  | Устное собеседование |
| 9 | 3 | 3 | 2 | 2 |  | 2 |  |  |  | Защита лабораторной работы 1 |
| 9 | 3 | 4 | 8 | 4 | 2 |  | 2 | 4 |  | Устное собеседование |
| 10 | 3 | 5 | 4 | 4 |  | 4 |  |  |  | Защита лабораторной работы 2 |
| 10 | 3 | 6 | 8 | 4 | 2 |  | 2 | 4 |  | Устное собеседование |
| 11 | 3 | 7 | 2 | 2 |  | 2 |  |  |  | Защита лабораторной работы 3 |
| 11 | 3 | 8 | 2 | 2 | 2 |  |  |  |  | Устное собеседование |
| 12 | 3 | 8 | 6 | 2 |  |  | 2 | 4 |  | Практическое задание |
| 12 | 3 | 9 | 2 | 2 |  | 2 |  |  |  | Защита лабораторной работы 4 |
| 13 | 3 | 10 | 8 | 4 | 2 |  | 2 | 4 |  | Устное собеседование |
| 13 | 3 | 11 | 2 | 2 |  | 2 |  |  |  | Защита лабораторной работы 5 |
| 14 | 3 | 12 | 8 | 4 | 2 |  | 2 | 4 |  | Устное собеседование |
| 14 | 3 | 13 | 2 | 2 |  | 2 |  |  |  | Защита лабораторной работы 6 |
| 15 | 3 | 14 | 8 | 4 | 2 |  | 2 | 4 |  | Устное собеседование |
| 15 | 3 | 15 | 7 | 2 |  | 2 |  | 5 |  | Защита лабораторной работы 7 |
| 15 | 3 | 16 | 2 | 2 |  |  | 2 |  |  | Устное собеседование |
| По материалам семестра | | | 81 | 48 | 16 | 16 | 16 | 33 | 27 | Зачет |
| *Всего в 4 семестре:* | | | *108* | *48* | *16* | *16* | *16* | *33* | *27* |  |
| **Всего:** | | | **288** | **116** | **48** | **24** | **48** | **96** | **72** |  |

4.2. Наименование и содержание разделов дисциплины

|  |  |  |
| --- | --- | --- |
| **№ раздела** | **Наименование раздела** | **Содержание раздела** |
|  | Введение в дисциплину | Задачи дисциплины. Пример оптимального выбора Структура хранения большого объема данных, для уменьшения времени работы программы. Три уровня представления данных (внешний, логический, физический). Понятие типа данных. Понятие Структура данных. Свойства структур данных. Представление данных в памяти компьютера. Иерархия коллекций данных Динамические и статические Структура данных, внешние Структура данных. |
|  | Базовые Структура данных языка С++ | Бинарные операции. Массив, строка, структура, объединение. Указатель. Операции над указателями. Указатель и массив. Косвенный указатель. Функции. Передача структур данных в качестве параметров. |
|  | Абстрактный тип данных | Определение АТД. Реализация АТД. АТД и объектно-ориентированное программирование. Поэтапная разработка программы с использованием АТД. |
|  | Критерии эффективности программы | Асимптотический анализ верхней и средней оценок сложности алгоритмов; сравнение наилучших, средних и наихудших оценок; O-, o-, ω- и θ-нотации; стандартные классы сложности; эмпирические измерения эффективности алгоритмов; накладные расходы алгоритмов по времени и памяти; рекуррентные соотношения и анализ рекурсивных алгоритмов |
|  | Линейные списки и их разновидности | Математическое понятие линейного списка. АТД - линейный список. Способы реализации линейных списков (массив, однонаправленные, двунаправленные связанные списки). Анализ алгоритмов операций над списком для различных реализаций. АТД линейных списков – Очередь, Стек и их способы реализации. Алгоритмы операций: поиск (линейный, бинарный), сортировка со сложностью О(n2) и О(n log(n)). Применение стека и очереди в алгоритмах организации вычисления выражения (Польская постфиксная и префиксные записи выражений). |
|  | Реализация АТД на основе класса С++ | Шаблонные функции. Перегрузка функций. Перегрузка операций. |
|  | Методы разработки алгоритма | Полный перебор. Рекурсивные алгоритмы и их реализация. Метод “разделяй и властвуй”. “Жадные” алгоритм. Метод динамического программирования. Бэктрекинг (перебор с возвратами). Метод ветвей и границ. Эвристический поиск. Поиск по образцу, алгоритмы обработки строк; алгоритмы аппроксимации числовых функций. |
|  | Реализация АТД на основе класса С++ | Определение класса. Конструкторы класса. Деструкторы класса. Перегрузка операций. Шаблонные классы. Абстрактные классы. |
|  | Внешние Структура данных. | Файл. Алгоритмы поиска в файле. Использование индексных файлов. В-дерево и поиск данных в файле. Алгоритмы внешней сортировки. |
|  | Иерархические Структура | Свойства. Иерархические Структура. Основные понятия. АТД дерево. Способы реализации. Алгоритмы обхода: в глубину и ширину. |
|  | Структура данных - бинарное дерево | АТД бинарное дерево. Сбалансированное дерево. Способы реализации. Рекурсивные алгоритмы обхода бинарного дерева. Дерево выражений. |
|  | Структура данных - бинарное дерево поиска | Алгоритмы поиска и иерархические Структура данных. Бинарное дерево поиска. Бинарное сбалансированное дерево. AVL дерево. Алгоритмы вставки и удаления. Красно-черные деревья. Анализ алгоритмов поиска с использованием бинарного дерева поиска и AVL дерева. |
|  | Хеш таблицы | Алгоритмы поиска. Организация быстрого поиска с использованием хеш таблицы. Хэш функции. Коллизия. Способы разрешения коллизий: цепное хещирование; открытый адрес. Поиск с возвратом. Анализ алгоритма поиска с использованием хеш таблицы. |
|  | Структура данных - граф. | Направленные и ненаправленные графы. Сети. АТД-граф. Способы реализации графа в памяти (списки и матрицы смежности). Алгоритмы обхода графа в ширину и в глубину.. Алгоритмы поиска кратчайших путей в графе между двумя заданными вершинами (Дейкстры).  Транзитивное замыкание (алгоритм Флойда). Алгоритмы построения минимального покрывающего дерева (алгоритмы Прима и Крускала). Топологическая сортировка |
|  | Стандартная библиотека шаблонов и Структура данных | Алгоритмы STL. Структура STL. |

* 1. Лабораторные работы (ЛБ)

|  |  |  |  |
| --- | --- | --- | --- |
| № п/п | № раздела дисциплины | Наименование лабораторных работ | Трудоемкость (в акад. час.) |
|  | 6,7 | Реализация АТД задачи средствами ООП языка С++ | 2 |
|  | 10,11 | Применение бинарного дерева поиска для поиска данных в файлах | 2 |
|  | 12 | Применение хеш таблицы для поиска данных в файлах | 2 |
|  | 14 | Применение аппарата шаблонных классов в реализации АТД задачи | 2 |
| *Всего во 3 семестре:* | | | *8* |
|  | 2,3 | Реализация алгоритмов через аппарат функций языка С++ | 2 |
|  | 2,3 | Базовые Структура данных языка С++. Указатель и массив статический и динамический. | 4 |
|  | 2,3 | Базовые Структура данных языка Си и С++ - строка | 2 |
|  | 2,3 | Использование битовых операций языка Си при обработке структур данных | 2 |
|  | 5 | Линейный однонаправленный список | 2 |
|  | 5 | Линейные списки – стек, очередь и их применение в решении задач. | 2 |
|  | 5 | Линейный двунаправленный список | 2 |
| *Всего в 4 семестре:* | | | *16* |
| **Всего:** | | | **24** |

4.4. Практические занятия (ПР)

|  |  |  |  |
| --- | --- | --- | --- |
| № п/п | № раздела дисциплины | Тематика практических занятий | Трудоемкость (в акад. час.) |
|  | 3 | Тип данных, основные операторы языка С++. Реализация стандартного ввода и вывода в стиле Си и С++. Упражнения по разработке программ в Си и функций. | 2 |
|  | 3 | Указатели в Си. Массив данных в С++. Передача массива в качестве параметра функции. | 2 |
|  | 3 | Обработка текстовой информации в С++. Функции библиотеки string.h и их использование при разработке программ. | 2 |
|  | 3 | Побитовые операции в С++. Упражнения по реализация алгоритмов, использующих побитовые операции. Тип данных struct, его особенности, применение при создании элементов нестандартных структур данных | 2 |
|  | 4 | Упражнения по оценке сложности алгоритмов обработки структур данных. | 2 |
|  | 5 | Упражнения по созданию и обработке линейных списков | 2 |
| *Всего в 1 семестре:* | | | *16* |
|  | 9 | Рекурсивные алгоритм и их реализация в С++. | 2 |
|  | 10 | Применение методов «Разделяй и властвуй», «Перебор с возвратом» по разработке алгоритмов. | 2 |
|  | 11 | Создание класса С++ и его применение в приложении. | 4 |
|  | 12 | Разработка программ по обработке потоковых файлов данных. | 4 |
|  | 13 | Реализация алгоритмов обхода сильноветвящегося дерева, способы реализации дерева в памяти компьютера. | 4 |
|  | 13 | Реализация основных операций над рекурсивной структурой данных – бинарное дерево. | 4 |
|  | 14 | Применение бинарного дерева поиска к поиску данных в файлах данных. | 4 |
|  | 14 | Применение хеш таблицы для поиска данных в файлах. | 4 |
|  | 15 | Решение задач с применением Структура граф. | 2 |
|  | 15 | Реализация программы, по обработке структур данных, на основе структур данных STL. | 2 |
| *Всего в 2 семестре:* | | | *32* |
| **Всего:** | | | **48** |

1. **Учебно-методическое обеспечение для самостоятельной работы обучающихся по дисциплине**

Виды самостоятельной работы обучающегося, порядок и сроки ее выполнения:

* выполнение курсового проекта; время выдачи задания на курсовой проект – не позднее 4-ой недели 4-го семестра; срок сдачи курсового проекта – не позднее 15-ой недели 4-го семестра; ориентировочный перечень тем курсового проекта:
* Организация прямого доступа к записям в базах данных.
* Создание пользовательского типа данных на основе класса С++ и его использование в приложении.
* Дерево Хаффмана.
* В – дерево.
* AVL – дерево. Применение AVL дерева в задаче проверки орфографии.
* Красно-черное дерево и его применение для поиска данных в файле.
* Мультисписки. Организация и применение.
* Структура данных – очередь с приоритетом и её применение для поиска данных в файле.
* Пирамидальная сортировка. Реализация класса. Демонстрационный пример.
* Граф. Применение класса граф к решению предложенных вами задач.
* Множество. Реализация на основе класса с хранением в битовом векторе.
* Разреженная матрица. Способы представления матрицы. Реализация матрицы выбранным способом.
* Кодирование информации. DES.
* Сравнение алгоритмов поиска, основанных на бинарном дереве поиска и на хеш-таблице, в файле данных
* Подготовка к лекциям, лабораторным работам и практическим занятиям с использованием конспекта лекций, материалов практических занятий и приведенных ниже (п/п.п. 8.1 и 8.2) источников (в течение 3-го и 4-го семестров в соответствии с расписанием занятий);
* оформление отчетов по выполненным лабораторным работам и теоретическая подготовка к их сдаче (в течение 3-го и 4-го семестров в соответствии с расписанием занятий).

Перечень вопросов для проведения текущего контроля и промежуточной аттестации – в соответствии с тематикой разделов дисциплины.

1. **Фонд оценочных средств для проведения текущего контроля успеваемости и промежуточной аттестации обучающихся по дисциплине**

6.1. **Перечень компетенций**, на освоение которых направлено изучение дисциплины «Структура и алгоритмы обработки данных», с указанием этапов их формирования в процессе освоения образовательной программы бакалавриата, представлен в п.3 настоящей рабочей программы.

6.2 **Описание показателей и критериев оценивания** компетенций на различных этапах их формирования, описание шкал оценивая

6.2.1. Показатели и критерии оценивания компетенций, используемые шкалы оценивания

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Элементы компетенций (знания, умения, владения)** | **Показатели оценивания** | **Критерии**  **оценивания** | **Средства**  **оценивания** | **Шкалы**  **оценивания** |
|  |  |  |  |  |
| **Знать**  **(ОПК-4)** | **Знание** базовых структур данных языка С++ и технологий создания не стандартных структур хранения данных, используемых для хранения, оптимизации поиска данных в больших наборах данных: линейные списки, бинарное дерево поиска, сбалансированное бинарное дерево поиска, хеш таблица, граф, файл. Алгоритмы поиска данных, алгоритмы модификации структур данных Возможности стандартной библиотеки шаблонов (STL) языка С++. | Правильность и полнота ответов, глубина понимания вопроса | Текущий контроль:  выполнение устных/ письменных заданий  Промежуточная аттестация:  зачет,  экзамен | Шкала 1 |
| **Уметь**  **(ОПК-4)** | **Умение** проводить анализ данных задачи и выбор Структура (базовой или разработанной пользователем) данных для оптимизации алгоритмов решения задач, связанных с хранением, поиском данных | Правильность выполнения учебных заданий, аргументированность выводов | Текущий контроль:  выполнение устных/ письменных заданий  Промежуточная аттестация:  зачет,  экзамен | Шкала 1 |
| **Владеть**  **(ОПК-4)** | **Владение** практическиминавыками реализации структур данных в задачах обработки больших наборов данных средствами языка С++ с применением объектно-ориентированного программирования; практическими навыками реализации структур данных с использованием класса – шаблон; навыками обработки файлов данных с использованием потокового ввода и вывода; навыками применения STL в задачах обработки больших наборов данных. | Обоснованность и аргументированность выполнения учебной деятельности | Текущий контроль:  выполнение практического задания,  защита лабораторной работы  Промежуточная аттестация:  защита курсового проекта,  зачет,  экзамен | Шкала 2 |
| **Знать**  **(ПК-1)** | **Знать** методику реализации рекурсивных алгоритмов, метод «разделяй и властвуй», метод ветвей и границ, метод динамического программирования; знать теорию объектно-ориентированного программирования языка С++; знать технологию использования элементов библиотеки STL. | Правильность выполнения учебных заданий, аргументированность выводов | Текущий контроль:  выполнение устных/ письменных заданий  Промежуточная аттестация:  зачет,  экзамен | Шкала 1 |
| **Уметь**  **(ПК-1)** | **Уметь** применять методику рекурсивных алгоритмов, методы «разделяй и властвуй», метод ветвей и границ, метод динамического программирования при разработке программного обеспечения | Обоснованность и аргументированность выполнения учебной деятельности | Текущий контроль:  выполнение практического задания,  защита лабораторной работы  Промежуточная аттестация:  защита курсового проекта,  зачет,  экзамен | Шкала 2 |
| **Владеть**  **(ПК-1)** | **Владеть** практическими навыками применения основных базовых типов данных языка С++, типов данных созданных пользователем, шаблонов STL при разработке программного обеспечения обработки больших наборов данных в среде Visual Studio (версия значения не имеет). | Правильность и полнота ответов, глубина понимания вопроса | Текущий контроль:  выполнение устных/ письменных заданий  Промежуточная аттестация:  зачет,  экзамен,  курсовая работа | Шкала 1 |

6.2.2. Описание шкал оценивания степени сформированности элементов компетенций

Шкала 1.Оценка сформированности отдельных элементов компетенций

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Обозначения | | **Формулировка требований к степени сформированности**  **компетенции** | | |
| Цифр. | Оценка |
| **Знать** | **Уметь** | **Владеть** |
| 1 | Неуд. | Отсутствие знаний | Отсутствие умений | Отсутствие навыков |
| 2 | Неуд. | Фрагментарные знания | Частично освоенное умение | Фрагментарное применение |
| 3 | Удовл. | Общие, но не структурированные знания | В целом успешное, но не систематически осуществляемое умение | В целом успешное, но не систематическое применение |
| 4 | Хор. | Сформированные, но содержащие отдельные пробелы знания | В целом успешное, но содержащие отдельные пробелы умение | В целом успешное, но содержащее отдельные пробелы применение навыков |
| 5 | Отл. | Сформированные систематические знания | Сформированное умение | Успешное и систематическое применение навыков |

Шкала 2. Комплексная оценка сформированности знаний, умений и владений

|  |  |  |
| --- | --- | --- |
| Обозначения | | **Формулировка требований к степени сформированности**  **компетенции** |
| Цифр. | Оценка |
| 1 | Неуд. | Не имеет необходимых представлений о проверяемом материале |
| 2 | Удовл.  или неуд. (*по усмотрению преподавателя)* | Знать на уровне о**риентирования**, представлений. Субъект учения знает основные признаки или термины изучаемого элемента содержания, их отнесенность к определенной науке, отрасли или объектам, узнает их в текстах, изображениях или схемах и знает, к каким источникам нужно обращаться для более детального его усвоения. |
| 3 | Удовл. | Знать и уметь на **репродуктивном** уровне. Субъект учения знает изученный элемент содержания репродуктивно: произвольно воспроизводит свои знания устно, письменно или в демонстрируемых действиях. |
| 4 | Хор. | Знать, уметь, владеть на **аналитическом** уровне. Зная на репродуктивном уровне, указывать на особенности и взаимосвязи изученных объектов, на их достоинства, ограничения, историю и перспективы развития и особенности для разных объектов усвоения. |
| 5 | Отл. | Знать, уметь, владеть на **системном** уровне. Субъект учения знает изученный элемент содержания системно, произвольно и доказательно воспроизводит свои знания устно, письменно или в демонстрируемых действиях, учитывая и указывая связи и зависимости между этим элементом и другими элементами содержания учебной дисциплины, его значимость в содержании учебной дисциплины. |

6.3. **Типовые контрольные задания или иные материалы**, необходимые для оценки знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций в процессе освоения образовательной программы.

**Типовые вопросы и задания для текущего контроля** (оценка сформированности элементов (знаний, умений) компетенций ПК-1, ОПК-4 в рамках текущего контроля по дисциплине) по разделам дисциплины

Примеры вопросов по разделу 1:

* расскажите о трех уровнях представления данных в разрабатываемом ПО, определите понятие Структура данных, поясните свойства Структура данных.
* назовите Структура хранения данных в оперативной памяти и приведите их особенности;
* назовите основные виды отношений между элементами в линейных структурах данных; в иерархических; в сетевых;
* какие Структура называют динамическими?

Примеры вопросов по разделу 2:

* приведите определение указателя в языке Си; какие операции можно выполнять над указателем (поясните на примерах); что общего между указателем и статическим массивом и чем они отличаются; как можно создать динамический массив из элементов целого типа; расскажите о двойном указателе;
* объясните, для каких целей в структурах данных можно использовать функцию пользователя; какие функции можно определять в программах С++ и их особенности в применении; какой формат имеет определение функции;
* расскажите о механизме параметров функции в языке С++.

Пример практического задания номер 4 по разделу 2

* Напишите функцию, которая заменяет n битов переменной х, на n самых правых битов переменной y, а остальные биты не изменились. Требования: алгоритм функции оформить в виде одного выражения, выполняющего битовые операции.
* Напишите функцию, которая возвращает значение целочисленного х, «повернутого» вправо на n позиций

**Защита лабораторных работ** (оценка сформированности элементов (знаний, умений) компетенций ПК-1, ОПК-4в рамках текущего контроля по дисциплине) по разделам дисциплины:

Примеры вопросов при защите лабораторной работы №1 (раздел 2):

* для реализации каких алгоритмов используется функция типа void?
* какие три понятия используются при использовании функций;
* из каких точек программы можно вызывать функции?

Примеры вопросов при защите лабораторной работы №6 (раздел 5):

* какую линейную структуру называют очередью;
* какую линейную структуру называют стеком;
* какую линейную структуру называют списком;
* приведите модель «круговой очереди» с реализацией на статическом массиве и механизм выполнения операций вставки элемента в очередь и удаления из очереди.

**Зачет** в 3 семестре выставляется по совокупности результатов текущего контроля по разделам дисциплины.

**Перечень вопросов для подготовки к экзамену** (оценка сформированности компетенции ПК-1, ОПК-4, рамках промежуточной аттестации по дисциплине).

Содержание экзаменационного билета:

1 вопрос – фундаментальная теория;

2 вопрос – прикладная теория (элементы языка С++);

3 вопрос – практическая комплексная задача.

Пример типового экзаменационного билета:

1 вопрос – Структура данных Очередь с приоритетом. Способы реализации.

2 вопрос – Указатель. Виды указателей. Правила прочтения типа указателя. Указатель и массив для хранения слов из текста.

3 вопрос – Разработайте функцию, которая выводит элементы бинарного дерева на экран, обходя дерево справа налево и сохраняя уровни дерева.

Комплекты контрольных заданий Фонда оценочных средств по дисциплине представлены в составе УМК дисциплины.

6.4. **Методические материалы, определяющие процедуры оценивания** знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций.

Процедуры и средства оценивания элементов компетенций по дисциплине

«Структура и алгоритмы обработки данных»

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Процедура проведения** | **Средство оценивания** | | | | | | |
| Текущий контроль | | | | Промежуточный контроль | | |
| Выполнение устных заданий | Выполнение письменных заданий | Выполнение практических заданий | Защита лабораторных работ | Защита курсового проекта | Зачет | Экзамен |
| Продолжительность контроля | По усмотрению преподавателя | По усмотрению преподавателя | По усмотрению преподавателя | По усмотрению преподавателя | В соответствии с принятыми нормами времени | В соответствии с принятыми нормами времени | В соответствии с принятыми нормами времени |
| Форма проведения  контроля | Устный опрос | Письменный опрос | Письменный опрос | Устная защита | В устной форме | В устной форме | В письменной форме |
| Вид проверочного задания | Устные вопросы | Письменные задания | Практические задания | Устные вопросы | задание на курсовую работу | вопросы к зачету | экзаменационный билет |
| Форма отчета | Устные ответы | Ответы в письменной форме | Ответы в письменной форме | Ответы в устной форме | Защита курсовой проекта | Ответы в письменной форме (устное собеседование) | Ответы в письменной форме |
| Раздаточный материал | Нет | Справочная литература | Справочная литература | Справочная литература | Справочная литература | Справочная литература | Справочная литература |

1. **Методические указания для обучающихся по освоению дисциплины**

Дисциплина «Структура и алгоритмы обработки данных» предусматривает лекции, практические занятия и лабораторные работы, а также выполнение и защиту курсового проекта. Успешное изучение дисциплины требует посещения лекций, активной работы на практических занятиях, выполнения лабораторных работ и учебных заданий преподавателя, ознакомления с основной и дополнительной литературой.

В ходе лекций преподаватель излагает и разъясняет основные, наиболее сложные понятия темы, а также связанные с ней теоретические и практические проблемы, дает рекомендации на практическое занятие и указания на самостоятельную работу.

При подготовке к лекционным занятиям студентам необходимо:

перед очередной лекцией необходимо просмотреть конспект материала предыдущей лекции. При затруднениях в восприятии материала следует обратиться к основным литературным источникам. Если разобраться в материале опять не удалось, то обратитесь к лектору (по графику его консультаций) или к преподавателю на практических занятиях.

Практические занятия завершают изучение наиболее важных тем учебной дисциплины. Они служат для закрепления изученного материала, развития умений и навыков подготовки докладов, сообщений, приобретения опыта устных публичных выступлений, ведения дискуссии, аргументации и защиты выдвигаемых положений, а также для контроля преподавателем степени подготовленности студентов по изучаемой дисциплине.

При подготовке к практическому занятию студенты имеют возможность воспользоваться консультациями преподавателя.

При подготовке к практическим занятиям студентам необходимо:

приносить с собой рекомендованную преподавателем литературу к конкретному занятию;

до очередного практического занятия по рекомендованным литературным источникам проработать теоретический материал, соответствующей темы занятия;

в начале занятий задать преподавателю вопросы по материалу, вызвавшему затруднения в его понимании и освоении при решении задач, заданных для самостоятельного решения;

в ходе семинара давать конкретные, четкие ответы по существу вопросов;

на занятии доводить каждую задачу до окончательного решения, демонстрировать понимание проведенных расчетов (анализов, ситуаций), в случае затруднений обращаться к преподавателю.

Студентам, пропустившим занятия (независимо от причин), не имеющие письменного решения задач или не подготовившиеся к данному практическому занятию, рекомендуется не позже чем в 2-недельный срок явиться на консультацию к преподавателю и отчитаться по теме, изучавшейся на занятии. Студенты, не отчитавшиеся по каждой не проработанной ими на занятиях теме к началу зачетной сессии, упускают возможность получить положенные баллы за работу в соответствующем семестре.

Методические указания по выполнению лабораторных работ приведены в составе программы бакалавриата.

Методические указания по выполнению и защите курсового проекта приведены в составе программы бакалавриата.

1. **Ресурсное обеспечение дисциплины**

8.1. Основная и дополнительная учебная литература, необходимая для освоения дисциплины

а) основная литература:

1. Вирт Н. Алгоритмы и структуры данных : Новая версия для Оберона + CD: Пер. с англ. / Н. Вирт. — М.: ДМК Пресс, 2014. — 272 с.: ил. — (Классика программирования).
2. Ахо А. В.[Структура данных и алгоритмы [Текст] / Ахо А.В., Хопкрофт Д., Ульман Дж.Д.; Пер. с англ. — М.: Вильямс, 2015. — 391 с.: ил. — Библиогр.: с. 377-382](http://library.mirea.ru/mgupi/43795)
3. Бабенко М.А., Левин М.В. Введение в теорию алгоритмов и структур данных:. Издательство: Московский центр непрерывного математического образования:, 2016

[e.lanbook.com/](http://e.lanbook.com/)

1. Стивен Прата Язык программирования C++. Лекции и упражнения, 6-е издание Автор: Издательство: Вильямс:,Год: 2014
2. Герберт Шилдт С++: методики программировании Шилдта.- М.: ИД Вильямс, 2015 г.

б) дополнительная литература:

1. Круз Р. Структуры данных и проектирование программ [Текст] / Р.Круз; Пер. с англ. — М.: БИНОМ. Лаборатория знаний, 2015. — 765 с.: ил. — (Программисту).
2. Алексеев В.Е. Графы и алгоритмы. Структуры данных. Модели вычислений : учебник для вузов / В. Е. Алексеев, В.А. Таланов. — М.: ИНТУИТ, 2013. — 319 с.: ил. — (Основы информационных технологий).
3. Лафоре Роберт Структуры данных и алгоритмы в Java : Пер. с англ. / Р. Лафоре. — СПб.: Питер, 2013. — 702 c.: ил. — (Классика computer science).
4. Кнут Д. Э. Искусство программирования [Текст]. Т1. / Дональд Э. Кнут. — М.: Вильямс, 2014. — 712 с.

8.3. Информационные технологии, используемые при осуществлении образовательного процесса по дисциплине, включая перечень программного обеспечения и информационных справочных систем

Среда разработки программного обеспечения Visual Studio 2013 и выше.

8.4. Материально-техническая база, необходимая для осуществления образовательного процесса по дисциплине

* учебная аудитория, оснащенная презентационным оборудованием и учебной доской для проведения лекций и практических занятий;
* компьютерный класс с наличием рабочих мест не менее 15 для выполнения лабораторных работ.

Программа составлена с учетом рекомендаций по направлению 09.03.04 «Программная инженерия» по профилю подготовки «Корпоративные информационные системы».

Автор(ы): доцент, к.т.н., Андрианова Е.Г.

Программа одобрена на заседании кафедры КИС

(протокол № 1 от 30 июня 2017 г.)

Заведующий кафедрой КИС Петров А.Б.

|  |  |
| --- | --- |
| **Вопросы к разделам дисциплины СиАОД для текущего контроля** | |
| № раздела | Вопросы |
|  | Какие данные представлены на внешнем уровне?  Как данные представлены на логическом уровне?  Какие существуют способы представления структур данных на физическом уровне?  Что определяет тип данных?  Введите понятие структуры данных.  Какая структура данных называется однородной(неоднородной)?  Какая структура называется линейной?  Какая структура называется нелинейной?  Какие структуры данных называют динамическими?  Какие структуры данных называют статическими?  Какие структуры данных называют внешними?  Как связаны представления данных на внешнем, логическом, физическом уровнях ?  Векторное и связанное хранение данных. |
|  | Как установить нужный бит в 1 или в 0?  Как выполнить инвертирование заданных битов переменной?  Какая арифметическая операция выполниться при выполнении операции <<?  Как объявить типизированный указатель?  Как объявить указатель без типа?  Что такое указатель?  Какие арифметические операции можно выполнять над указателями?  Как объявить статический массив?  Как храниться в памяти массив?  Что общего у массива и указателя?  Как создать динамический массив?  Как проверить наличие свободной динамической памяти, достаточного объема?  Как увеличить (уменьшить) размер массива на несколько ячеек средствами языка Си?  Можно ли присвоить значение одного массива другому? Ответ объясните.  Чем строка языка Си отличается от массива?  Алгоритмы внутренних сортировок.  Алгоритм линейного поиск в массиве.  Можно ли присвоить одной строковой переменной значение другой строковой переменной?  Как можно сравнить строковые значения?  Когда требуется использовать структуру struct?  В каких задачах используются битовые поля структуры?  Как формируется имя поля в структуре?  Как формируется имя поля через указатель на структуру?  В каких задачах для представления данных используется структура объединение?  Как объявить статический многомерный массив?  Что определяет указатель: int \*\*x; char \*\*m;  Приведите фрагмент кода по созданию динамического двумерного массива из n строк с переменной длиной строки.  Приведите заголовок функции , которая возвращает: а) одномерный массив; б) двумерный массив.  Приведите заголовок функции, которая предназначена для сортировки динамического массива.  Приведите вызов функции, заголовок которой оформлен так: void swap(int \*x, int \*y);  Приведите вызов функции, заголовок которой оформлен так: void swap(int &x, int &y); |
|  | Что определяет абстрактный тип данных (АТД)?  Назначение АТД?  Связь между АТД, математической моделью и структурой данных языка программирования?.  Определите данные задачи: школьный журнал – как программное обеспечение хранит отдельно данные о школьниках и данные по о полученных ими оценках. Надо разработать программу, которая определяет, кто из школьников на данный момент не имеет оценок.  Связь между неформальным алгоритмом, программой на псевдокоде, программой на языке программирования.  Разработайте АТД для задачи п.8.  Какую математическую модель удобно использовать для задачи п.8?  Поэтапная разработка программы с использованием АТД.  Реализация АТД. |
|  | Какие факторы влияют на время выполнения алгоритма?  Расскажите об оценке временной сложности алгоритма, по обработке структур данных размера n.  Расскажите о зависимости временной сложности алгоритма от самих данных?  Введите понятии об оценке времени выполнения алгоритма по нотации «Большое О».  Приведите правила быстрой оценки временной сложности алгоритма с помощью «Большого О».  Какие элементарные функции используются при оценке сложности по нотации «Большое О»?  Введите понятие доминирующей функции при оценке сложности с применением нотации «Большое О».  Выполните анализ «Большое О». для каждой из функций: n+5; n2+6n+7.  Выведите функцию оценки времени выполнения алгоритма линейного поиска в массиве из n элементов в худшем случае.  Выведите функцию оценки времени выполнения алгоритма бинарного поиска в массиве из n элементов.мя  Покажите что время вычисления 2n+n3 имеет порядок роста О(2n).  Приведите оценки времени по нотации выполнения циклов по нотации «Большое О» как функции от n:   1. for(s=0.0, i=0; i<n; i++) s+=a[i]\*b[i]; 2. for(i=0; i<n; i++) if (a[i]==k) return 1; return 0; 3. for(i=0; i<n; i++)   for(j=0; j<n; j++) b[i][j]\*=c; |
|  | Дайте определение линейного списка на языке математики.  Определите операции над структурой данных линейный список.  Структура узла линейного однонаправленного списка.  Запишите на псевдокоде алгоритм вставки нового узла в конец однонаправленного списка.  Запишите на псевдокоде алгоритм вставки нового узла в вершину однонаправленного списка.  Запишите на псевдокоде алгоритм вставки нового узла в любую позицию однонаправленного списка.  Представьте структуру узла двунаправленного списка.  Запишите на псевдокоде алгоритм вставки нового узла в любую позицию двунаправленного списка.  Разработайте АТД линейный список с выполнением базовых операций  Приведите примеры способов реализации линейного списка.  Дайте определение линейному списку Стек.  Разработайте АТД для структуры Стек.  Приведите алгоритм вычисления значения арифметического выражения в инфиксной форме с применением АТД Стек.  Приведите алгоритм формирования польской постфиксной записи выражения с применением структуры данных Стек.  Дайте определение линейному списку Очередь.  Дайте определение линейному списку Дек.  Приведите алгоритмы операций вставки и удаления узлов из круговой очереди  Приведите алгоритм вычисления значения выражения, записанного в префиксной форме, используя АТД Очередь и Стек.. |
|  | Какие функции считаются перегруженными?  Какие функции называют шаблонными?  Что означает понятие «перегруженная операция».  Назовите особенность реализации АТД с использованием класса С++.  Структура класса в С++.  Приведите касс для реализации линейного упорядоченного списка, информационная часть узла которого целое число. Реализация методов-операций через функции.  Какие классы называют абстрактными?  Какие классы называют шаблонными?  Приведите шаблонный касс для реализации линейного списка.. Реализация методов – операций с использованием перегрузки операций.  Приведите примеры создания объектов на основе шаблонного класса – линейного списка, информационная часть которых: простого типа; структура.  Что называют контейнером в ООП?  Какие возможности предоставляет STL разработчикам ПО?  Реализуйте операции над линейным списком : создание, вывод, поиск, используя шаблон list из STL. Примените к объектам ,в которых информационная часть: простого типа; структура. |
|  | Какие процессы называют рекурсивными?  Какие функции называют рекурсивными?  Приведите основные правила реализации рекурсивных процессов посредством рекурсивных функций.  Какую рекурсию называют линейной?  Что означают понятия: глубина рекурсии; шаг рекурсии.  Реализуйте на языке Си алгоритм бинарного поиска в массиве.  Расскажите об организации алгоритмов полного перебора. Перечислите методы, обеспечивающие сокращение выполнения полного перебора.  Приведите пример метода перебор с возвратом для решения задачи: поиск выхода из лабиринта.  Суть метода динамического программирования. |
|  | Что считается логическим файлом?  Что считается физическим файлом?  Какова структура текстового файла?  Какова структура бинарного файла?  Что представляет собой имя файловой переменной в С++?  Объясните операцию открытия файла.  Как организуется прямой доступ к записям файла?  Алгоритм сортировки данных в фале – метод слияния.  Понятие индексного файла. |
|  | Отношения между элементами в иерархической структуре.  Что представляет собой путь к элементу в дереве?  Дайте определение иерархической структуры данных (сильно ветвящееся дерево).  Основные понятия: уровень, степень дерева, лист дерева, высота дерева, корень дерева.  Приведите АТД Tree.  Реализуйте на абстрактном уровне алгоритмы обхода сильно ветвящегося дерева в глубину, используя АТД.  Алгоритмы обхода сильно ветвящегося дерева в ширину.  Способы реализации: на одномерном массиве, двумерном массиве, списке правых сыновей и правых братьев. |
|  | Определение бинарного дерева.  Сбалансированное бинарное дерево.  Способы реализации.  Привести список меток узлов дерева при обходе в глубину каждым из алгоритмов.  Приведите характеристики дерева выражений.  Вывести префиксную, постфиксную и инфиксную формы выражений. |
|  | Определение бинарного дерева поиска.  Применение алгоритмов обхода бинарного дерева к обходу дерева поиска.  Рекурсивные алгоритмы определения высоты дерева, количества листьев.  Привести алгоритм рекурсивного поиска узла в бинарном дереве поиска.  Привести алгоритм удаления узла с одним поддеревом.  Определение бинарного сбалансированного дерева поиска.  Как осуществляется выполнение операция вставки узла в AVL дерева?  Алгоритмы поворотов: левый, правый, двойной и их особенности.. |
|  | Структура данных – таблица (словарь).  Назначение и структура хеш-таблицы.  Суть использования хеширования.  Что является результатом хеш - функции и ее аргументом?  Приведите пример, объясняющий понятие коллизии.  Какие вы знаете способы устранения коллизий?  Всегда ли таблица может обеспечить прямой доступ к элементу?  Какие методы позволяют поддерживать прямой доступ в каждой из реализаций?  В каком способе устранения коллизий используется понятие Кластер и что оно означает?  Расскажите о методе двойного хеширования. |
|  | Приведите определение структуры данных граф.  Какой граф называют направленным?  Какой граф называют ненаправленным?  Какой граф называют сетью?.  Способы реализации графа в памяти: списки инцидентности; матрица смежности.  Структуры при реализации алгоритма обхода графа в глубину.  Приведите алгоритм поиска кратчайшего пути в графе между двумя заданными вершинами (Дейкстры).  Транзитивное замыкание (алгоритм Флойда).  Алгоритмы построения минимального покрывающего дерева (алгоритмы Прима и Крускала).  Алгоритм топологической сортировки. |
|  | Понятие контейнера в ООП.  Понятие итератора в ООП.  Как подключить STL в программу?  Какую структуру реализует контейнер vector STL?  Какой контейнер реализует хеш - таблицу?  Какой контейнер реализует линейный список? |

|  |
| --- |
| **ОПК-4** способностью осуществлять поиск, хранение, обработку и анализ информации из различных источников и баз данных, представлять ее в требуемом формате с использованием информационных, компьютерных и сетевых технологий |
| **Знать**  Какие данные представлены на внешнем уровне?  Как данные представлены на логическом уровне?  Какие существуют способы представления структур данных на физическом уровне?  Что определяет тип данных?  Введите понятие структуры данных.  Какая структура данных называется однородной(неоднородной)?  Какая структура называется линейной?  Какая структура называется нелинейной?  Какие структуры данных называют динамическими?  Какие структуры данных называют статическими?  Какие структуры данных называют внешними?  Как связаны представления данных на внешнем, логическом, физическом уровнях ?  Векторное и связанное хранение данных.  Что считается логическим файлом?  Что считается физическим файлом?  Какова структура текстового файла?  Какова структура бинарного файла?  Что представляет собой имя файловой переменной в С++?  Объясните операцию открытия файла.  Как организуется прямой доступ к записям файла?  Алгоритм сортировки данных в фале – метод слияния.  Понятие индексного файла.  Отношения между элементами в иерархической структуре.  Что представляет собой путь к элементу в дереве?  Дайте определение иерархической структуры данных (сильно ветвящееся дерево).  Основные понятия: уровень, степень дерева, лист дерева, высота дерева, корень дерева.  Приведите АТД Tree.  Реализуйте на абстрактном уровне алгоритмы обхода сильно ветвящегося дерева в глубину, используя АТД.  Алгоритмы обхода сильно ветвящегося дерева в ширину.  Способы реализации: на одномерном массиве, двумерном массиве, списке правых сыновей и правых братьев.  Определение бинарного дерева.  Сбалансированное бинарное дерево.  Способы реализации.  Привести список меток узлов дерева при обходе в глубину каждым из алгоритмов.  Приведите характеристики дерева выражений.  Вывести префиксную, постфиксную и инфиксную формы выражений.  Определение бинарного дерева поиска.  Определение бинарного сбалансированного дерева поиска.  Как осуществляется выполнение операция вставки узла в AVL дерева?  Алгоритмы поворотов: левый, правый, двойной и их особенности.  Структура данных – таблица (словарь).  Назначение и структура хеш-таблицы.  Суть использования хеширования.  Что является результатом хеш - функции и ее аргументом?  Приведите пример, объясняющий понятие коллизии.  Какие вы знаете способы устранения коллизий?  Всегда ли таблица может обеспечить прямой доступ к элементу?  Какие методы позволяют поддерживать прямой доступ в каждой из реализаций?  В каком способе устранения коллизий используется понятие Кластер и что оно означает?  Расскажите о методе двойного хеширования.  Понятие контейнера в ООП.  Понятие итератора в ООП.  Как подключить STL в программу?  Какую структуру реализует контейнер vector STL?  Какой контейнер реализует хеш - таблицу?  Какой контейнер реализует линейный список? |
| **Уметь** проводить анализ данных задачи и выбор структуры (базовой или разработанной пользователем) данных для оптимизации алгоритмов решения задач, связанных с хранением, поиском данных  Какие факторы влияют на время выполнения алгоритма?  Расскажите об оценке временной сложности алгоритма, по обработке структур данных размера n.  Расскажите о зависимости временной сложности алгоритма от самих данных?  Введите понятии об оценке времени выполнения алгоритма по нотации «Большое О».  Приведите правила быстрой оценки временной сложности алгоритма с помощью «Большого О».  Какие элементарные функции используются при оценке сложности по нотации «Большое О»?  Введите понятие доминирующей функции при оценке сложности с применением нотации «Большое О».  Выполните анализ «Большое О». для каждой из функций: n+5; n2+6n+7.  Выведите функцию оценки времени выполнения алгоритма линейного поиска в массиве из n элементов в худшем случае.  Выведите функцию оценки времени выполнения алгоритма бинарного поиска в массиве из n элементов.  Покажите что время вычисления 2n+n3 имеет порядок роста О(2n).  Приведите оценки времени по нотации выполнения циклов по нотации «Большое О» как функции от n:   1. for(s=0.0, i=0; i<n; i++) s+=a[i]\*b[i]; 2. for(i=0; i<n; i++) if (a[i]==k) return 1; return 0; 3. for(i=0; i<n; i++)   for(j=0; j<n; j++) b[i][j]\*=c; |
| **Владеть** лабораторные работы и защита лабораторных работ  Приведите оценки времени по нотации выполнения циклов по нотации «Большое О» как функции от n:   1. for(s=0.0, i=0; i<n; i++) s+=a[i]\*b[i]; 2. for(i=0; i<n; i++) if (a[i]==k) return 1; return 0; 3. for(i=0; i<n; i++)   for(j=0; j<n; j++) b[i][j]\*=c;  Как осуществляется выполнение операция вставки узла в AVL дерева?  Алгоритмы поворотов: левый, правый, двойной и их особенности.  Рекурсивные алгоритмы определения высоты дерева, количества листьев.  Привести алгоритм рекурсивного поиска узла в бинарном дереве поиска.  Привести алгоритм удаления узла с одним поддеревом.  Вывести префиксную, постфиксную и инфиксную формы выражений по дереву выражений.  Как осуществляется выполнение операция вставки узла в AVL дерева?  Алгоритмы поворотов: левый, правый, двойной и их особенности.  Структура данных – таблица (словарь).  Реализуйте операции над линейным списком : создание, вывод, поиск, используя шаблон list из STL.  Запишите на псевдокоде алгоритм вставки нового узла в вершину однонаправленного списка.  Запишите на псевдокоде алгоритм вставки нового узла в любую позицию однонаправленного списка.  Представьте структуру узла двунаправленного списка.  Запишите на псевдокоде алгоритм вставки нового узла в любую позицию двунаправленного списка. |

|  |
| --- |
| **ПК-1** (готовность применять основные методы и инструменты разработки программного обеспечения)  **Знать** методику реализации рекурсивных алгоритмов, метод «разделяй и властвуй», метод ветвей и границ, метод динамического программирования; знать теорию объектно-ориентированного программирования языка С++; знать технологию использования элементов библиотеки STL.  Какие процессы называют рекурсивными?  Какие функции называют рекурсивными?  Приведите основные правила реализации рекурсивных процессов посредством рекурсивных функций.  Какую рекурсию называют линейной?  Что означают понятия: глубина рекурсии; шаг рекурсии.  Реализуйте на языке Си алгоритм бинарного поиска в массиве.  Расскажите об организации алгоритмов полного перебора. Перечислите методы, обеспечивающие сокращение выполнения полного перебора.  Приведите пример метода перебор с возвратом для решения задачи: поиск выхода из лабиринта.  Суть метода динамического программирования.  Реализуйте на языке Си алгоритм бинарного поиска в массиве. |
| **Уметь** применять методику рекурсивных алгоритмов, методы «разделяй и властвуй», метод ветвей и границ, метод динамического программирования при разработке программного обеспечения  Реализуйте на абстрактном уровне алгоритмы обхода сильно ветвящегося дерева в глубину, используя АТД.  Алгоритмы обхода сильно ветвящегося дерева в ширину. |
| **Владеть** практическими навыками применения основных базовых типов данных языка С++, типов данных созданных пользователем, шаблонов STL при разработке программного обеспечения обработки больших наборов данных в среде Visual Studio (версия значения не имеет).  Привести алгоритм рекурсивного поиска узла в бинарном дереве поиска.  Привести алгоритм удаления узла с одним поддеревом  Применение алгоритмов обхода бинарного дерева к обходу дерева поиска.  Рекурсивные алгоритмы определения высоты дерева, количества листьев.  Привести алгоритм рекурсивного поиска узла в бинарном дереве поиска.  Как подключить STL в программу?  Какую структуру реализует контейнер vector STL?  Какой контейнер реализует хеш - таблицу?  Какой контейнер реализует линейный список?  Какие алгоритмы включены в STL ? |

**ФОС для промежуточного контроля**

**Приблизительный список экзаменационных вопросов (3 семестр)**

1. Три уровня представления данных. Понятие типа данных. Понятие Структура данных. Свойства структур данных.
2. Представление данных в памяти компьютера. Иерархия коллекций данных Динамические и статические Структура данных, внешние Структура данных.
3. Бинарные операции. Массив, строка, структура, объединение. Указатель. Операции над указателями. Указатель и массив. Косвенный указатель. Функции. Передача структур данных в качестве параметров.
4. Определение АТД. Реализация АТД. АТД и объектно-ориентированное программирование. Поэтапная разработка программы с использованием АТД.
5. Асимптотический анализ верхней и средней оценок сложности алгоритмов; сравнение наилучших, средних и наихудших оценок; O-, o-, ω- и θ-нотации;
6. Стандартные классы сложности; эмпирические измерения эффективности алгоритмов; накладные расходы алгоритмов по времени и памяти; рекуррентные соотношения и анализ рекурсивных алгоритмов
7. Математическое понятие линейного списка. АТД - линейный список.
8. Способы реализации линейных списков (массив, однонаправленные, двунаправленные связанные списки).
9. Анализ алгоритмов операций над списком для различных реализаций. АТД линейных списков – Очередь, Стек и их способы реализации.
10. Алгоритмы операций: поиск (линейный, бинарный), сортировка со сложностью О(n2) и О(n log(n)). Применение стека и очереди в алгоритмах организации вычисления выражения (Польская постфиксная и префиксные записи выражений).
11. Полный перебор. Рекурсивные алгоритмы и их реализация. Метод “разделяй и властвуй”.
12. “Жадные” алгоритм. Метод динамического программирования. Бэктрекинг (перебор с возвратами). Метод ветвей и границ. Эвристический поиск. Поиск по образцу, алгоритмы обработки строк; алгоритмы аппроксимации числовых функций.
13. Определение класса. Конструкторы класса. Деструкторы класса. Перегрузка операций. Шаблонные классы. Абстрактные классы.
14. Файл. Алгоритмы поиска в файле. Использование индексных файлов. В-дерево и поиск данных в файле. Алгоритмы внешней сортировки.
15. Свойства. Иерархические Структура. Основные понятия. АТД дерево. Способы реализации. Алгоритмы обхода: в глубину и ширину.
16. АТД бинарное дерево. Сбалансированное дерево. Способы реализации. Рекурсивные алгоритмы обхода бинарного дерева. Дерево выражений.
17. Алгоритмы поиска и иерархические Структура данных. Бинарное дерево поиска. Бинарное сбалансированное дерево. AVL дерево. Алгоритмы вставки и удаления. Красно-черные деревья. Анализ алгоритмов поиска с использованием бинарного дерева поиска и AVL дерева.
18. Алгоритмы поиска. Организация быстрого поиска с использованием хеш таблицы. Хэш функции. Коллизия. Способы разрешения коллизий: цепное хещирование; открытый адрес. Поиск с возвратом. Анализ алгоритма поиска с использованием хеш таблицы.
19. Направленные и ненаправленные графы. Сети. АТД-граф. Способы реализации графа в памяти (списки и матрицы смежности). Алгоритмы обхода графа в ширину и в глубину.. Алгоритмы поиска кратчайших путей в графе между двумя заданными вершинами (Дейкстры).
20. Транзитивное замыкание (алгоритм Флойда). Алгоритмы построения минимального покрывающего дерева (алгоритмы Прима и Крускала). Топологическая сортировка

**Примеры задач**

1. Двунаправленный список «Пациенты». Узел списка содержит данные о пациенте: номер полиса пациента и указатель на однонаправленный список заболеваний пациента. Информационная часть узла однонаправленного списка содержит: целое число – код заболевания, месяц и год регистрации заболевания. Разработать функцию, которая подсчитывает количество пациентов, у которых зарегистрировано одно и то же заболевание в один и тот же месяц одного и того же года.
2. В узлах двунаправленного списка хранятся положительные и отрицательные числа, причем справа от вершины располагаются только положительные, а слева только отрицательные числа. Разработать функцию, которая вставляет положительные числа справа от вершины, а отрицательные слева
3. Создание слова–путаницы. Длина слова не известна. Поступающий текст (слово) перепутывают согласно значению, выработанному датчиком rand(2): если датчик вернул 0, то символ вставляется в начало списка, если 1, то в конец. Реализуйте задачу на связанном однонаправленном динамическом списке с оптимизацией операций включения новых узлов в начало и в конец списка.
4. Даны два списка L и М. Где L {L0,L1,L2,…..Lki } где 0<=i<=ki; и 0<=Li<=99, а М={M0, M1,…..Mkj} где 0<=j<=kj и 100<=Mj<=199. Объедините эти списки в один список LM={L0M0, L1M1….., LiMj….} где Мj>Li, а Li и Mj это два значения одного элемента списка LM. Реализуйте операцию по созданию списка LM на основе однонаправленного списка, из списков однонаправленных списков L и М.

**Вопросы к зачету за 4 семестр**

Вопросы к зачету.

1. Анализ сложности алгоритма.
2. Оценка времени выполнения программы.
3. Оценка объема памяти для хранения данных.
4. Рекурсивные алгоритмы.
5. Методы разработки программ: разделяй и властвуй; жадный алгоритм; динамическое программирование.
6. Алгоритмы поиска и их эффективность.
7. Алгоритмы сортировки и их эффективность.
8. Бинарные деревья.
   1. Определение бинарного дерева. Полное дерево. Раздваивающееся дерево.
   2. Представление в памяти
   3. Алгоритмы обхода: рекурсивный, нерекурсивный (записать на псевдокоде)
   4. Алгоритмы обхода дерева в ширину, в глубину.
   5. Алгоритмы расчета: количество листьев в дереве, высота дерева, количество элементов.
9. Сбалансированное бинарное дерево. Алгоритм создания дерева с известным числом узлов.
10. Бинарное дерево поиска
    1. Определение бинарного дерева
    2. Представление в памяти
    3. Операции Вставки, удаления, поиска.
    4. Использование дерева поиска.
11. AVL дерево. Его назначение. Особенности операций по вставке, удалению и поиску элементов.
12. Хеширование. Хеш функция. Открытая адресация. Сцепленное хеширование. Коллизия. Двойное хеширование. Кластеризация и способы ее устранения.
13. Структуры данных во внешней памяти. Способы организации и методы доступа к файлу. Выполнение операций над записями в файле: на основе индексов: (разреженный, плотный); на основе хеширования. Вторичные индексы.
14. Граф и его представление в памяти. Алгоритмы обхода графа. Нахождение всех возможных путей от вершины к вершине в связном графе.