

Министерство науки и высшего образования Российской Федерации
федеральное государственное бюджетное образовательное учреждение
высшего образования
«Курганский государственный университет»
(КГУ)

Кафедра «Автоматизация производственных процессов»



УТВЕРЖДАЮ:
Первый проректор
/Щербич С.Н. /

«03» сентября 2019 г.

Рабочая программа учебной дисциплины

Структуры и алгоритмы обработки данных

образовательной программы высшего образования –
программы бакалавриата

27.03.04 Управление в технических системах

Направленность:

Системы и технические средства автоматизации и управления

Формы обучения: очная, заочная

Курган 2019

Рабочая программа дисциплины «Структуры и алгоритмы обработки данных» составлена в соответствии с учебными планами по программе бакалавриата Управление в технических системах (Системы и технические средства автоматизации и управления) утвержденными :

- для очной формы обучения «29» августа 2019 года;
- для заочной формы обучения «29» августа 2019 года.

Рабочая программа дисциплины одобрена на заседании кафедры «Автоматизация производственных процессов» «02» сентября 2019 года, протокол № 1.

Рабочую программу составил

Старший преподаватель

И.П. Камкин

Согласовано:

Заведующий кафедрой
«Автоматизация производственных
процессов»

Е.К. Карпов

Специалист по учебно-
методической работе
Учебно-методического отдела

Г.В. Казанкова

Начальник Управления
Образовательной деятельности

С.Н. Синицын

1. ОБЪЕМ ДИСЦИПЛИНЫ

Всего: 3 зачетных единицы трудоемкости (108 академических часов)

Очная форма обучения

Вид учебной работы	На всю дисциплину	Семестр
		3
Аудиторные занятия (контактная работа с преподавателем), всего часов	32	32
в том числе:		
Лекции	12	12
Лабораторные работы	20	20
Самостоятельная работа, всего часов	76	76
в том числе:		
Подготовка к зачету	18	18
Другие виды самостоятельной работы (самостоятельное изучение тем (разделов) дисциплины)	58	58
Вид промежуточной аттестации	Зачет	Зачет
Общая трудоемкость дисциплины и трудоемкость по семестрам, часов	108	108

Заочная форма обучения

Вид учебной работы	На всю дисциплину	Семестр
		6
Аудиторные занятия (контактная работа с преподавателем), всего часов	10	10
в том числе:		
Лекции	2	2
Лабораторные работы	6	6
Практические занятия	2	2
Самостоятельная работа, всего часов	98	98
в том числе:		
Подготовка контрольной работы	18	18
Подготовка к зачету	18	18
Другие виды самостоятельной работы (самостоятельное изучение тем (разделов) дисциплины)	62	62
Вид промежуточной аттестации	Зачет	Зачет
Общая трудоемкость дисциплины и трудоемкость по семестрам, часов	108	108

2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ

Дисциплина «Структуры и алгоритмы обработки данных» относится к блоку 1 обязательным дисциплинам вариативной части учебного плана подготовки бакалавров.

Изучение дисциплины базируется на результатах обучения, сформированных при изучении следующих дисциплин:

- Математика;
- Информатика;
- Программирование и алгоритмизация.

Результаты обучения по дисциплине необходимы для изучения общеинженерных и специальных технических дисциплин, в частности «Информационные сети и телекоммуникации», «Проектирование автоматизированных систем», «Автоматизированные информационно-управляющие системы» и других профилирующих дисциплин, а также в последующей инженерной деятельности; планировании и проведении научно-исследовательских и проектно-конструкторских работ.

Требования к входным знаниям, умениям, навыкам и компетенциям:

- понимание устной (монологической и диалогической) речи на бытовые и общекультурные темы;
- знание общих теоретических принципов программирования, методов и средств разработки систем автоматизации и управления;
- владение навыками поиска, хранения и обработки цифровой информации с использованием средств компьютерной техники;
- освоение следующих компетенций на уровне не ниже порогового: ОПК-6 (способность 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) и ОПК-9 (способность использовать навыки работы с компьютером, владеть методами информационных технологий, соблюдать основные требования информационной безопасности).

3. ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ ОБУЧЕНИЯ

Целью освоения дисциплины «Структуры и алгоритмы обработки данных» является: формирование знаний и умений в использовании современных средств программирования и интегрированных систем программирования.

Задачами дисциплины являются: сформировать базовые теоретические понятия, лежащие в основе процесса разработки алгоритмов и структур данных; научить реализации типовых алгоритмов и структур данных и их модификаций на выбранном рабочем языке программирования.

Компетенции, формируемые в результате освоения дисциплины:

- способность осуществлять сбор и анализ исходных данных для расчета и проектирования систем и средств автоматизации и управления (ПК-5);

- готовность производить инсталляцию и настройку системного, прикладного и инструментального программного обеспечения систем автоматизации и управления (ПК-17).

В результате изучения дисциплины обучающийся должен:

- Знать: основные типы данных, принципы построения производных структур данных в языках программирования (для ПК-5).

- Знать: основные методы и средства реализации алгоритмов обработки базовых данных и производных структур данных, основные алгоритмы обработки данных (для ПК-17).

- Знать: типовые объекты и структуры данных предоставляемые интегрированными средами программирования для решения производственных задач (для ПК-5).

- Уметь: использовать языки программирования и интегрированную среду для построения структур данных (для ПК-17).

- Уметь: реализовывать алгоритмы и структуры в виде программного кода и готовых программных решений (для ПК-5).

- Уметь: использовать конструкции языка и стандартные библиотеки для создания приложений (для ПК-17).

- Владеть: навыками разработки структур данных и классов (для ПК-5).

- Владеть: навыками использования стандартных алгоритмов обработки данных и структур (для ПК-17).

В рамках освоения дисциплины «Структуры и алгоритмы обработки данных» обучающиеся готовятся к решению следующих профессиональных задач:

- участие в расчетах и проектировании информационных систем средств автоматизации и управления в соответствии с техническим заданием;

- участие в разработке информационных систем по автоматизации производства, управлению бизнес-процессами;

В рамках освоения дисциплины «Структуры и алгоритмы обработки данных» обучающиеся готовятся к исполнению следующих трудовых функций профессионального стандарта:

- Управление проектами в области ИТ на основе полученных планов проектов;

- Выполнение работ по созданию (модификации) и сопровождению информационных систем, автоматизирующих задачи организационного управления и бизнес-процессы;

4. СОДЕРЖАНИЕ ДИСЦИПЛИНЫ

4.1. Учебно-тематический план

Очная форма обучения

Рубеж	Номер раздела, темы	Наименование раздела, темы	Количество часов контактной работы с преподавателем		
			Лекции	Практич. занятия	Лабораторные работы
Рубеж 1	1	Цели и задачи курса. Основные понятия и определения	1		
	2	Базовые типы данных языков программирования, преобразование и использование.	1		
	3	Принципы формирования производных типов данных и структур. Использование шаблонов типа данных	2		2
	4	Алгоритмы сортировки и поиска	2		8
		Рубежный контроль 1	-		0,5
Рубеж 2	5	Связные структуры данных. Списки.	2		4
	6	Связные структуры данных. Деревья.	2		4
	7	Библиотечные реализации структур данных. Другие структуры данных.	2		1
		Рубежный контроль 2			0,5
Всего:			12	-	20

Заочная форма обучения

Номер раздела, темы	Наименование раздела, темы	Количество часов контактной работы с преподавателем		
		Лекции	Практич. занятия	Лабораторные работы
1	Цели и задачи курса. Основные понятия и определения	0,1		
2	Базовые типы данных языков программирования, преобразование и использование.	0,1		
3	Принципы формирования производных типов данных и структур. Использование шаблонов типа данных	0,5		0,5
4	Алгоритмы сортировки и поиска	0,5	1	3
5	Связные структуры данных. Списки.	0,5	1	1
6	Связные структуры данных. Деревья.	0,2		1
7	Библиотечные реализации структур данных. Другие структуры данных.	0,1		0,5
Всего:		2	2	6

4.2. Содержание лекционных занятий

Тема 1. Цели и задачи курса. Основные понятия и определения.

Цели и задачи курса. Алгоритмы, типы алгоритмов. Основные парадигмы программирования. Понятие о типе данных.

Тема 2. Базовые типы данных языков программирования, преобразование и использование.

Свойства типа данных. Базовые типы данных языка программирования. Преобразования типов данных, кодирование, машинное представление.

Тема 3. Принципы формирования производных типов данных и структур. Использование шаблонов типа данных.

Принципы формирования производных типов данных. Формирование массивов, структур данных, комбинирование массивов и структур данных. Использование шаблонов типа данных при проектировании функций (процедур) в языке программирования.

Тема 4. Алгоритмы сортировки и поиска.

Понятие внутренней сортировки. «Классические» методы сортировки (пузырьковая, выбором, Шелла). Быстрая сортировка. Процедура разделения. Рекурсивный и не рекурсивный алгоритмы быстрой сортировки. Анализ сложности.

Понятие поиска. Процедуры сравнения данных. Обзор алгоритмов поиска данных. Последовательный поиск в сортированных и несортированных массивах.

Тема 5. Связные структуры данных. Списки.

Понятие. Основные определения. Линейный однонаправленный список. Реализация базового класса. Вставка элементов, удаление элементов, поиск элемента, удаление списка. Понятие об итераторах. Реализация класса внешнего итератора. Функции вставки, удаления и поиска элемента списка, с использованием внешнего итератора. Двухнаправленные списки.

Тема 6. Связные структуры данных. Деревья.

Деревья. Основные понятия и определения. Бинарные деревья. Упаковка бинарных деревьев в массив. Реализация алгоритма пирамидальной сортировки. Бинарный поиск в сортированных массивах.

Тема 7. Библиотечные реализации структур данных. Другие структуры данных.

Реализация списков и деревьев в стандартной библиотеке C++ и библиотеке .NET Core. Общее представление о графах, hash – таблицах, ассоциативных массивах, кучах.

4.3. Лабораторные занятия

Номер раздела, темы	Наименование раздела, темы	Наименование лабораторной работы	Норматив времени, час.	
			Очная форма обучения	Заочная форма обучения
3	Принципы формирования производных типов данных и структур. Использование шаблонов типа данных	Принципы формирования производных типов данных и структур. Использование шаблонов типа данных	2	0,5
4	Алгоритмы сортировки и поиска	Алгоритмы сортировки.	4	1,5
		Алгоритмы поиска	2	1
		Исследование времени работы различных алгоритмов	2	0,5
5	Связные структуры данных. Списки.	Реализация класса линейного однонаправленного списка.	2	0,5
		Реализация внешнего итератора для линейного однонаправленного списка	2	0,5
6	Связные структуры данных. Деревья.	Алгоритмы поиска и сортировки с использованием деревьев.	4	1
7	Библиотечные реализации структур данных. Другие структуры данных.	Реализация структур данных, функций сортировки и поиска в стандартной библиотеке C++.	2	0,5
Всего:			20	6

4.4. Практические занятия

Номер раздела, темы	Наименование раздела, темы	Наименование лабораторной работы	Норматив времени, час.
			Заочная форма обучения
4	Алгоритмы сортировки и поиска	Описание основных алгоритмов сортировки в графическом виде	1

5	Связные структуры данных. Списки.	Описание структуры линейного однонаправленного списка и алгоритмов работы с ним в графическом виде	1
Всего:			2

4.5. Контрольная работа (для заочной формы обучения).

Выполняется студентом на заданную тему. Примерный список тем указан в фонде оценочных средств раздел 6.4. Контрольная работа выполняется в виде реферата на заданную тему. Обязательно прикладывается исходный код программы, написанной студентом. Алгоритм работы этой программы описывается в виде блок-схемы.

5. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ОБУЧАЮЩИХСЯ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ

При прослушивании лекций рекомендуется в конспекте отмечать все важные моменты, на которых заостряет внимание преподаватель, в частности те, которые направлены на качественное выполнение соответствующих лабораторных работ.

Преподавателем запланировано использование при чтении лекций и проведении лабораторных занятий технологии учебной дискуссии. Поэтому рекомендуется фиксировать для себя интересные моменты с целью их активного обсуждения на дискуссии в конце лекции и на лабораторных занятиях.

Залогом качественного выполнения лабораторных и практических (для заочной формы обучения) работ является самостоятельная подготовка к ним путем повторения материалов лекций. Рекомендуется подготовить вопросы по неясным моментам и обсудить их с преподавателем в начале лабораторной работы.

Преподавателем запланировано применение на лабораторных занятиях технологий коллективного взаимодействия. Лабораторные работы выполняются с использованием программного обеспечения: Microsoft Visual Studio 2010, Microsoft Visual Studio Community 2019.

Преподавателем запланировано применение на практических занятиях технологий развивающейся кооперации, коллективного взаимодействия. Поэтому приветствуется групповой метод выполнения заданий и защиты отчетов.

Для текущего контроля успеваемости по очной форме обучения преподавателем используется балльно - рейтинговая система контроля и оценки академической активности. Поэтому настоятельно рекомендуется тщательно прорабатывать материал дисциплины при самостоятельной работе, участвовать во всех формах обсуждения и взаимодействия, как на лекциях, так и на лабораторных занятиях в целях лучшего освоения

материала и получения высокой оценки по результатам освоения дисциплины.

Выполнение самостоятельной работы подразумевает самостоятельное изучение разделов дисциплины, подготовку к лабораторным и практическим (для заочной формы обучения) работам, к рубежным контролям (для очной формы обучения), выполнение контрольной работы (для заочной формы обучения), подготовку к зачету.

Контрольная работа выполняется студентом самостоятельно, в соответствии с выбранной темой. Определяется полнота раскрытия темы и понимание обучающимся, изложенного материала. Оценка контрольной работы производится по результату собеседования.

Рекомендуемая трудоемкость самостоятельной работы представлена в таблице:

Рекомендуемый режим самостоятельной работы

Наименование вида самостоятельной работы	Рекомендуемая трудоемкость, акад. час.	
	Очная форма обучения	Заочная форма обучения
Самостоятельное изучение тем дисциплины:	34\	54-
Подготовка к лабораторным работам	20	6
Подготовка к практическим работам	-	2
Подготовка к рубежным контролям	4	0
Изучение преобразования различных типов данных языка программирования.	1	2
Изучение конструирования структур данных и их использование для решения практических задач.	4	5
Изучение методов сортировки и поиска данных и применение их для конкретных структур данных	8	14
Изучение способов оптимизации сортировки и поиска.	2	6
Изучение связанных структур данных	8	12
Изучение древовидных структур данных (деревьев) и методов сортировки и поиска на таких структурах	8	12
Изучение библиотечных функций для реализации структур данных и алгоритмов стандартных операций	3	3
Выполнение контрольной работы	-	18
Подготовка к зачету	18	18
Всего:	76	98

6. ФОНД ОЦЕНОЧНЫХ СРЕДСТВ ДЛЯ АТТЕСТАЦИИ ПО ДИСЦИПЛИНЕ

6.1. Перечень оценочных средств

1. Балльно-рейтинговая система контроля и оценки академической активности студентов в КГУ (для очной формы обучения).
2. Отчеты студентов по лабораторным работам.

3. Банк заданий к рубежным контролям №1, №2 (для очной формы обучения).

4. Список вопросов к зачету.

5. Контрольная работа (для заочной формы обучения).

6. Задания к практическим занятиям (для заочной формы обучения).

**6.2. Система балльно-рейтинговой оценки
работы студентов по дисциплине
Очная форма обучения**

№	Наименование	Содержание					
1	Распределение баллов за семестры по видам учебной работы, сроки сдачи учебной работы (доводятся до сведения студентов на первом учебном занятии)	Распределение баллов					
		Вид учебной работы:	Посещение лекций	Выполнение и защита отчетов по лабораторным работам	Рубежный контроль №1	Рубежный контроль №2	Зачет
		Балльная оценка:	До 6	До 40	До 12	До 12	До 30
	Примечания:	по 1 баллу за 2 часовую лекцию	До 4 баллов за 2-х часовую лабораторную, до 8-ми за 4-х часовую				
2	Критерий пересчета баллов в традиционную оценку по итогам работы в семестре и зачета	60 и менее баллов – незачтено; 61 и более баллов - зачтено					
3	Критерии допуска к промежуточной аттестации, возможности получения автоматического зачета (экзаменационной оценки) по дисциплине, возможность получения бонусных баллов	<p>Для допуска к промежуточной аттестации (зачету) студент должен набрать по итогам текущего и рубежного контроля не менее 50 баллов и должен выполнить все лабораторные работы и контрольную работу (для заочной формы обучения).</p> <p>Для получения зачета «автоматически» студенту необходимо набрать следующее минимальное количество баллов: - 61.</p> <p>Бонусные баллы могут быть начислены за активную работу на лабораторных занятиях (до 10 баллов за семестр).</p>					

4	<p>Формы и виды учебной работы для неуспевающих (восстановившихся на курсе обучения) студентов для получения недостающих баллов в конце семестра</p>	<p>В случае если к промежуточной аттестации (зачету) набрана сумма менее 50 баллов, студенту необходимо набрать недостающее количество баллов за счет выполнения дополнительных заданий, до конца последней (зачетной) недели семестра. При этом необходимо проработать материал всех пропущенных лабораторных работ.</p> <p>Формы дополнительных заданий (назначаются преподавателем):</p> <ul style="list-style-type: none"> - выполнение и защита пропущенной лабораторной работы (при невозможности дополнительного проведения лабораторной работы преподаватель устанавливает форму дополнительного задания по тематике пропущенной лабораторной работы самостоятельно) – до 8 баллов. <p>Ликвидация академических задолженностей, возникших из-за разности в учебных планах при переводе или восстановлении, проводится путем выполнения дополнительных заданий, форма и объем которых определяется преподавателем.</p>
---	--	--

6.3. Процедура оценивания результатов освоения дисциплины

Рубежные контроли и зачет проводятся в форме тестирования или устного опроса. Тестирование проводится письменно или с применением информационных систем.

Перед проведением каждого рубежного контроля преподаватель прорабатывает со студентами основную материал соответствующих разделов дисциплины в форме краткой лекции-дискуссии. Варианты задания для рубежного контроля №1 и 2 состоят из задачи.

На каждый рубежный контроль студенту отводится время не менее 30 минут. По результатам рубежного контроля студент получает до 12 баллов.

Возможно использование теста для неуспевающих студентов.

Преподаватель оценивает в баллах результаты тестирования каждого студента по количеству правильных ответов и заносит в ведомость учета текущей успеваемости.

Зачет проводится в традиционной форме. Билет к зачету состоит из 2 вопросов. Количество баллов за каждый правильный ответ до 15 баллов. Время, отводимое студенту на зачет, составляет 1 астрономический час. Результаты текущего контроля успеваемости и зачета заносятся преподавателем в ведомость, которая сдается в организационный отдел института в день зачета, а также выставляются в зачетную книжку студента.

6.4. Примеры оценочных средств для рубежных контролей и зачета

Пример задания для рубежного контроля 1

Написать программу выполняющую следующие действия:

Сортировку информации, хранящейся в файле в виде записей заданной структуры. Сортировку производить по реквизитам различных типов, использовать один из известных алгоритмов сортировок, например, алгоритм Шелла, Пузырька и т.д...

Пример задания для рубежного контроля 2

Написать программу выполняющую следующие действия:

Написать программу реализующую работу со списком студенческой группы. Необходимые действия: вывод на экран, сохранение в файл, добавление и удаление студента из группы. Информацию о студенте хранится в отдельном классе.

Тест для неуспевающих студентов:

1. Алгоритм - это ...

линейная последовательность действий исполнителя записанная с использованием его системы команд, и воспринимаемая посредством устройства чтения информации

конечная совокупность точно заданных правил решения некоторого класса задач или понятное и точное предписание исполнителю совершить последовательность действий, направленных на достижение указанной цели или на решение поставленной задачи

путь решения какой-либо задачи записанный в виде блок-схемы

набор правил, устанавливающих, какие комбинации символов являются осмысленными предложениями на этом языке.

2. Какое это свойство алгоритма « Алгоритм должен содержать предписания о выполнении только таких действий, которые входят в систему команд исполнителя»

Понятность

Дискретность

Определенность

Результативность

Массовость

3. Выберите пункты, относящиеся к отличительным особенностям понятия «тип данных»:

тип определяет класс значений, которые могут принимать переменная или выражение.

каждое значение принадлежит одному и только одному типу

тип значения константы, переменной или выражения можно вывести либо из контекста, либо из самого операнда, не обращаясь к значениям, вычисляемым во время работы программы

в общем случае, один тип данных может объединять объекты определенные в разных файлах и имеющие разную структуру

типы данных увеличивают неопределенность в выполнении программ, но применяется для упрощения разработки

4. Укажите в какой строке записаны константы типа double в C++

*"Привет, мир!", "двенадцать", "88.5"
'x', 't', '4'
5, 8, 10
8.0f, 44.4f, 3.14f
9.0, 2.7, 53.4*

5. Описание какой парадигмы программирования дано: «Методология этой парадигмы основана на принципе конструирования программ как описаний последовательного изменения состояния вычислителя пошаговым образом. Для этой парадигмы характерна полная определенность и контролируемость переходов из одного состояния в другое

*Объектно-ориентированное
Императивное
Процедурное
Функциональное
Нейро-сетевое*

6. Укажите верное обращение к элементам объекта следующей структуры

```
struct Color  
{  
    char R;  
    char G;  
    char B;  
};  
Color Red;
```

*Red::R=255;
Red->R=255;
Red[R] = 255;
Red.R = 255;*

7. Чем структуры в C++ отличаются от классов?

Классы могут содержать описания функций, а структуры не могут

Структуры предоставляют возможность создавать собственные типы данных, а классы нет.

Отличаются модификаторами доступа, члены классов по умолчанию считаются private, а структур public.

Для структур невозможно использовать модификаторы доступа private, public, protected, а для классов возможно.

8. Укажите где верно описан массив классов

```
class Car  
{
```

```

double tnk_vol,speed;
Engine eng;
Car Car(double t_v,Engine e)
{
    tnk_vol = t_v; eng=e;
}
};

```

```

Car Cars(15);
Car Cars[15] = new Car();
Car Cars[15];
Cars = new Car(15);

```

9. Для сравнения сложности алгоритмов используют класс (набор) функций $O(f)$. Выберите верное определение этого класса

Функции описывающие количество элементов массива, которые могут быть обработаны данным алгоритмом за 1 секунду. Для каждого типа данных своя функция.

Функции определяющие конечное число шагов, которые необходимо сделать для обработки алгоритмом 1 элемента

Функции, растущие не быстрее f . f - является верхней границей для класса.

Функции, растущие не медленнее f . f - является нижней границей для класса.

10. Выберите, где классы функций роста сложности алгоритмов расположены от самой медленной до самой быстрой

$\log_2(n)$, n , $n \cdot \log_2(n)$, n^2 , n^3 , 2^n
 2^n , n^3 , n^2 , $n \cdot \log_2(n)$, n , $\log_2(n)$
 $n \cdot \log_2(n)$, 2^n , n^3 , n^2 , n , $\log_2(n)$
 $\log_2(n)$, $n \cdot \log_2(n)$, n^3 , n^2 , n , 2^n

11. Код функции какой сортировки приведен:

```

void SortInt1( int *array, int n )
{
    for( int i = 1; i < n; i++ )
        for( int j = n-1; j >= i; j-- )
            if( array[ j ] < array[ j-1 ] )
            {
                int copy = array[ j ];
                array[ j ] = array[ j-1 ];
                array[ j-1 ] = copy;
            }
}

```

Быстрая

Простыми подстановками

Простыми обмeнами (пузырьковая)

Пирамидальная

Шелла

12. Укажите, где приведен корректный прототип шаблонной функции в C++?

*template <class T> void BetterBubbleSortPlus(T *a, int n);*

*void BetterBubbleSortPlus(template T *a, int n);*

*<class T> void BetterBubbleSortPlus(int *a, int n);*

*void template <class T> BetterBubbleSortPlus(T *a, int n);*

13. Описание элемента какой структуры данных приведено?

```
template <class ITEM>
struct TItem
{
    ITEM data;
    TItem<ITEM> *next;

    TItem( const ITEM& data, TItem<ITEM>* next = 0 )
    {
        this->data = data;
        this->next = next;
    }
};
```

массив
линейный двунаправленный список
линейный однонаправленный список
бинарное дерево
граф

14. Узел дерева не имеющий родителя называется:

листом
сиротой
корневым узлом
главным узлом

15. Для чего предназначена приведенная ниже функция

```
void Func( int size, const char* filename )
{
    fInput = fopen( filename, "rt" );
    if( fInput == NULL )
    {
        printf( "Error while opening key file\n" );
        exit( 1 );
    }
}
```



```

InitSearchTree( size );
PrepareTree( 1 );
if( fclose( fInput ) == EOF )
{
    printf( "Error while closing key file \n" );
    exit( 2 );
}
}

```

*Подготовка таблицы поиска на основе информации, загружаемой из файла
Создание дерева поиска заданного размера
Первоначальной подготовки массива ключей для бинарного поиска
Чтения текстового файла и вывод его содержимого на экран*

Примерный список вопросов к зачету:

1. Алгоритмы, типы алгоритмов. Основные парадигмы программирования.
2. Понятие типа данных. Свойства типа данных. Базовые типы данных в языке программирования C++.
3. Структуры и классы. Их конструирование. Способы доступа к элементам структуры (класса).
4. Конструирование своих типов данных на основе структур и массивов, конструирование массива структур и структур, содержащих массивы.
5. Преобразование типов данных.
6. Сортировка. Понятие внутренней сортировки.
7. Назовите и кратко опишите известные Вам алгоритмы сортировки.
8. Примеры использования алгоритма сортировки для произвольной структуры данных.
9. Назначение алгоритмов поиска данных. Применение алгоритмов поиска, условия применения.
10. Приведите примеры кода алгоритмов поиска данных.
11. Связные структуры данных. Понятие. Основные определения.
12. Линейный однонаправленный список. Реализация базового класса (без кода функций).
13. Линейный однонаправленный список. Базовые действия. Вставка элементов, удаление элементов, поиск элемента, удаление списка.
14. Линейный однонаправленный список. Понятие об итераторах. Реализация класса внешнего итератора. Базовые действия.

15. Линейный однонаправленный список. Функции вставки, удаления и поиска элемента списка, с использованием внешнего итератора.
16. Деревья. Основные понятия и определения.
17. Деревья. Упаковка деревьев в массив. Реализация алгоритма пирамидальной сортировки.
18. Напишите код программы «Сортировка пузырьком» для заданной структуры данных хранящейся в файле.
19. Напишите код программы «Сортировка выбором» для заданной структуры данных хранящейся в файле.
20. Напишите код программы «Сортировка Шелла» для заданной структуры данных хранящейся в файле.
21. Напишите код программы «Пирамидальная сортировка» для заданной структуры данных хранящейся в файле.
22. Напишите код программы «Быстрая сортировка» для заданной структуры данных хранящейся в файле.

Примерная тематика контрольных работ.

1. Формирование сложных структур данных в языке программирования.
2. Преобразование различных типов данных в языке программирования.
3. Формирование классов и объектов в современном языке программирования.
4. Сравнительные характеристики различных алгоритмов сортировки.
5. Практическое использование «Сортировка пузырьком».
6. Практическое использование «Сортировка выбором».
7. Практическое использование «Сортировка Шелла».
8. Практическое использование «Пирамидальная сортировка».
9. Практическое использование «Быстрая сортировка».
10. Нахождение минимальных и максимальных значений.
11. Поиск информации в произвольном текстовом файле.
12. Поиск информации в упорядоченном массиве.
13. Поиск в структуре по нескольким реквизитам.
14. Поиск в HTML документе.

6.5. Фонд оценочных средств

Полный банк заданий для текущего, рубежных контролей и промежуточной аттестации по дисциплине, показатели, критерии, шкалы оценивания компетенций, методические материалы, определяющие процедуры оценивания образовательных результатов, приведены в учебно-методическом комплексе дисциплины.

7. ОСНОВНАЯ И ДОПОЛНИТЕЛЬНАЯ УЧЕБНАЯ ЛИТЕРАТУРА

7.1. Основная учебная литература

1. Колдаев, В. Д. Структуры и алгоритмы обработки данных : учебное пособие / В. Д. Колдаев. - Москва : РИОР : ИНФРА-М, 2020. - 296 с. - (Высшее образование: Бакалавриат). - ISBN 978-5-369-01264-2. - Текст : электронный. - URL: <https://znanium.com/catalog/product/1054007>
2. Дроздов, С. Н. Структуры и алгоритмы обработки данных: Учебное пособие / Дроздов С.Н. - Таганрог:Южный федеральный университет, 2016. - 228 с.: ISBN 978-5-9275-2242-2. - Текст : электронный. - URL: <https://znanium.com/catalog/product/991928>

7.2. Дополнительная учебная литература

1. Пышкин Е.В. Структуры данных и алгоритмы: реализация на С/С++. - СПб.: ФТК СПбГПУ, 2009.- 200 с., ил.
2. Кубенский А.А. Структуры и алгоритмы обработки данных: объектно-ориентированный подход на С++. - СПб.: Питер,2004. - 464 с.
3. Камаев В.А. Технологии программирования: Учебник/В.А. Камаев, В.В. Костерин. — 2-е изд., перераб. и доп. — М.: Высш. шк.,2006. - 454 с: ил.
4. Пол А. Объектно-ориентированное программирование на С++. М.: Финансы и статистика, 2005. -477 с.
5. Буч Г. Объектно-ориентированное проектирование с примерами применения, М.: Финансы и статистика, 2002 515 с.
6. Дж. Элджер, С++: библиотека программиста, СПб: Питер, 2000. – 320 с.

8. УЧЕБНО-МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ САМОСТОЯТЕЛЬНОЙ РАБОТЫ ОБУЧАЮЩИХСЯ

1. Камкин И. П. Методические указания к контрольной работе по дисциплине «Структуры и алгоритмы обработки данных» для студентов заочной формы обучения - Курган, 2019.
2. Камкин И. П. Методические указания к комплексу практических работ по дисциплине «Структуры и алгоритмы обработки данных» для студентов заочной формы обучения - Курган, 2019.
3. Камкин И. П. Методические указания к комплексу лабораторных работ по дисциплине «Структуры и алгоритмы обработки данных» для студентов очной и заочной форм обучения – Курган, 2019.

9. РЕСУРСЫ СЕТИ «ИНТЕРНЕТ», НЕОБХОДИМЫЕ ДЛЯ ОСВОЕНИЯ ДИСЦИПЛИНЫ

1. dist.kgsu.ru - Система поддержки учебного процесса КГУ;

2. <http://www.cetiforum.ru> - Электронные версии учебников, самоучителей по программированию на алгоритмических языках

3. <https://docs.microsoft.com/ru-ru/> - Документация по Microsoft Visual Studio и многое другое.

4. <https://code-live.ru/tag/cpp-manual/> - Простые задачи на C++ для начинающих

5.

https://www.youtube.com/watch?v=SW_UCzFO7X0&list=PLawfWYMUziZqyUL5QDLVbe3j5BKWj42E5&index=2 – Курс CS50 по программированию от Гарварда. Видео лекций на русском языке.

10. ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ, ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ И ИНФОРМАЦИОННЫЕ СПРАВОЧНЫЕ СИСТЕМЫ

При чтении лекций используются слайдовые презентации.

Для лабораторных работ необходим программный продукт Microsoft Visual Studio 2010 и Microsoft Visual Studio 2019. Минимальные требования к операционной системе и программному обеспечению компьютера, используемого при чтении лекций: Windows XP, Foxit Reader, Microsoft Visual Studio 2010.

11. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

Компьютерный класс, мультимедийное оборудование (переносной персональный компьютер, мультимедийный проектор, мультимедийный экран).

Аннотация к рабочей программе дисциплины
«СТРУКТУРЫ И АЛГОРИТМЫ ОБРАБОТКИ ДАННЫХ»

образовательной программы высшего образования –
программы бакалавриата
27.03.04 – Управление в технических системах

Направленность:
Системы и технические средства автоматизации и управления

Трудоемкость дисциплины: 3 ЗЕ (108 академических часов)
Семестр: 5 (очная форма обучения), 6 (заочная форма обучения)
Форма промежуточной аттестации: Зачет

Содержание дисциплины

Алгоритмы, типы алгоритмов. Основные парадигмы программирования. Понятие о типе данных. Свойства типа данных. Базовые типы данных языка программирования. Преобразования типов данных, кодирование, машинное представление. Принципы формирования производных типов данных. Формирование массивов, структур данных, комбинирование массивов и структур данных. Использование шаблонов типа данных при проектировании функций (процедур) в языке программирования. Понятие внутренней сортировки. «Классические» методы сортировки (пузырьковая, выбором, Шелла). Быстрая сортировка. Понятие поиска. Процедуры сравнения данных. Обзор алгоритмов поиска данных. Последовательный поиск в сортированных и несортированных массивах. Линейный однонаправленный список. Понятие об итераторах. Двухнаправленные списки. Деревья. Бинарные деревья. Упаковка бинарных деревьев в массив. Реализация алгоритма пирамидальной сортировки. Бинарный поиск в сортированных массивах. Реализация списков и деревьев в стандартной библиотеке C++ и библиотеке .NET Core.