

**КАЗАНСКИЙ КООПЕРАТИВНЫЙ ИНСТИТУТ (ФИЛИАЛ)
АВТОНОМНОЙ НЕКОММЕРЧЕСКОЙ ОБРАЗОВАТЕЛЬНОЙ
ОРГАНИЗАЦИИ ВЫСШЕГО ОБРАЗОВАНИЯ
ЦЕНТРОСОЮЗА РОССИЙСКОЙ ФЕДЕРАЦИИ
«РОССИЙСКИЙ УНИВЕРСИТЕТ КООПЕРАЦИИ»**

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

МЕТОДЫ ПРОГРАММИРОВАНИЯ

Специальность: 10.05.01 Компьютерная безопасность

Специализация: «Разработка защищенного программного обеспечения»

Формы обучения: очная

Объем дисциплины:

в зачетных единицах: 5 з.е.

в академических часах: 180 ак.ч.

Форма промежуточной аттестации: экзамен

Рабочая программа по дисциплине «Методы программирования» по специальности 10.05.01 Компьютерная безопасность, специализации «Разработка защищенного программного обеспечения», составлена в соответствии с требованиями Федерального государственного образовательного стандарта высшего образования – специалитет по специальности 10.05.01 Компьютерная безопасность, утвержденного приказом Министерства науки и высшего образования Российской Федерации от «26» ноября 2020 г. № 1459, профессионального стандарта 06.032 «Специалист по безопасности компьютерных систем и сетей», утвержденного приказом Министерства труда и социальной защиты Российской Федерации от 14.09.2022 г. № 533н, профессионального стандарта 06.033 «Специалист по защите информации в автоматизированных системах», утвержденного приказом Министерства труда и социальной защиты Российской Федерации от 14.09.2022 г. №525н.

Рабочая программа:

обсуждена и рекомендована к утверждению решением кафедры естественных дисциплин, сервиса и туризма от 25 марта 2025 г., протокол № 6

одобрена Научно-методическим советом Казанского кооперативного института (филиала) от 2 апреля 2025 г., протокол № 3

СОДЕРЖАНИЕ

1. Цель и задачи освоения дисциплины	4
2. Место дисциплины в структуре образовательной программы	4
3. Перечень планируемых результатов обучения по дисциплине	5
4. Объем дисциплины и виды учебной работы.....	5
5. Содержание дисциплины, структурированное по темам (разделам) с указанием количества академических часов и видов учебных занятий.....	6
5.1. Содержание дисциплины	6
5.2. Разделы, темы дисциплины и виды занятий	8
6. Практические занятия	11
7. Лабораторные работы	12
8. Примерная тематика курсовых работ (проектов)	12
9. Самостоятельная работа студента	12
10. Оценивание результатов обучения и уровня сформированности компетенций	20
12. Перечень программного обеспечения, профессиональных баз данных и информационных справочных систем, необходимых для освоения дисциплины.....	22
13. Материально–техническое обеспечение дисциплины	22
Обновление рабочей программы дисциплины (модуля)	24

1. Цель и задачи освоения дисциплины

Целью освоения дисциплины является приобретение знаний о базовых принципах объектно-ориентированного программирования и получение практических навыков программирования на языке высокого уровня.

Формирование компетенции обучающегося в области использования компьютера как средства управления информацией; изучение методов программирования для овладения знаниями в области технологии программирования; подготовка обучающихся к осознанному применению, как языков программирования, так и методов программирования.

Создание необходимой основы для использования современных средств вычислительной техники и прикладных программ при изучении студентами естественнонаучных, общепрофессиональных и специальных дисциплин.

Освоение, предусмотренного программой теоретического материала и приобретение практических навыков использования информационных систем, языков программирования и технологий на базе современных ПК.

Задачи дисциплины.

- Основные задачи курса на основе системного подхода;
- знакомство с методами структурного и объектно-ориентированного программирования как наиболее распространенными и эффективными методами разработки программных продуктов;
- обучение разработке алгоритмов на основе структурного и объектноориентированного подхода;
- закрепление навыков алгоритмизации и программирования на основе изучения современных языков программирования;
- знакомство с основными структурами данных и типовыми методами обработки этих структур.

2. Место дисциплины в структуре образовательной программы

Дисциплина «Методы программирования» относится к части, формируемой участниками образовательных отношений Блока 1 «Дисциплины (модули)» основной образовательной программы – программы специалитета по специальности 10.05.01 Компьютерная безопасность безопасность специализация «Разработка защищенного программного обеспечения».

Код компетенции	Предшествующие дисциплины (модули), практики	Изучаемые в текущем семестре дисциплины (модули), практики	Последующие дисциплины (модули), практики
-----------------	--	--	---

ОПК-7	-	Методы программирования	Интерпретируемые языки программирования Дискретное программирование
-------	---	------------------------------------	---

3. Перечень планируемых результатов обучения по дисциплине

Изучение дисциплины направлено на формирование у обучающихся профессиональных компетенций.

Формируемые компетенции (код и наименование компетенции)	Индикаторы достижения компетенций	Планируемые результаты обучения
ОПК-7 Способен создавать программы на языках высокого и низкого уровня, применять методы и инструментальные средства программирования для решения профессиональных задач, осуществлять обоснованный выбор инструментария программирования и способов организации программ	ОПК-7.3 Демонстрирует умение создавать программы на языках низкого уровня, при менять методы и инструментальные средства реверсинжиниринга для решения профессиональных задач	Знать: Типовые компьютерные методы и современное программное обеспечение для решения задач профессиональной деятельности на языке программирования высокого уровня C++. Уметь: Установить и использовать программное обеспечение для решения задач профессиональной деятельности, в частности Microsoft Visual Studio. Применять: методы компьютерного проектирования, разработки структур данных с использованием основных алгоритмических и программных решений в области информационно-коммуникационных технологий на языке высокого уровня C++. Владеть: Проектирование и использование структур данных при реализации алгоритмических и программных решений на языке программирования высокого уровня C++.
	ОПК- 7.4 Демонстрирует умение создавать программы на языках высокого уровня, при менять методы визуального объектного моделирования в области разработки программ и соответствующие инструментальные средства в процессе решения профессиональных задач	Знать: Типовые методы и приемы поиска, анализа и синтеза информации необходимой для решения профессиональных алгоритмических задач. Уметь: Выполнять поиск, анализ и синтез научно-практической информации, полученной из достоверных источников. Применять результат поиска для решения профессиональных задач. Владеть: Применение методов системного подхода для решения профессиональных задач, в частности в области разработки программного обеспечения.

4. Объем дисциплины и виды учебной работы

Объем дисциплины и виды учебной работы в академических часах с выделением объема контактной работы обучающихся с преподавателем и

самостоятельной работы обучающихся

очная форма обучения

Вид учебной деятельности	ак. часов	
	Всего	По семестрам
		2
1. Контактная работа обучающихся с преподавателем:	103	103
Аудиторные занятия, часов всего, в том числе:	102	102
• занятия лекционного типа	50	50
• занятия семинарского типа:	52	52
практические занятия	52	52
лабораторные занятия	-	-
в том числе занятия в форме практической подготовки	-	-
Консультации	0,5	0,5
Контактные часы на аттестацию в период экзаменационных сессий	0,5	0,5
2. Самостоятельная работа студентов всего, в том числе	41	41
- подготовка курсовой работы	-	-
- проработка учебного (теоретического) материала	20	20
- выполнение домашних заданий по практическим занятиям	21	21
- подготовка к экзамену	-	-
3. Контроль:	36	36
Промежуточная аттестация: экзамен	-	-
ИТОГО:	180	180
общая трудоемкость	часов	5

5. Содержание дисциплины, структурированное по темам (разделам) с указанием количества академических часов и видов учебных занятий

5.1. Содержание дисциплины

Тема 1. Линейные динамические информационные структуры

Информационные структуры. Линейные динамические информационные структуры. Основные определения. Примеры.

Тема 2. Стек. Очередь. Дек.

Стек. Очередь. Дек. Логическая организация. Основные операции. Абстрактные структуры.

Тема 3. Моделирование стека средствами языка C++.

Моделирование стека средствами языка C++. Набор основных операций

Тема 4. Моделирование очереди средствами языка C++.

Моделирование очереди средствами языка C++. Набор основных

операций

Тема 5. Кольцевой буфер

Кольцевой буфер. Моделирование. Набор основных операций.

Тема 6. Реализация двух однотипных стеков

Реализация двух однотипных стеков. Набор основных операций

Тема 7. Однонаправленные списки. Операция вставки и удаления элемента

Связное распределение памяти. Однонаправленные списки. Построение списка. Операция вставки и удаления элемента.

Тема 8. Двухнаправленные списки. Построение, операция удаления и вставки элемента

Двухнаправленные списки. Построение, операция удаления элемента. Построение, операция вставки элемента. Рекурсивная обработка списков.

Тема 9. Деревья. Дерево двоичного поиска. Обходы ДДП

Деревья. Основные понятия. Дерево двоичного поиска (структура и построение). Обходы ДДП.

Тема 10. Обработка дерева двоичного поиска.

Обработка дерева двоичного поиска (поиск и замена элемента, вычисление суммы или количества заданных элементов).

Тема 11. Система управления вводом-выводом. Буферизация ввода-вывода

Ввод-вывод. Система управления вводомвыводом. Блок FCB. Открытие и закрытие файлов. Буферизация ввода-вывода

Тема 12. Общие операции над файлами

Общие операции над файлами. Обработка текстовых, числовых бинарных файлов.

Тема 13. Ориентированные графы. Представления ориентированных графов.

Ориентированные графы. Основные определения. Представления ориентированных графов. АТД для ориентированных графов. Основные операторы.

Тема 14. Задача нахождения кратчайшего пути.

Алгоритм Дейкстры. Задача нахождения кратчайшего пути. Алгоритм Дейкстры. Реализация на C++.

Тема 15. Нахождение кратчайших путей между парами вершин.

Алгоритм Флойда.

Нахождение кратчайших путей между парами вершин. Алгоритм Флойда. Реализация на C++.

Тема 16. Поиск центра ориентированного графа.

Поиск центра ориентированного графа. Реализация на C++.

Тема 17. Обход ориентированных графов. Поиск в глубину.

Обход ориентированных графов. Поиск в глубину. Реализация на C++.

Тема 18. Алгоритм нахождения сильно связанных компонент.

Алгоритм нахождения сильно связанных компонент. Реализация на C++.

Тема 19. Представление неориентированных графов. Основные деревья минимальной стоимости.

Представление неориентированных графов. Основные деревья минимальной стоимости. Основные понятия. Структура дерева

Тема 20. Алгоритм Прима. Алгоритм Крускала.

Алгоритм Прима. Алгоритм Крускала. Реализация на C++.

Тема 21. Контейнерный класс vector.

Контейнерный класс vector. Основные методы, примеры использования.

Тема 22. Контейнерный класс list.

Контейнерный класс list. Основные методы, примеры использования.

Тема 23. Ассоциативные контейнеры set и map.

Ассоциативные контейнеры set и map. Основные методы, примеры использования.

Тема 24. Общая схема компиляции.

Общая схема компиляции. Информационные таблицы. Сканер. Лексический анализатор. Синтаксический и семантический анализатор. Функция хеширования. Метод цепочек переполнения

5.2. Разделы, темы дисциплины и виды занятий

очная форма обучения

№ п/п	Наименование раздела, темы дисциплины	Виды занятий, включая самостоятельную работу студентов (в часах)	Индикаторы достижения
-------	---------------------------------------	--	-----------------------

		занятия лекционного типа	занятия семинарского типа / из них в форме практической подготовки (лабораторные работы)	самос- тоятельная работа	компетенций
1	Линейные динамические информационные структуры	2	4	1	ОПК-7, ПРК-7.3, ОПК-7.4
2	Стек. Очередь. Дек.	2	4	2	ОПК-7, ПРК-7.3, ОПК-7.4
3	Моделирование стека средствами языка C++.	2	2	2	ОПК-7, ПРК-7.3, ОПК-7.4
4	Моделирование очереди средствами языка C++.	2	2	2	ОПК-7, ПРК-7.3, ОПК-7.4
5	Кольцевой буфер.	2	2	2	ОПК-7, ПРК-7.3, ОПК-7.4
6	Реализация двух однотипных стеков.	2	2	2	ОПК-7, ПРК-7.3, ОПК-7.4
7	Однонаправленные списки. Операция вставки и удаления элемента.	2	2	2	ОПК-7, ПРК-7.3, ОПК-7.4
8	Двунаправленные списки. Построение, операция удаления и вставки элемента.	2	2	2	ОПК-7, ПРК-7.3, ОПК-7.4
9	Деревья. Дерево двоичного поиска. Обходы ДДП.	2	2	2	ОПК-7, ПРК-7.3, ОПК-7.4
10	Обработка дерева двоичного поиска.	2	2	2	ОПК-7, ПРК-7.3, ОПК-7.4
11	Система управления Буферизация ввода-вывода	2	2	2	ОПК-7, ПРК-7.3, ОПК-7.4
12	Общие операции над файлами	2	2	2	ОПК-7, ПРК-7.3, ОПК-7.4
13	Ориентированные графы. Представления ориентированных графов.	2	2	2	ОПК-7, ПРК-7.3, ОПК-7.4
14	Задача нахождения кратчайшего пути. Алгоритм Дейкстры.	2	2	2	ОПК-7, ПРК-7.3, ОПК-7.4
15	Нахождение кратчайших путей между парами вершин. Алгоритм Флойда.	2	2	2	ОПК-7, ПРК-7.3, ОПК-7.4
16	Поиск центра ориентированного графа.	2	2	2	ОПК-7, ПРК-7.3, ОПК-7.4
17	Обход ориентированных графов. Поиск в глубину.	2	2	2	ОПК-7, ПРК-7.3, ОПК-7.4
18	Алгоритм нахождения сильно связных компонент.	2	2	1	ОПК-7, ПРК-7.3, ОПК-7.4

№ п/п	Наименование раздела, темы дисциплины	Виды занятий, включая самостоятельную работу студентов (в часах)			Индикаторы достижения компетенций
		занятия лекционного типа	занятия семинарского типа / из них в форме практической подготовки (лабораторные работы)	самос- тоятельная работа	
19	Представление неориентированных графов. Остовные деревья минимальной стоимости.	2	2	1	ОПК-7, ПРК- 7.3, ОПК-7.4
20	Алгоритм Прима. Алгоритм Крускала.	2	2	1	ОПК-7, ПРК- 7.3, ОПК-7.4
21	Контейнерный класс vector.	2	2	1	ОПК-7, ПРК- 7.3, ОПК-7.4
22	Контейнерный класс list.	4	2	1	ОПК-7, ПРК- 7.3, ОПК-7.4
23	Ассоциативные контейнеры set и map.	2	2	1	ОПК-7, ПРК- 7.3, ОПК-7.4
24	Общая схема компиляции.	2	2	1	ОПК-7, ПРК- 7.3, ОПК-7.4
	Всего	50	52	41	

6. Практические занятия

№ п/п	Наименование раздела, темы дисциплины	Содержание практических занятий	Объем (час.)
			Очная форма обучения
1.	Линейные динамические информационные структуры	Линейная структура данных тип string.	4
2.	Моделирование стека средствами языка C++.	Реализация стека. Решение задач с использованием стека	4
3.	Моделирование очереди средствами языка C++.	Реализация очереди. Решение задач с использованием очереди	2
4.	Кольцевой буфер.	Реализация кольцевого буфера. Решение задач с использованием кольцевого буфера	2
5.	Реализация двух односторонних стеков.	Реализация двух стеков. Решение задач с использованием общего стека	2
6.	Однонаправленные списки. Операция вставки и удаления элемента.	Обработка однонаправленных списков	2
7.	Двунаправленные списки. Построение, операция удаления и вставки элемента.	Обработка двунаправленных списков	2
8.	Деревья. Дерево двоичного поиска. Обходы ДДП.	Построение и обход дерева двоичного поиска	2
9.	Обработка дерева двоичного поиска.	Обработка дерева двоичного поиска.	2
10.	Система управления вводом-выводом. Буферизация ввода-вывода	Создание и обработка числовых файлов	2
11.	Общие операции над файлами	Создание и обработка текстовых файлов	2
12.	Ориентированные графы. Представления ориентированных графов.	Построение графа	2
13.	Задача нахождения кратчайшего пути. Алгоритм Дейкстры.	Решение задач на нахождение кратчайшего пути	2
14.	Нахождение кратчайших путей между парами	Решение задач на нахождение путей между двумя вершинами	2

№ п/п	Наименование раздела, темы дисциплины	Содержание практических занятий	Объем (час.)
			Очная форма обучения
	вершин. Алгоритм Флойда.		
15.	Поиск центра ориентированного графа.	Поиск центра ориентированного графа	2
16.	Обход ориентированных графов. Поиск в глубину.	Обход ориентированных графов	2
17.	Алгоритм нахождения сильно связанных компонент.	Нахождение сильно связанных компонент	2
18.	Представление неориентированных графов. Остовные деревья минимальной стоимости.	Построение и обход неориентированного графа	2
19	Алгоритм Прима. Алгоритм Крускала.	Построение остова минимальной стоимости	2
20	Контейнерный класс vector.	Обработка данных типа vector	2
21	Контейнерный класс list.	Обработка данных типа list	2
22	Ассоциативные контейнеры set и map.	Обработка данных типа Set и map	2
	Всего		52

7. Лабораторные работы

Лабораторные работы не предусмотрены

8. Примерная тематика курсовых работ (проектов)

Курсовые работы (проекты) не предусмотрены.

9. Самостоятельная работа студента

Самостоятельная работа студента при изучении дисциплины «Методы программирования» направлена на:

- освоение нормативных правовых актов и учебной литературы, необходимых для освоения дисциплины;
- самостоятельный поиск информации в Интернете и других источниках;
- выполнение домашних заданий по практическим занятиям;
- подготовку к зачету.

Краткие рекомендации по выполнению самостоятельной работы:

Успешное усвоение дисциплины предполагает большой, упорный, серьезный, систематический труд студентов. Важнейшая его составная часть – выполнение разных видов самостоятельной работы.

1. Составление тематического конспекта на основе изученной основной и дополнительной учебной литературы. В тематическом конспекте за основу берется содержание темы, вопросы для обсуждения.

Этапы работы.

1.1 Конспектирование делается только после того, как прочитан или усвоен материал для конспектирования.

1.2. Необходимо мысленно или письменно составить план конспекта. По этому плану и будет строиться конспект далее.

1.3. Составление самого конспекта. Можно сказать, что конспект – это расширенные тезисы, дополненные рассуждениями и доказательствами, содержащимися в материалах для конспекта, а также собственными мыслями и положениями составителя конспекта.

Писать конспект рекомендуется четко и разборчиво. В конспекте можно выделять места текста в зависимости от их значимости. Для этого применяются различного размера буквы, подчеркивания, замечания на полях.

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 контрольные вопросы

1. Дайте определение информационной структуры.
2. Дайте определение и приведите примеры линейных динамических информационных структур.
3. Перечислите операции работы со стеком.
4. Дайте определение абстрактному и ограниченному стеку.
5. Укажите принцип работы стека.
6. Перечислите операции работы с очередью.
7. Дайте определение абстрактной и ограниченной очереди.
8. Укажите принцип работы очереди.
9. Перечислите операции работы с деком.
10. Дайте определение абстрактному и ограниченному деку.
11. Укажите принцип работы дека.
12. Перечислите основные блоки системы управления вводом-выводом, укажите их функционал.
13. Дайте определение блока FCB.
14. Перечислите функции при открытии и закрытии файлов.
15. Кратко опишите принцип буферизации ввода-вывода
16. Опишите общую схему компиляции.
17. Укажите назначение информационных таблиц
18. Дайте определение хеш-функции.

Типовые контрольные задания

1. Дан текст. Подсчитать число вхождений буквы f в первые три слова, в слове должны быть только буквы, слова разделяются пробелами.
2. Дан текст. Найти число таких слов (в слове только буквы), которые начинаются и оканчиваются одной и той же буквой, причем начальная буква заглавная, слова разделяются пробелами.
3. Дан текст. Найти все такие слова (в слове только буквы), в которые буква 'а' входит не менее двух раз, слова разделяются пробелами.
4. Дано несколько строк без пробелов. Напечатать строки, в которые входят знаки «+», «-», «*» в первую половину, а вторая половина строки не содержит цифр.
5. Дано несколько строк. Заменить все цифры, находящиеся в первой половине строки на «*», а из второй половины строки удалить все «.».
6. Дано несколько строк. Найти количество симметричных строк, не содержащих цифры.
7. Дано несколько строк. Если длина строки нечетна, то удалить средний символ, иначе вставить в середину строки «авс».
8. На вход программе подается предложение (посимвольно) на английском языке, заканчивающееся точкой. Требуется написать программу, которая определяет, можно ли переставить английские буквы этого предложения (не учитывая остальные символы) так, чтобы получить палиндром – слово, которое читается одинаково слева направо и справа налево. Строчные и прописные буквы не различаются. Если это сделать нельзя, программа должна вывести на экран слово «Нет», а если можно – слово «Да» и в следующей строчке искомое слово прописными буквами. Если таких слов несколько, нужно вывести последнее в алфавитном порядке слово. Например, если исходная последовательность была такая: Deed daad N.

то результат должен быть следующий:

Да

EDDANADDE

9. На вход программе подается строка (длиной не более 200 символов, посимвольно), в которой нужно зашифровать все английские слова (словом называется непрерывная последовательность английских букв, слова друга от друга отделяются любыми другими символами, длина слова не превышает 20 символов). Строка заканчивается символом #, других символов # в строке нет. Каждое слово зашифровано с помощью циклического сдвига на длину этого слова. Например, если длина слова равна K, каждая буква в слове заменяется на букву, стоящую в английском алфавите на K букв дальше (алфавит считается циклическим, то есть, за буквой Z стоит буква A). Строчные буквы при этом остаются строчными, а прописные – прописными. Символы, не являющиеся английскими буквами, не изменяются.

Требуется написать программу, которая будет выводить на экран текст зашифрованного сообщения. Например, если исходный текст был таким: Day, mice. "Year" is a mistake# то результат шифровки должен быть следующий: Gdb, qmgi. "Ciev" ku b tpzahr!#

10. Заменить отрицательные элементы однонаправленного списка, за которыми следуют нечетные элементы, модулями этих отрицательных чисел.
11. Найти количество кратных 5 элементов, значение которых превосходит среднее арифметическое положительных элементов списка.
12. Заменить некратные 3 элементы списка, суммой нечетных элементов.
13. Проверить список на упорядоченность по возрастанию.
14. Увеличить двузначные положительные, за которыми следуют отрицательные на значение наибольшего элемента списка.
15. Удалить все положительные элементы, за которыми следуют четные.
16. Удалить все кратные 5 элементы, до которых идет отрицательный элемент.
17. Удалить все двузначные элементы, расположенные между равными элементами.
18. Вставить новый элемент между двумя соседними четными элементами.
19. Вставить элемент равный 5 до каждого нечетного элемента списка.
20. Удалить из списка все простые элементы.
21. Вставить между двумя совершенными элементами новый элемент со значением «11».
22. Дан одномерный массив. Записать в однонаправленный список только те элементы массива, в записи которых ровно две «2». Затем из списка удалить все элементы, оканчивающиеся на 2.
23. Дано два однонаправленных списка. Получить третий список, содержащий только положительные элементы исходных списков.
24. Дано два однонаправленных, упорядоченных по возрастанию, списка. Получить третий список, также упорядоченный по возрастанию.
25. Проверить является ли список симметричным.
26. Вычислить $x_1 * x_n + x_2 * x_{n-1} + \dots + x_n * x_1$.
27. Заменить все оканчивающиеся на 5 элементы, значение последнего четного элемента (движение от последнего элемента).
28. Вычислить $\max(x_1 * x_n, x_2 * x_{n-1}, \dots, x_{n/2} * x_{n/2})$.
29. Удалить все простые (функция) элементы двунаправленного списка, расположенные между четными элементами.
30. Дан двунаправленный список. Удалить все кратные 5 элементы списка, до которых идет элемент, в записи которого нет нулей (функция).
31. Дан двунаправленный список. Вставить максимальный элемент списка перед каждым нечетным элементом.

32. Дан двунаправленный список. Вставить элемент равный 11, между двумя элементами, содержащими цифры «1».
33. Дан двунаправленный список. Построить одномерный массив, включающий только элементы массива, состоящие из одинаковых цифр (функция). Затем найти в массиве индекс максимального элемента.
34. Дан двунаправленный кольцевой список. Найти количество четных пар идущих подряд элементов.
35. Дан двунаправленный кольцевой список. Если в нем более двух элементов кратных пяти, то удалить эти элементы.
36. Дан одномерный массив целых чисел. Построить двунаправленный список, содержащий только кратные 5 элементы массива. Затем в списке удалить все элементы, оканчивающиеся на 55.
37. Дано два однонаправленных списка. Построить двунаправленный список, содержащий элементы из первого списка, не входящие во второй список.
38. Дано дерево двоичного поиска. Найти количество положительных элементов, не являющихся листьями.
39. Найти произведение делящихся на 4 элементов дерева, имеющих ровно одного сына.
40. Заменить отрицательные элементы дерева их модулями.
41. Увеличить некратные 3 элементы дерева вдвое.
42. Найти максимальный элемент дерева.
43. Увеличить в дереве все элементы, являющиеся листьями на значение количества элементов, оканчивающихся на 5.
44. Верно, ли что в дереве все элементы четные.
45. Верно, ли что в дереве нет кратных 5 листьев.
46. Верно ли, что в дереве хотя бы один лист четный.
47. Найти наибольшее количество уровней в дереве (глубину дерева).
48. Дан массив чисел. Отсортировать его с помощью дерева в порядке возрастания и в порядке убывания.
49. Дан массив чисел. Исключить из него все повторяющиеся элементы с помощью дерева.
50. Дан массив чисел. Исключить из него все четные элементы с помощью дерева и записать в новый массив в порядке убывания.
51. Дан однонаправленный список целых чисел. Построить дерево, состоящее из положительных элементов однонаправленного списка.
52. Дан однонаправленный список целых чисел. Построить дерево, состоящее из нечетных элементов однонаправленного списка. Элемент дерева должен содержать значение и порядковый номер значения в списке.
53. Дан двунаправленный список целых чисел. Построить дерево, состоящее из оканчивающихся на 2 элементов списка. Затем в дереве найти сумму всех элементов, не являющихся листьями.
54. Дан файл целых чисел. Найти сумму чисел, не оканчивающихся на 5.

55. Дан файл целых чисел. Найти максимальный элемент файла.
56. Дан файл целых чисел. Найти произведение последних цифр чисел, значение которых не кратно 3.
57. Найти произведение порядковых номеров делящихся на 8 элементов файла.
58. Дан файл целых чисел. Проверить является ли файл упорядоченным по убыванию.
59. Дан файл целых чисел. Если файл не упорядочен по возрастанию, то записать в новый файл числа, у которых последняя цифра равна предпоследней.
60. Дан файл целых чисел. Записать в новый файл квадраты чисел, значения которых больше среднего арифметического всех элементов файла.
61. Дан файл целых чисел. Записать в новый файл те числа, значение которых больше чем $2 * \min$.
62. Дан текстовый файл f. Добавить в его конец символы e, n, d.
63. Дан текстовый файл f. Подсчитать число вхождений в файл сочетаний «ab».
64. Между N пунктами ($N \leq 50$) заданы дороги длиной $A(i,j)$, где I,J-номера пунктов. Дороги проложены на разной высоте и пересекаются только в общих пунктах. В начальный момент времени из заданных пунктов начинают двигаться с постоянной скоростью M роботов ($M=2$ или 3), независимо меняя направление движения только в пунктах. Роботы управляются таким образом, чтобы минимизировать время до встречи всех роботов в одном месте. Скорость I-того робота может быть равна 1 или 2. Остановка роботов запрещена.

Задание:

Написать программу, которая:

- 1) при заданных N,M и сети дорог единичной длины (все имеющиеся $A(i,j)=1$) определяет минимальное время, через которое может произойти встреча всех M роботов, при этом начальное положение роботов и скорость их движения известны.
- 2) Выполнить те же действия, что и в п. 1, но только для различных значений $A(i,j)$.

Примечание: В случае невозможности встречи всех M роботов в одном месте ни в какой момент времени в результате выполнения программы должно быть сформировано соответствующее сообщение.

65. Найти кратчайшее расстояние между двумя вершинами в графе. Найти все возможные пути между этими двумя вершинами в графе не пересекающиеся по

а) ребрам б) вершинам

66. Лабиринт задается матрицей смежности $N*N$, где $C(i,j)=1$, если узел i связан узлом j посредством дороги. Часть узлов назначается входами, часть - выходами. Входы и выходы задаются последовательностями узлов $X(1),...,X(p)$ и $Y(1),...,Y(k)$ соответственно. Найти максимальное число людей, которых можно провести от входов до выходов таким образом, чтобы:

- а) их пути не пересекались по дорогам, но могут пересекаться по узлам;
- б) их пути не пересекались по узлам.

67. В заданном графе необходимо определить, существует ли цикл, проходящий по каждому ребру графа ровно один раз.

68. Имеется N городов. Для каждой пары городов (I, J) можно построить дорогу, соединяющую эти два города и не заходящие в другие города. Стоимость такой дороги $A(I, J)$. Вне городов дороги не пересекаются.

Написать алгоритм для нахождения самой дешевой системы дорог, позволяющей попасть из любого города в любой другой. Результаты задавать таблицей $B[1:N, 1:N]$, где $B[I, J]=1$ тогда и только тогда, когда дорогу, соединяющую города I и J , следует строить.

69. N колец сцеплены между собой (задана матрица $A(n \times n)$, $A(i, j)=1$ в случае, если кольца i и j сцеплены друг с другом и $A(i, j)=0$ иначе). Удалить минимальное количество колец так, чтобы получилась цепочка.

70. Проверить являются ли два графа изоморфными

71. Построить остовное дерево минимальной стоимости на заданном графе.

72. Использовать класс `vector`: Дан массив целых чисел. Найти произведение индексов совершенных элементов.

73. Использовать класс `vector`: Дан массив целых чисел. Уменьшить все элементы – числа Фибоначчи на значение индекса.

74. Использовать класс `vector`: Дан массив целых чисел. Найти количество элементов, у которых все цифры одинаковые.

75. Использовать класс `list`: Дан однонаправленный список. Заменить все четные элементы на значение суммы нечетных элементов списка.

76. Использовать класс `list`: Дан однонаправленный список. Если после замены всех отрицательных на их модули в списке нет четных элементов, то найти произведение элементов, оканчивающихся на 1.

77. Использовать класс `list`: Дан однонаправленный список. Уменьшить все элементы кратные трем на минимальный элемент списка.

Типовой вариант коллоквиума

1. Кольцевой буфер. Моделирование
2. Рекурсивная обработка списков
3. Деревья. Основные понятия. Дерево двоичного поиска (структура и построение). Обходы ДДП.

4. С клавиатуры задается число N , а затем матрица $N \times N$ целых элементов. Из элементов матрицы, лежащих вне главной диагонали необходимо построить однонаправленный список. Используя рекурсивную функцию 1 до каждого элемента списка, в записи которого больше двух цифр «3», вставить новый элемент, со значением Z . Определение количества цифры «3» в числе находится с помощью рекурсивной функции

2. Значение Z задается с клавиатуры. Новый список вывести.

Типовые задания для самостоятельной работы

1. Дано два однонаправленных списка целых чисел. Сформировать третий список, содержащий простые числа исходных списков.
2. Дан двунаправленный кольцевой список целых чисел. Удалить все положительные элементы, до и после которых следуют элементы с нечетной суммой цифр.
3. Дано дерево двоичного поиска. Найти количество совершенных листьев дерева, больших разности максимального и минимального элемента.
4. Дан файл *f*, компонентами которого являются целые числа. Переписать в файл *g* все совершенные числа, индексы которых являются простыми.
5. Найти в графе простой цикл максимальной длины.
6. Даны действительные числа $a_1, a_2, \dots, a_n$. Если в результате замены отрицательных элементов последовательности $a_1, a_2, \dots, a_n$ их квадратами элементы будут образовывать неубывающую последовательность, то получить сумму элементов исходной последовательности, в противном случае получить их произведение.

Зачетно-экзаменационные материалы для промежуточной аттестации
(экзамен/зачет)

Вопросы для подготовки к экзамену

1. Информационные структуры. Линейные динамические информационные структуры
2. Стек. Очередь. Дек. Логическая организация
3. Моделирование стека средствами языка C++.
4. Моделирование очереди средствами языка C++.
5. Кольцевой буфер. Моделирование
6. Реализация двух однотипных стеков.
7. Связное распределение памяти. Однонаправленные списки. Построение списка.
8. Однонаправленные списки. Операция вставки и удаления элемента.
9. Двунаправленные списки. Построение, операция удаления элемента.
10. Двунаправленные списки. Построение, операция вставки элемента.
11. Рекурсивная обработка списков.
12. Деревья. Основные понятия. Дерево двоичного поиска (структура и построение). Обходы ДДП.
13. Обработка дерева двоичного поиска (поиск и замена элемента, вычисление суммы или количества заданных элементов).
14. Ввод-вывод. Система управления вводом-выводом.

15. Блок FCB. Открытие и закрытие файлов. Буферизация ввода-вывода
16. Общие операции над файлами
17. Ориентированные графы. Основные определения. Представления ориентированных графов.
18. АТД для ориентированных графов. Основные операторы.
19. Задача нахождения кратчайшего пути. Алгоритм Дейкстры.
20. Нахождение кратчайших путей между парами вершин. Алгоритм Флойда.
21. Поиск центра ориентированного графа.
22. Обход ориентированных графов. Поиск в глубину.
23. Алгоритм нахождения сильно связанных компонент.
24. Представление неориентированных графов. Остовные деревья минимальной стоимости.
- Основныe понятия.
25. Алгоритм Прима.
26. Алгоритм Крускала.
27. Контейнерный класс vector.
28. Контейнерный класс list.
29. Ассоциативные контейнеры set и map.
30. Общая схема компиляции.

10. Оценивание результатов обучения и уровня сформированности компетенций

Оценивание результатов обучения по дисциплине и уровня сформированности компетенций (части компетенции) осуществляется в рамках текущего контроля успеваемости и промежуточной аттестации в соответствии оценочным средствам.

11. Учебно-методическое обеспечение дисциплины

Основная литература:

1. Ульянова, Н. Д. Основные принципы алгоритмизации : учебно-методическое пособие / Н. Д. Ульянова. — Брянск : Брянский ГАУ, 2020. — 56 с. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/172114> (дата обращения: 26.05.2024). — Режим доступа: для авториз. пользователей.

2. Алексеев, Ю. Е. Программирование инженерных задач на базе использования алгоритмов циклической структуры на языке C в среде VS C++. Модуль 2 : учебное пособие / Ю. Е. Алексеев, А. В. Куров. — Москва : МГТУ им. Н.Э. Баумана, 2021. — 134 с. — ISBN 978-5-7038-5142-5. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/172820> (дата обращения: 19.05.2024). — Режим доступа: для авториз. пользователей.

3. Огнева, М. В. Программирование на языке C++: практический курс : учебное пособие для вузов / М. В. Огнева, Е. В. Кудрина. — Москва : Издательство Юрайт, 2021. — 335 с. — (Высшее образование). — ISBN 978-5-534-05123-0. — Текст : электронный // ЭБС Юрайт [сайт]. — URL: <https://urait.ru/bcode/473054> (дата обращения: 19.05.2024).

4. Зыков, С. В. Программирование : учебник и практикум для вузов / С. В. Зыков. —

Москва : Издательство Юрайт, 2021. — 320 с. — (Высшее образование). — ISBN 978-5534-02444-9. — Текст : электронный // ЭБС Юрайт [сайт]. — URL: <https://urait.ru/bcode/469579> (дата обращения: 19.05.2024).

Дополнительная литература:

1. 1. Павловская, Т. А. Программирование на языке C++ : учебное пособие / Т. А. Павловская. — 2-е изд. — Москва : ИНТУИТ, 2016. — 154 с. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/100409> (дата обращения: 19.05.2024). — Режим доступа: для авториз. пользователей.

2. Керниган, Б. В. Язык программирования С : учебник / Б. В. Керниган, Д. М. Ричи. — 2-е изд. — Москва : ИНТУИТ, 2016. — 313 с. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/100543> (дата обращения: 19.05.2024). — Режим доступа: для авториз. пользователей.

3. Страуструп, Б. Язык программирования C++ для профессионалов : учебное пособие / Б. Страуструп. — 2-е изд. — Москва : ИНТУИТ, 2016. — 670 с. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/100542> (дата обращения: 19.05.2024). — Режим доступа: для авториз. пользователей.

Электронные библиотечные системы (ЭБС) и электронные образовательные ресурсы

1. Электронно-библиотечная система ZNANIUM <https://www.znanium.com>

2. Универсальная справочно-информационная полнотекстовая база данных периодических изданий «East View» (ИВИС) <https://dlib.eastview.com/>

3. Электронно-библиотечная система «BOOK.ru» www.book.ru

4. Электронно-библиотечная система «IPRbooks» <http://www.iprbookshop.ru/>

5. ЭБС «Национальный цифровой ресурс «Руконт» <https://lib.rucont.ru/search>

6. Образовательная платформа ЮРАЙТ <http://www.urait.ru>

Электронный каталог Университета – Автоматизированная информационная библиотечная система (АИБС «МегаПро»)

12. Перечень программного обеспечения, профессиональных баз данных и информационных справочных систем, необходимых для освоения дисциплины

Лицензионное программное обеспечение, в том числе отечественного производства:

- DsktpEdu ALNG LicSARk OLV F1 1Y Acdmc Ent (MS Windows) (подписка на ПО).

Свободно распространяемое программное обеспечение, в том числе отечественного производства:

- 7-Zip (свободный файловый архиватор с высокой степенью сжатия данных) (отечественное ПО).

- FastStone Image Viewer (программа для просмотра изображений в Microsoft Windows (лицензия бесплатного программного обеспечения)).

- Foxit Reader (Бесплатное прикладное программное обеспечение для просмотра электронных документов в стандарте PDF (лицензия бесплатного программного обеспечения)).

- Indigo (система программ для создания и проведения компьютерного тестирования знаний).

- Google Chrome, ЯндексБраузер (браузер).

- Adobe Acrobat Reader DC (• ПО для просмотра, печати и комментирования документов в формате PDF)

- Visual Studio Code (интегрированная среда разработки программного обеспечения)

Профессиональные базы данных не используются.

Информационные справочные системы:

- СПС КонсультантПлюс. Компьютерная справочная правовая система, широко используется учеными, студентами и преподавателями (подписка на ПО)

Каждый обучающийся в течение всего периода обучения обеспечивается индивидуальным неограниченным доступом к электронно-библиотечной

13. Материально–техническое обеспечение дисциплины

Образовательный процесс обеспечен учебными аудиториями для проведения учебных занятий, а именно для проведения занятий лекционного типа, практических занятий, групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации, помещениями для самостоятельной работы студентов.

Учебная аудитория для проведения занятий лекционного типа, групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации.

Рабочее место преподавателя (стол, стул), рабочие места обучающихся, доска для написания мелом, кафедра.

Технические средства обучения: персональный компьютер-1,

мультимедийное оборудование (проектор, проекционный экран) - 1

Учебная аудитория для проведения, практических занятий, групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации

Рабочее место преподавателя (стол, стул), рабочие места обучающихся, доска для написания мелом, кафедра.

Технические средства обучения:

персональный компьютер-1, мультимедийное оборудование (проектор, проекционный экран) – 1.

Помещение для самостоятельной работы обучающихся, оснащенное компьютерной техникой с возможностью подключения к сети «Интернет» и обеспечением доступа в электронную информационно-образовательную среду института

Рабочие места обучающихся.

Технические средства обучения: персональный компьютер - 13.

Учебные аудитории соответствуют действующим противопожарным правилам и нормам, укомплектованы учебной мебелью.

Обновление рабочей программы дисциплины (модуля)

Наименование раздела рабочей программы, в который внесены изменения

(измененное содержание раздела)

обсуждено и рекомендовано к утверждению решением кафедры
наименование кафедры

от ____ ____ 20__ г., протокол № ____

одобрено Научно-методическим советом ____ от ____ ____ 20__ г.,
протокол № ____