Подписано электронной подписью:
Вержицкий Данил Григорьевич
Должность: Директор КГПИ ФГБОУ ВО «КемГУ»
Дата и время: 2024-02-21 00:00:00
471086fad29a3b30e244c728abc3661ab35e9d50210dcf0e75e03a5b6fdf6436

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Кемеровский государственный университет»
Новокузнецкий институт (филиал)
федерального государственного бюджетного образовательного учреждения
высшего образования
«Кемеровский государственный университет»

УТВЕРЖДАЮ:
Декан факультета информатики,
математики и экономики

Фомина А.В. _____
« 14» февраля 2020 г.

Рабочая программа дисциплины
Б1.О. 04 Структуры и алгоритмы компьютерной обработки данных

Направление подготовки
*«***09.03.03 Прикладная информатика***»*

Направленность (профиль) подготовки
«Прикладная информатика в образовании»

Программа подготовки
бакалавриата

Квалификация выпускника
бакалавр

Форма обучения
заочная

Год набора 2019

Новокузнецк 2020

Лист внесения изменений

в РПД Б1.О.01.02 История (история России, всеобщая история)

Сведения об утверждении:

утверждена Ученым советом факультета информатики, математики и экономики
(протокол Ученого совета факультета № 9 от 14.02.2019 г.)

для ОПОП 2019 год набора на 2019 / 2020 учебный год
по направлению подготовки *09.03.03 Прикладная информатика*
направленность (профиль) *Прикладная информатика в образовании*
Одобрена на заседании методической комиссии факультета ФИМЭ
протокол методической комиссии факультета № 6 от 14.02.2019)
Одобрена на заседании обеспечивающей кафедры ИОТД
протокол № 5 от 19.01.2019 г. Можаров М.С. / _____
(Ф. И.О. зав. кафедрой) *(Подпись)*

Переутверждение на учебный год:

на 2020 / 2021 учебный год
утверждена Ученым советом факультета информатики, математики и экономики
(протокол Ученого совета факультета № 8 от 13.02.2020 г.
Одобрена на заседании методической комиссии факультета ФИМЭ
протокол методической комиссии факультета № 6 от 06.02.2020 г.
Одобрена на заседании обеспечивающей кафедры ИОТД
протокол № 5 от19.12.2019г. Можаров М.С. / _____
(Ф. И.О. зав. кафедрой) *(Подпись)*

на 20____ / 20____ учебный год

утверждена Ученым советом факультета_____
(протокол Ученого совета факультета № __ от __.__.201_ г.
Одобрена на заседании методической комиссии факультета _____
протокол методической комиссии факультета № __ от __.__.20__ г.
Одобрена на заседании обеспечивающей кафедры _____
протокол № __ от ___.___.20____г. _____ / _____
(Ф. И.О. зав. кафедрой) *(Подпись)*

на 20____ / 20____ учебный год

утверждена Ученым советом факультета_____
(протокол Ученого совета факультета № __ от __.__.201_ г.
Одобрена на заседании методической комиссии факультета _____
протокол методической комиссии факультета № __ от __.__.20__ г.
Одобрена на заседании обеспечивающей кафедры _____
протокол № __ от ___.___.20____г. _____ / _____
(Ф. И.О. зав. кафедрой) *(Подпись)*

Оглавление

1 Цель дисциплины 4
1.1 Формируемые компетенции 4
1.2 Индикаторы достижения компетенций 4
1.3 Знания, умения, навыки (ЗУВ) по дисциплине 4
2 Объём и трудоёмкость дисциплины по видам учебных занятий. Формы промежуточной аттестации. 5
3. Учебно-тематический план и содержание дисциплины 5
3.1 Учебно-тематический план 5
3.2. Содержание занятий по видам учебной работы 6
4 Порядок оценивания успеваемости и сформированности компетенций обучающегося в текущей и промежуточной аттестации. 8
5 Учебно-методическое обеспечение дисциплины. 9
5.1 Учебная литература 9
5.2 Материально-техническое и программное обеспечение дисциплины. 9
5.3 Современные профессиональные базы данных и информационные справочные системы 10
6 Иные сведения и (или) материалы. 10
6.1.Примерные темы письменных учебных работ 10
6.2. Примерные вопросы и задания / задачи для промежуточной аттестации 15

1 Цель дисциплины

В результате освоения данной дисциплины у обучающегося должны быть сформированы компетенции основной профессиональной образовательной программы бакалавриата (далее - ОПОП):

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

Содержание компетенций как планируемых результатов обучения по дисциплине см. таблицы 1 и 2.

1.1 Формируемые компетенции

Таблица 1 - Формируемые дисциплиной компетенции

Наименование вида компетенции	Наименование категории (группы) компетенций	Код и название компетенции
общепрофессиональная		ОПК-7 Способен разрабатывать алгоритмы и программы, пригодные для практического применения

1.2 Индикаторы достижения компетенций

Таблица 2 – Индикаторы достижения компетенций, формируемые дисциплиной

Код и название компетенции	Индикаторы достижения компетенции по ОПОП	Дисциплины и практики, формирующие компетенцию ОПОП
ОПК-7 Способен разрабатывать алгоритмы и программы, пригодные для практического применения	ОПК 7.1 Осуществляет и обосновывает выбор стандартных алгоритмов и программных средств для реализации практических задач ОПК 7.2 Разрабатывает алгоритмы и программы для реализации прикладных практических задач	Б1.О.04 Структуры и алгоритмы компьютерной обработки данных Б1.О.07 Программирование Б2.О.01(У) Ознакомительная Б2.О.02(П) Эксплуатационная Б2.О.04(Пд) Преддипломная

1.3 Знания, умения, навыки (ЗУВ) по дисциплине

Таблица 3 – Знания, умения, навыки, формируемые дисциплиной

Код и название компетенции	Индикаторы достижения компетенции, закрепленные за дисциплиной	Знания, умения, навыки (ЗУВ), формируемые дисциплиной
ОПК-7 Способен разрабатывать алгоритмы и программы, пригодные для практического применения	ОПК 7.1 Осуществляет и обосновывает выбор стандартных алгоритмов и программных средств для реализации практических задач ОПК 7.2 Разрабатывает алгоритмы и программы для реализации прикладных практических задач	Знать: - классификацию программных средств, языков программирования; - основные парадигмы программирования; - понятия и методы алгоритмизации; - основы и методы структурного программирования; - основные понятия объектно-ориентированного программирования; - основы теории алгоритмов и основы теории сложности; Уметь:

Код и название компетенции	Индикаторы достижения компетенции, закрепленные за дисциплиной	Знания, умения, навыки (ЗУВ), формируемые дисциплиной
		- разрабатывать алгоритмы для решения прикладных практических задач; - разрабатывать программы для реализации прикладных практических задач; - обосновывать выбор стандартных алгоритмов для решения практических задач; - осуществлять выбор стандартных средств для программной реализации алгоритмов и программ; Владеть: - методами алгоритмизации, оценки сложности алгоритмов; - графическим способом описания алгоритмов; - методами структурного программирования; - навыками реализации алгоритмов и программ, с учетом их сложности

2 Объём и трудоёмкость дисциплины по видам учебных занятий. Формы промежуточной аттестации.

Таблица 4 – Объем и трудоемкость дисциплины по видам учебных занятий

Общая трудоемкость и виды учебной работы по дисциплине, проводимые в разных формах	Объём часов по формам обучения		
	ОФО	ЗФО	ОЗФО
1 Общая трудоемкость дисциплины		144	
2 Контактная работа обучающихся с преподавателем (по видам учебных занятий) (всего)		12	
Аудиторная работа (всего):		12	
в том числе:			
лекции		6	
практические занятия, семинары			
практикумы			
лабораторные работы		6	
в интерактивной форме			
в электронной форме			
Внеаудиторная работа (всего):			
в том числе, индивидуальная работа обучающихся с преподавателем			
подготовка курсовой работы /контактная работа			
групповая, индивидуальная консультация и иные виды учебной деятельности, предусматривающие групповую или индивидуальную работу обучающихся с преподавателем)			
творческая работа (эссе)			
3 Самостоятельная работа обучающихся (всего)		123	
4 Промежуточная аттестация обучающегося: экзамен		9	

3. Учебно-тематический план и содержание дисциплины

3.1 Учебно-тематический план

Таблица 5 - Учебно-тематический план очной / заочной формы обучения

№ недели п/п	Разделы и темы дисциплины по занятиям	Общая трудоёмкость *(всего час.)*	Трудоемкость занятий (час.)						Формы[1] текущего контроля и промежуточной аттестации успеваемости
			ОФО			ЗФО			
			Аудиторн. занятия		СРС	Аудиторн. занятия		СРС	
			лекц.	практ.		лекц.	практ.		
Семестр 1									
1. Структуры компьютерной обработки данных									
1.1	Структурная организация данных.	**22**				1	1	20	ПР-6 – отчет по лабораторным работам ПР-4 реферат
1.2	Линейные структуры данных	**22**				1	1	20	ПР-6 – отчет по лабораторным работам ПР-2 - контрольная работа
1.3	Нелинейные структуры данных	**22**				1	1	20	ПР-6 – отчет по лабораторным работам ИЗ –индивидуальное задание
2. Алгоритмы компьютерной обработки данных									
2.1	Алгоритмы сортировки	**22**				1	1	20	ПР-6 – отчет по лабораторным работам ИЗ –индивидуальное задание
2.2	Алгоритмы поиска	**22**				1	1	20	ПР-6 – отчет по лабораторным работам ИЗ –индивидуальное задание
2.3	Алгоритмы на графах	**22**				1	1	20	ПР-6 – отчет по лабораторным работам ИЗ –индивидуальное задание
	Промежуточная аттестация - *экзамен*	**9**							УО-4 – экзамен
ИТОГО по семестру 1									
	Всего:	**144**				**6**	**6**	**123**	

3.2. Содержание занятий по видам учебной работы

Таблица 6 – Содержание дисциплины

№ п/п	Наименование раздела, темы дисциплины	Содержание занятия
Семестр 1		
Содержание лекционного курса		
1. Структуры компьютерной обработки данных		
1.1	Структурная организация данных.	Основные понятия структур данных. Классификация структур данных по признаку изменчивости. Линейные и нелинейные структуры данных. Типы и структуры данных в современных

[1] УО - устный опрос, УО-1 - собеседование, УО-2 - коллоквиум, УО-3 - зачет, УО-4 – экзамен, ПР - письменная работа, ПР-1 - тест, ПР-2 - контрольная работа, ПР-3 эссе, ПР-4 - реферат, ПР-5 - курсовая работа, ПР-6 - научно-учебный отчет по практике, ПР-7 - отчет по НИРС, ИЗ –индивидуальное задание; ТС - контроль с применением технических средств, ТС-1 - компьютерное тестирование, ТС-2 - учебные задачи, ТС-3 - комплексные ситуационные задачи

№ п/п	Наименование раздела, темы дисциплины	Содержание занятия
		языках программирования высокого уровня.
1.2	Линейные структуры данных	Линейные структуры данных: списки, стеки, очереди, деки. Линейный список. Операции над линейным списком. Циклический список. Операции над циклическим списком. Односвязная реализация циклического списка. Реализация основных операций над односвязным циклическим списком. Реализация линейного списка в виде двусвязной динамической структуры. Реализация основных операций над двусвязным списком. Циклический двусвязный список. Реализация основных операций над двусвязным циклическим списком. Стек. Операции над стеком. Использование стека для преобразования форм записи выражений. Очередь. Операции над очередью. Реализация очереди. Дек. Операции над деком. Реализация дека.
1.3	Нелинейные структуры данных	Нелинейные структуры данных: деревья, графы. Бинарные деревья. Основные понятия и определения. Построение бинарного дерева. Операции над бинарным деревом. Дерево выражения. Дерево поиска. Операции над деревом поиска. Сбалансированные деревья. Включение в сбалансированное дерево. Дерево Фибоначчи. Графы. Основные понятия и определения. Операции над графом. Обход ориентированного графа. Вычисление расстояния между узлами ориентированного графа.
2. Алгоритмы компьютерной обработки данных		
2.1	Алгоритмы сортировки	Сортировка выбором. Сортировка вставкой. Сортировка слиянием. Сортировка обменом. Шейкерная сортировка. Сортировка Шелла. Быстрая сортировка (сортировка Хоара). Турнирная сортировка. Пирамидальная сортировка. Сравнение различных алгоритмов сортировки по временной и объемной сложности.
2.2	Алгоритмы поиска	Последовательный поиск. Бинарный поиск. Фибоначчиев поиск. Интерполяционный поиск. Поиск по бинарному дереву. Поиск по бору. Поиск хешированием. Оценка сложности алгоритмов поиска.
2.3	Алгоритмы на графах	Структуры для представления графа. Обходы графа в глубину и ширину. Поиск путей в графе. Алгоритм Дейкстры. Алгоритм Флойда. Алгоритм построения минимального остовного дерева.
Содержание лабораторных занятий		
1. Структуры компьютерной обработки данных		
1.1	Структурная организация данных.	Понятие типа данных. Типы данных в языке С++. Простые типы данных. Числовые типы данных. Логический тип данных. Символьный тип. Перечисляемые типы данных. Структурные типы данных. Способы представления структурных данных. Структуры. Массивы. Строки.
1.2	Линейные структуры данных	Реализация линейного списка в виде односвязной динамической структуры. Реализация основных операций над односвязным списком. Реализация стека. Реализация основных операций над стеком. Реализация очереди и дека.
1.3	Нелинейные структуры данных	Реализация бинарного дерева. Реализация основных операций над бинарным деревом. Реализация дерева поиска. Реализация операций над деревом поиска. Реализация графа. Реализация основных операций над ориентированным графом. Реализация основных операций над очередью и деком.

№ п/п	Наименование раздела, темы дисциплины	Содержание занятия
2. Алгоритмы компьютерной обработки данных		
2.1	Алгоритмы сортировки	Реализация алгоритмов сортировки. Сравнение различных алгоритмов сортировки по временной и объемной сложности.
2.1	Алгоритмы поиска	Реализация алгоритмов поиска на линейных структурах данных. Оценка сложности различных алгоритмов поиска.
2.2	Алгоритмы на графах	Реализация обхода графа в глубину. Реализация обхода графа в ширину. Построение минимального остовного дерева. Поиск кратчайшего пути из одной вершины. Поиск кратчайших путей между всеми парами вершин. Нахождение максимального потока.

4 Порядок оценивания успеваемости и сформированности компетенций обучающегося в текущей и промежуточной аттестации.

Для положительной оценки по результатам освоения дисциплины обучающемуся необходимо выполнить все установленные виды учебной работы. Оценка результатов работы обучающегося в баллах (по видам) приведена в таблице 7.

Таблица 7 - Балльно-рейтинговая оценка результатов учебной работы

Составляющие	Сумма баллов	Учебная деятельность студента	Оценка в аттестации	Баллы (18 недель)
Текущая учебная деятельность	**60**	Посещение лекций (3 занятия).	**1 балл** (присутствие на лекции) **2 балла** (активная работа, конспектирование)	3 - 6
		Лабораторные работы (3 работы).	**1 балл** (посещение занятия, выполнение работы на 51-65%) **2 балла** (существенный вклад на занятии относительно всей группы, самостоятельность при выполнении работы, выполнение работы на 85,1-100%)	3 - 6
		Реферат	**5 баллов** (пороговое значение) **10 баллов** (максимальное значение)	5 - 10
		Контрольная работа	**10 баллов** (пороговое значение) **18 баллов** (максимальное значение)	10 - 18
		Разработка программного проекта	**10 баллов** (пороговое значение) **20 баллов** (максимальное значение)	10-20
Итого по текущей работе в семестре				**31 - 60**
Промежуточ-ная аттестация (экзамен)	**40**	Теоретический вопрос 1.	**5 баллов** (пороговое значение) **10 баллов** (максимальное значение)	5 - 10
		Теоретический вопрос 2.	**5 баллов** (пороговое значение) **5 баллов** (максимальное значение)	5 - 10
		Защита программного проекта	**10 баллов** (пороговое значение) **20 баллов** (максимальное значение)	10 - 20
Итого по промежуточной аттестации (экзамен)				**20 – 40 б.**
Суммарная оценка по дисциплине: Сумма баллов текущей и промежуточной аттестации				**51 – 100 б.**

Соотношение между оценками в баллах и их числовыми и буквенными эквивалентами устанавливается следующим образом:

Перевод баллов из 100-балльной шкалы в числовой и буквенный эквивалент

Сумма баллов для дисциплины	*Оценка*	*Буквенный эквивалент*
86 - 100	5	отлично
66 - 85	4	хорошо
51 - 65	3	удовлетворительно
0 - 50	2	неудовлетворительно

5 Учебно-методическое обеспечение дисциплины.

5.1 Учебная литература

Основная учебная литература

1. Белов, В. В. Алгоритмы и структуры данных : учебник / В. В. Белов, В. И. Чистякова. - Москва : КУРС : ИНФРА-М, 2020. - 240 с. - (Бакалавриат). - ISBN 978-5-906818-25-6. - Текст : электронный. - URL: https://znanium.com/catalog/product/1057212 (дата обращения: 27.10.2020). – Режим доступа: по подписке.
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 (дата обращения: 27.10.2020). – Режим доступа: по подписке.

Дополнительная учебная литература

1. Засорин, С. В. Структуры и алгоритмы компьютерной обработки данных. Лабораторный практикум: Учебное пособие / Засорин С.В., Ломтева О.А. - Москва :КУРС, 2018. - 384 с. (Бакалавриат) ISBN 978-5-907064-14-0. - Текст : электронный. - URL: https://znanium.com/catalog/product/977719 (дата обращения: 27.10.2020)
2. 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/105400 7 (дата обращения: 27.10.2020). – Режим доступа: по подписке.
3. Царев, Р. Ю. Алгоритмы и структуры данных (CDIO): Учебник / Царев Р.Ю., Прокопенко А.В. - Краснояр.:СФУ, 2016. - 204 с.: ISBN 978-5-7638-3388-1. - Текст : электронный. - URL: https://znanium.com/catalog/product/967108 (дата обращения: 27.10.2020). – Режим доступа: по подписке.

5.2 Материально-техническое и программное обеспечение дисциплины.

Учебные занятия по дисциплине проводятся в учебных аудиториях НФИ КемГУ:

Структуры и алгоритмы компьютерной обработки данных	303 Компьютерный класс. Учебная аудитория (мультимедийная) для проведения занятий: - занятий лекционного типа; - семинарского (практического) типа. - текущего контроля и промежуточной аттестации Специализированная (учебная) мебель:	654027, Кемеровская область - Кузбасс, г. Новокузнецк, пр-кт Пионерский, д.13, пом.2

	доска маркерно-меловая, столы компьютерные, стулья. Оборудование для презентации учебного материала: стационарное - ноутбук преподавателя, экран, проектор. Оборудование: компьютеры для обучающихся (11 шт.). Используемое программное обеспечение: MSWindows (MicrosoftImaginePremium 3 year по сублицензионному договору № 1212/КМР от 12.12.2018 г. до 12.12.2021 г.), LibreOffice (свободно распространяемое ПО), Firefox 14 (свободно распространяемое ПО), Яндекс.Браузер (отечественное свободно распространяемое ПО), UML-диаграммы (бесплатная версия), Dia(свободно распространяемое ПО), AdobeReaderXI (свободно распространяемое ПО), WinDjView(свободно распространяемое ПО) Интернет с обеспечением доступа в ЭИОС.	

5.3 Современные профессиональные базы данных и информационные справочные системы.

Перечень СПБД и ИСС по дисциплине

1. ALGOLIST.MANUAL.RU - Сайт, посвященный алгоритмам и методам. [Электронный ресурс]. – Режим доступа: http://algolist.manual.ru
2. Хранилище документации Майкрософт для пользователей, разработчиков и ИТ-специалистов [Электронный ресурс]: информационно-справочная онлайн система / Компания Microsoft. – Режим доступа: https://docs.microsoft.com/ru-ru/

6 Иные сведения и (или) материалы.

6.1.Примерные темы письменных учебных работ

Темы рефератов

1. Типы данных C++: логический, целые, символьные, вещественные, тип void. Переименование типов.
2. Одномерные и многомерные статические массивы в C++.
3. Обработка символов и строк в C++. Функции стандартной библиотеки.
4. Перечисления в C++. Структуры. Объединения.
5. Алгоритмы сортировки (пузырьковая, подсчетом, выбором).
6. Алгоритмы сортировки (вставками, слиянием, быстрая).
7. Алгоритмы сортировки (поразрядная, пирамидальная).
8. Последовательный, бинарный, интерполяционный поиск.
9. Точный поиск подстроки в строке. Нечеткий поиск.
10. Динамические структуры данных: список, очередь, стек.
11. Динамические структуры данных: словари, множества.
12. Поиск на графе и его обход.
13. Графы. Нахождение кратчайших путей.
14. Нахождение на графе минимального остовного дерева.

15. Нахождение на графе максимального пропускного потока.

Программный проект (индивидуальное задание)

Разработать программу для решения задачи, указанной в вашем варианте задания. Каждая используемая при решении задачи структура данных должна быть оформлена как самостоятельный тип — класс с набором операций, необходимых для работы с этой структурой. Класс (или иерархия родственных классов) размещается в отдельном модуле. В методах класса и вспомогательных подпрограммах необходимо предусмотреть обработку исключительных ситуаций, возникающих при работе со структурами данных.

Примерные темы программных проектов

1. Программа преобразования выражения из инфиксной формы в постфиксную.
   Программа выполняет следующее:
   – считывает из внешнего файла последовательность выражений в инфиксной форме;
   – преобразует форму представления выражения из полноскобочной инфиксной в постфиксную;
   – осуществляет в процессе перевода синтаксический контроль правильности записи выражения;
   – помещает результаты работы во внешний файл.
   Исходное выражение содержит идентификаторы с числом буквенно-цифровых символов <=6, целые десятичные константы (<=32767) и операции +, –, *, /, ^; операнды и операции отделяются друг от друга любым количеством пробелов.
   При синтаксическом контроле проверяются: длина идентификатора (<=6), правильность его записи (первый символ — буква), значение константы (<=32767), наличие более одного подряд записанного знака операции, парность и правильность вложения скобок.
   Программа использует стек на динамических переменных.

2. Программа, имитирующая работу условной вычислительной машины:
   Программа выполняет следующее:
   – считывает из внешнего файла последовательность выражений в постфиксной форме;
   – генерирует последовательность команд условной вычислительной машины, необходимых для вычисления выражения, вычисляет с их помощью значение выражения и оставляет результат в регистре;
   – переводит выражение из постфиксной записи в полноскобочную инфиксную;
   – помещает результаты работы во внешний файл.
   Исходное выражение содержит идентификаторы с числом буквенно-цифровых символов не больше шести, целые десятичные константы в диапазоне –32768 … 32767 и операции +, –, *, /; операнды и операции отделяются друг от друга любым количеством пробелов.
   Условная вычислительная машина имеет один регистр и следующую систему команд:
   LD A – помещает операнд A в регистр;
   ST A – помещает содержимое регистра в переменную A;
   AD A – прибавляет содержимое переменной A к регистру;
   SB A – вычитает содержимое переменной A из регистра;
   ML A – умножает содержимое регистра на переменную A;
   DV A – делит содержимое регистра на переменную A.
   Программа использует стек на динамических переменных;

3. Программа моделирования работы магазина самообслуживания

Составить программу моделирования работы магазина самообслуживания с заданным числом касс K. Каждый покупатель подходит в момент времени t1 к свободной кассе или кассе с наиболее короткой очередью и требует для своего обслуживания времени t2 (определяемого числом покупок).
Время прибытия покупателей и время обслуживания задаются в минутах (t1 задается с момента открытия магазина) и распределяются по нормальному закону с заданным средним значением M и отклонением S.
Программа должна определить среднее, минимальное и максимальное время обслуживания (время нахождения в очереди плюс время расчета) одного покупателя (для K = 3 и K = 5).
Исходные данные: K, M, S, P (число покупателей). Параметры t1 и t2 для каждого покупателя определяются в программе с помощью генератора случайных чисел.

4. Программа выполнения арифметических операций с длинными числами
Составить программу алгебраического сложения и умножения длинных целых чисел с использованием двунаправленных связанных списков. Каждый элемент списка должен быть целым значением с максимально допустимым в используемой системе программирования количеством цифр.
Наборы исходных длинных целых чисел и операции, которые необходимо выполнять над ними, построчно вводятся из внешнего текстового файла в формате
число1_операция_число2 (_ — символ пробела)
число3_операция_число4
...
например:
–1592678 + 27829115
178591 * –1756
2997518 — –324277

5. Программа пирамидальной сортировки
Составить программу упорядочения методом пирамидальной сортировки односвязного списка, каждый элемент которого состоит из двух полей: номер группы и фамилия студента.
Первоначально осуществить сортировку по возрастанию номера факультета, затем по возрастанию номера курса, далее по возрастанию номера группы на факультете. Внутри каждой группы упорядочить фамилии по алфавиту.
Размер рабочего исходного текстового файла — 100 записей.

6. Программа внутренней сортировки методом Неймана
Сравнить по быстродействию реализации метода Неймана (естественное слияние) для сортировки элементов односвязного списка и одномерного массива одинаковой длины.
В характеристики быстродействия включить число операций сравнения, число перестановок для массива и переключения связей для списка.
Типы данных и их значения выбрать произвольно. Длины массивов и списков принять равными 100, 200, 500 (для их формирования использовать генератор случайных чисел).

7. Программа быстрой сортировки
Оценить характеристики быстрой сортировки (число сравнений ключей и число перестановок элементов) для сортировки:
1) упорядоченного массива;
2) случайного массива;

3) упорядоченного в обратном порядке массива.
Размер сортируемого массива: 10, 50, 100, 500 (для формирования использовать генератор случайных чисел).

8. Программа поразрядной сортировки
   Оценить характеристики метода поразрядной сортировки числового массива с использованием связанных списков. Сортировка выполняется сначала для младших значащих цифр, затем полученные списки сливаются в файл и сортируются по следующей значащей цифре, и т.д. до старшей цифры. В характеристики быстродействия включить: число операций сравнения, число переключений связей для элементов списков (аналог числа перестановок). Исходный массив состоит из 500 случайных чисел, равномерно распределенных в диапазоне от 0 до 9999.
   Сравнить по быстродействию поразрядную сортировку с разбиением файла на 10 связанных списков по старшей значащей цифре с включением элементов в список в порядке, не нарушающем упорядоченность списка.

9. Программа индексно-последовательного поиска
   Составить программу формирования таблицы символических имен в виде индексно-последовательного файла и последующего поиска имени в таблице.
   Сравнить по среднему времени поиска (аналог — среднее число просмотренных при поиске элементов) реализации индексно-последовательного поиска:
   1) число индексов: 1 и 2;
   2) размер индекса: 3 и 10.
   Файл с исходным списком символических имен сформировать с помощью подпрограммы подготовки данных следующим образом:
   проанализировать текст любой программы, содержащей не менее 200–300 строк, и включить в таблицу все ключевые слова и идентификаторы программы.

10. Программа поиска по дереву
    Составить программу формирования таблицы имен в виде дерева поиска и последующего поиска имени в таблице.
    Оценить выигрыш по скорости поиска для оптимального дерева поиска (построенного по предварительно отсортированной и считанной с последовательными отступлениями влево и вправо от середины в последовательности имен) по сравнению с деревом, сформированным при произвольном порядке поступления входных имен.
    В качестве аналога скорости поиска принять среднее число просмотренных при поиске элементов. Оценить проигрыш в скорости построения дерева.
    Файл с исходным списком символических имен сформировать с помощью подпрограммы подготовки данных следующим образом:
    проанализировать текст любой программы, содержащей не менее 200–300 строк, и включить в таблицу все ключевые слова и идентификаторы программы.

11. Программа поиска методом расстановки

    Составить программу формирования таблицы символических имен методом расстановки с раздельным сцеплением и последующего поиска имени в таблице.
    Сравнить по скорости поиска в таблице (в качестве аналога скорости поиска принять среднее число просмотренных при поиске элементов) различные варианты выбора функции расстановки, проанализировать 3–4 варианта.
    Оценить коэффициент заполнения таблицы и среднюю длину поиска при занесении 25%, 50%, 75% и 100% имен из файла. Оптимизировать параметры функции расстановки по результатам этих исследований.

Файл с исходным списком символических имен сформировать с помощью подпрограммы подготовки данных следующим образом: проанализировать текст любой программы, содержащей не менее 200–300 строк, и включить в таблицу все ключевые слова и идентификаторы программы.

12. Программа нахождения максимального пути в ориентированном графе
    Составить программу нахождения всех максимальных путей во взвешенном ориентированном графе с одним начальным и одним конечным узлом, используя представление графа с помощью связанного списка заголовочных узлов, каждый из которых является головой списка дуговых узлов.
    Минимальный набор операций для структуры — графа должен включать следующие операции:
    – формирование пустого графа;
    – проверка наличия узлов в графе;
    – проверка наличия заданного узла в графе;
    – проверка наличия дуги между заданными узлами графа;
    – включение нового узла в граф;
    – исключение узла из графа;
    – включение дуги с заданным весом между двумя заданными узлами (если дуга существует, то устанавливается нужный вес);
    – исключение дуги между заданными узлами с выдачей веса дуги.
    Дополнительно включаются все операции, необходимые для работы с графом при решении задачи.
    Результаты работы программы должны быть представлены в виде списка узлов, принадлежащих максимальным путям, и значений соответствующих максимальных путей.

13. Программа нахождения минимальных расстояний между узлами графа
    Составить программу нахождения минимальных расстояний между всеми или заданными узлами неориентированного взвешенного графа, используя представление графа с помощью связанного списка заголовочных узлов, каждый из которых является головой списка дуговых узлов.
    Минимальный набор операций для структуры — графа должен включать следующие операции:
    – формирование пустого графа;
    – проверка наличия узлов в графе;
    – проверка наличия заданного узла в графе;
    – проверка наличия дуги между заданными узлами графа;
    – включение нового узла в граф;
    – исключение узла из графа;
    – включение дуги с заданным весом между двумя заданными узлами (если дуга существует, то устанавливается нужный вес);
    – исключение дуги между заданными узлами с выдачей веса дуги.
    Дополнительно включаются все операции, необходимые для работы с графом при решении задачи.
    Результаты работы программы должны быть представлены в виде списка вершин, принадлежащих минимальным путям, и значений соответствующих минимальных путей.

14. Программа планирования действий
    Составить программу планирования действий при решении задачи, представленной ориентированным невзвешенным графом работ при наличии любого необходимого

числа исполнителей этих работ, используя представление графа с помощью связанного списка заголовочных узлов, каждый из которых является головой списка дуговых узлов.
Результаты решения выдавать в виде перечня работ, которые необходимо выполнить в каждый очередной момент времени. Спланировать действия при наличии только одного исполнителя, при этом выводится порядок решения работ (задача топологической сортировки).
Минимальный набор операций для структуры — графа должен включать следующие операции:
– формирование пустого графа;
– проверка наличия узлов в графе;
– проверка наличия заданного узла в графе;
– проверка наличия дуги между заданными узлами графа;
– включение нового узла в граф;
– исключение узла из графа;
– включение дуги с заданным весом между двумя узлами.
Дополнительно включаются все операции, необходимые для работы с графом при решении задачи.

15. Программа построения сетевого графика и определения характеристик его работ при неограниченном числе исполнителей.
Составить программу нахождения основных характеристик сетевого графика выполнения некоторой задачи, представляющего собой ориентированный взвешенный граф, в котором каждая дуга представляет работу, вес дуги — время выполнения этой работы, каждый узел представляет время, к которому могут быть завершены работы, оканчивающиеся в этом узле. Использовать представление графа с помощью связанного списка заголовочных узлов, каждый из которых является головой списка дуговых узлов.
Характеристики сетевого графика:
– самые ранние моменты времени завершения работ, оканчивающихся в каждом узле;
– самые поздние моменты времени завершения работ, оканчивающихся в каждом узле;
– перечень узлов, у которых эти характеристики равны (критический путь).
Минимальный набор операций для структуры — графа должен включать следующие операции:
– формирование пустого графа;
– проверка наличия узлов в графе;
– проверка наличия заданного узла в графе;
– проверка наличия дуги между заданными узлами графа;
– включение нового узла в граф;

– исключение узла из графа;
– включение дуги с заданным весом между двумя заданными узлами (если дуга существует, то устанавливается нужный вес);
– исключение дуги между заданными узлами с выдачей веса дуги.
Дополнительно включаются все операции, необходимые для работы с графом при решении задачи.

6.2. Примерные вопросы и задания / задачи для промежуточной аттестации

Таблица 9 - Примерные теоретические вопросы и практические задания / задачи к экзамену

Разделы и темы	Примерные теоретические вопросы	Примерные практические задания / задачи
1. Структуры компьютерной обработки данных		
1.1 Структурная организация данных	1. Основные понятия структур данных. Классификация структур данных. 2. Типы и структуры данных в современных языках программирования высокого уровня.	Типовое задание (программный проект, темы см. п. 6.1)
1.2 Линейные структуры данных	3. Односвязный линейный список. Операции над односвязным линейным списком. 4. Двусвязный линейный список. Операции над двусвязным линейным списком. 5. Односвязный циклический список. Операции над односвязным циклическим списком. 6. Двусвязный циклический список. Операции над двусвязным циклическим списком. 7. Стек. Операции над стеком. 8. Очередь. Операции над очередью. 9. Дек. Операции над деком.	Типовое задание (программный проект, темы см. п. 6.1)
1.3 Нелинейные структуры данных	10. Бинарные деревья. Операции над бинарным деревом. 11. Дерево поиска. Операции над деревом поиска.. 12. Графы. Основные понятия и определения. 13. Операции над графом. Обход ориентированного графа.	Типовое задание (программный проект, темы см. п. 6.1)
2. Алгоритмы компьютерной обработки данных		
2.1 Алгоритмы сортировки	14. Алгоритмы сортировки. Сортировка выбором. Сортировка вставкой. Сортировка слиянием. 15. Алгоритмы сортировки. Сортировка обменом. Шейкерная сортировка. Сортировка Шелла. 16. Быстрая сортировка (сортировка Хоара). Турнирная сортировка. Пирамидальная сортировка.	Типовое задание (программный проект, темы см. п. 6.1)
2.2 Алгоритмы поиска	17. Алгоритмы поиска. Последовательный поиск. Бинарный поиск. 18. Алгоритмы поиска. Фибоначчиев поиск. Интерполяционный поиск. 19. Поиск по бинарному дереву. Поиск по бору. Поиск хешированием.	Типовое задание (программный проект, темы см. п. 6.1)
2.3 Алгоритмы на графах	20. Структуры для представления графа. Обходы графа в глубину и ширину. 21. Поиск путей в графе. Алгоритм Дейкстры.	Типовое задание (программный проект, темы см. п. 6.1)

Разделы и темы	Примерные теоретические вопросы	Примерные практические задания / задачи
	22. Поиск путей в графе. Алгоритм Флойда. 23. Алгоритм построения минимального остовного дерева. 24. Алгоритмы на графах. Нахождение максимального потока.	

Составитель (и): Бойченко Г.Н, доцент кафедры ИОТД