

Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Смирнов Сергей Николаевич
Должность: врио ректора
Дата подписания: 13.10.2023 15:36:10
Уникальный программный ключ:
69e375c64f7e975d4e8830e7b4fcc2ad1b959f08

Министерство науки и высшего образования Российской Федерации
ФГБОУ ВО «Тверской государственный университет»



Утверждаю:

Руководитель ООП:

 Н.А. Семькина

« 9 » 06 2023 г.

Рабочая программа дисциплины (с аннотацией)

Методы программирования

Специальность

10.05.01 Компьютерная безопасность

Специализация

«Математические методы защиты информации»

Для студентов 3 и 4 курсов дневной формы обучения

Составитель:



к. ф.-м. н., доцент Пирулева В. М.

Тверь 2023

I. Аннотация

1. Наименование дисциплины (модуля) в соответствии с учебным планом

Методы программирования

2. Цель и задачи дисциплины (модуля)

Целью освоения дисциплины (модуля) является:

теоретическая и практическая подготовка специалистов к деятельности, связанной с применением технологий программирования и с анализом вычислительной сложности алгоритмов для обеспечения информационной безопасности.

Задачами освоения дисциплины (модуля) являются:

изучение основных подходов к организации процесса разработки программного обеспечения;

изучение базовых структур данных;

изучение основных алгоритмов сортировки и поиска;

изучение основных алгоритмов поиска подстрок;

изучение алгоритмов генерации перестановок, подмножеств и псевдослучайных последовательностей;

изучение основных алгоритмов на графах;

изучение основных методов оценки вычислительной сложности алгоритмов.

3. Место дисциплины (модуля) в структуре ООП

Дисциплина входит в базовую часть ООП.

Освоение её базируется на знаниях, умениях и навыках, сформированных в процессе изучения дисциплин:

«Информатика» – работа с программными средствами общего назначения;

«Языки программирования» – знание одного из языков программирования высокого уровня;

«Математический анализ» – знание основных положений теории пределов функций, теории числовых и функциональных рядов;

«Теория вероятностей и математическая статистика» – основные понятия, виды распределений;

«Дискретная математика» – основные понятия и методы дискретной математики, включая дискретные функции, конечные автоматы, комбинаторный анализ и теорию графов.

Дисциплина "Методы программирования" является предшествующей для следующих базовых дисциплин: «Сети и системы передачи информации», «Криптографические методы защиты информации», «Компьютерные сети», «Системы управления базами данных», «Основы построения защищённых операционных систем», «Основы построения защищённых СУБД», «Основы построения защищённых сетей», «Защита программ и данных». Знания и практические навыки, полученные при изучении дисциплины «Методы программирования», используются студентами при разработке курсовых и дипломных работ.

4. Объем дисциплины (модуля):

11 зачетных единиц, 396 академических часа, в том числе

контактная работа: 185 часов, **самостоятельная работа:** 211 часов.

5. Перечень планируемых результатов обучения по дисциплине (модулю), соотнесенных с планируемыми результатами освоения образовательной программы

| Планируемые результаты освоения образовательной программы (формируемые компетенции) | Планируемые результаты обучения по дисциплине (модулю) |
|---|--|
| Базовый ОПК-10. способность к самостоятельному построению алгоритма, проведению его анализа и реализации в современных программных комплексах | Владеть: навыками разработки алгоритмов решения типовых профессиональных задач. Уметь: разрабатывать программы реализации в информационно-аналитических системах алгоритмов решения типовых задач обработки информации, проводить оценку сложности алгоритмов; разрабатывать эффективные алгоритмы и программы. Знать: современные технологии программирования, базовые структуры данных. |
| Продвинутый | Владеть: навыками документирования программного обеспечения. Уметь: строить алгоритмы решения типовых задач анализа информации в ИАС и создавать программы их реализации, планировать разработку сложного программного обеспечения; оценивать качество готового программного обеспечения. Знать: показатели качества программного обеспечения, основные комбинаторные и теоретико-графовые алгоритмы, а также способы их эффективной реализации и оценки сложности. |

6. Форма промежуточной аттестации

Зачет в 6 семестре.

Экзамен в 7 семестре.

7. Язык преподавания русский.

II. Содержание дисциплины и структура учебных видов деятельности

| №п /п | Наименование раздела и темы дисциплины | ВСЕГО (час) | Контактная работа (час) | | Самост оятельн ая работа (час) |
|----------|---|----------------|----------------------------|-----------------------------|--|
| | | | лекции | лаборатор ные занятия | |
| 1. | Жизненный цикл программного обеспечения. | 8 | 2 | | 6 |
| 2. | Качество программных систем | 8 | 2 | | 6 |
| 3. | Анализ и разработка требований | 10 | 2 | | 8 |
| 4. | Проектирование архитектуры программных систем | 20 | 4 | 8 | 8 |
| 5. | Аттестация и верификация | 20 | 4 | 8 | 8 |
| 6. | Управление проектами | 12 | 4 | | 8 |
| 7. | Базовые структуры данных | 26 | 4 | 11 | 11 |
| 8. | Понятие алгоритма. Оценки сложности работы алгоритмов | 20 | 4 | 8 | 8 |
| 9. | Деревья | 34 | 6 | 14 | 14 |
| 10. | Алгоритмы поиска | 22 | 6 | 8 | 8 |
| | ИТОГО в 6 семестре | 180 | 38 | 57 | 85 |
| 11. | Алгоритмы сортировки. Внутренние сортировки | 48 | 8 | 12 | 28 |
| 12 | Алгоритмы сортировки. Внешние сортировки | 24 | 4 | 6 | 14 |
| 13 | Алгоритмы поиска. | 24 | 4 | 6 | 14 |

| №п/п | Наименование раздела и темы дисциплины | ВСЕГО (час) | Контактная работа (час) | | Самостоятельная работа (час) |
|------|---|-------------|-------------------------|----------------------|------------------------------|
| | | | лекции | лабораторные занятия | |
| | Хеширование | | | | |
| 14 | Алгоритмы поиска подстроки | 24 | 4 | 6 | 14 |
| 15 | Алгоритмы на графах | 60 | 10 | 15 | 35 |
| 16 | Алгоритмы генерации перестановок, подмножеств, сочетаний и разбиений. | 24 | 4 | 6 | 14 |
| 17 | Генерация псевдослучайных чисел | 12 | 2 | 3 | 7 |
| | ИТОГО в 7 семестре | 216 | 36 | 54 | 126 |
| | ИТОГО | 396 | 74 | 111 | 211 |

Учебная программа

Раздел 1. Технологии программирования

Тема №1. Жизненный цикл программного обеспечения. Программные системы. Основные проблемы разработки сложных программных систем. Жизненный цикл ПО. Стандартизация жизненного цикла в системе государственных стандартов ЕСПД и ГОСТ Р ИСО/МЭК 12207. Модели жизненного цикла (каскадная, спиральная, формальные и др.). Гибкие методологии разработки (XP, Scrum и др.).

Тема №2. Качество программных систем. Критерии оценки качества программных систем, характеристики качества и метрики качества. Методы контроля качества. Модель процесса оценивания качества. ГОСТ Р ИСО 9000 и ГОСТ Р ИСО 9126. Инструментальные системы оценки качества программных систем.

Тема №3. Анализ и разработка требований. Функциональные и нефункциональные требования. Разработка требований. Спецификация требований к программному обеспечению, техническое задание (ГОСТ 19.201-78, ГОСТ 34.602-89). Методы первичного сбора требований. Модели системы. Языки спецификации требований (сценарии, PDL, потоковые диаграммы и др.). Оценка требований (ГОСТ Р ИСО/МЭК 12207). Методы аттестации требований (обзор, протипирование, тестовые сценарии). Управление требованиями. Инструментальные средства отслеживания и контроля требований.

Тема №4. Проектирование архитектуры программных систем. Архитектурное и детальное проектирование. Основные принципы проектирования. Сквозная функциональность. Сцепление и связность. Архитектурные стили (парадигмы: объектно-ориентированная, компонентная, многослойная, DDD, клиент-сервер, многоуровневая, SOA и др.). Принципы объектно-ориентированного проектирования (SOLID). Паттерны проектирования. Унифицированный язык моделирования (UML). Кодогенерация. Документирование архитектуры.

Тема №5. Аттестация и верификация. Понятие аттестации и верификации. Методы инспектирования. Инструментальные средства инспектирования и оценки качества. Тестирование ПО. Неразрешимость проблемы тестирования. Виды и уровни тестирования. Стратегии восходящего и нисходящего тестирования. Методы "белого" и "черного" ящика. Автоматизированное и ручное тестирование. Разработка через тестирование (TDD). Непрерывная интеграция. Покрытие кода тестами. Методы отладки программ. Инструментальные средства поддержки тестирования и отладки.

Тема №6. Управление проектами. Группы процессов и процедуры управления проектом. План управления проектом. Инструментальные средства управления проектами. Управление персоналом. Права и обязанности членов коллектива. Мотивация. Организация совместной работы коллектива. Управление рисками. Модель зрелости возможностей.

Инструментальные средства планирования и управления процессом разработки.

Раздел 2. Структуры данных и прикладные алгоритмы

Тема № 7. Базовые структуры данных. Базовые структуры данных. Стек, очередь, список, поиск в последовательно организованных структурах. Потоки, разреженные матрицы.

Тема № 8. Понятие алгоритма. Оценки сложности работы алгоритмов. Определения алгоритма, рекурсивные алгоритмы, оценка сложности алгоритмов.

Тема № 9. Деревья. Деревья, двоичные деревья поиска, монотонные деревья, сбалансированные деревья: AVL- деревья, красно-черные деревья.

Тема № 10. Алгоритмы поиска. Основные понятия. Поиск в последовательно организованных структурах. Поиск в древовидных структурах. деревья двоичного поиска, монотонные деревья, сбалансированные деревья, деревья цифрового поиска.

Тема № 11. Алгоритмы сортировки. Внутренние сортировки. Классификация алгоритмов сортировки. Алгоритмы внутренней сортировки: Сортировки сравнениями: вставками, выбором, слиянием и обменами. Распределяющие сортировки. Оценка сложности работы алгоритмов внутренней сортировки.

Тема № 12. Алгоритмы сортировки. Внешние сортировки. Внешняя сортировка. Построение цепочек. Многофазное и каскадное слияния.

Тема № 13. Алгоритмы поиска. Хеширование. Поиск в последовательно организованных структурах, поиск в массивах. Хеширование и разрешение коллизий. Алгоритмы исчерпывающего поиска.

Тема № 14. Алгоритмы поиска подстрок. Алгоритм простого поиска подстрок. Алгоритм Рабина-Карпа. Алгоритм Кнута-Морриса-Пратта. Алгоритм поиска подстрок с помощью построения конечного автомата. Алгоритм Бойера- Мура.

Тема № 15 Алгоритмы на графах. Способы задания графов. Алгоритмы обхода графов в ширину и глубину. Алгоритмы нахождения сильно связанных и двусвязных компонент графа. Остовные деревья минимальной стоимости и алгоритмы их построения. Алгоритмы нахождения кратчайших расстояний между вершинами. Алгоритм Беллмана-Форда. Потoki в сетях.

Тема № 16. Алгоритмы генерации перестановок, подмножеств, сочетаний и разбиений. Порождение перестановок в лексикографическом порядке, с помощью векторов инверсий, вложенных циклов и в порядке минимального изменения. Алгоритмы порождения подмножеств. Коды Грея. Композиции и разбиения целых чисел и алгоритмы их порождения. Оценка вычислительной сложности алгоритмов.

Тема № 17. Генерация псевдослучайных последовательностей.

III. Перечень учебно-методического обеспечения для самостоятельной работы обучающихся по дисциплине (или модулю)

Самостоятельная работа обучающихся направлена на освоение учебного материала и развитие практических умений. Самостоятельная работа включает следующие виды самостоятельной работы студентов

- работа с рекомендованной учебной литературой;
- выполнение домашних заданий;
- подготовка реферата и электронной презентации по одной из тем первого раздела;
- выполнение заданий по лабораторным работам;
- подготовка к зачёту;
- подготовка к экзамену.

1.Методические рекомендации по организации изучения дисциплины

Цель обучения достигается сочетанием применения традиционных и инновационных педагогических технологий.

При проведении лекционных занятий широко применяется такая форма, как лекция-визуализация, сопровождающая изложение теоретического материала презентациями.

Основной упор в методике проведения лабораторных занятий сделан на отработку и закреплении учебного материала в процессе выполнения заданий с вычислительной техники в компьютерном классе. Особое внимание при этом уделено применению элементов проблемного обучения, опережающей самостоятельной работе студентов.

Во втором семестре изучения дисциплины студенты должны выполнить домашнее задание, способствующее приобретению навыков ведения научно-исследовательской деятельности в области анализа вычислительной сложности алгоритмов.

Текущий контроль усвоения знаний осуществляется путем подготовки и сдачи отчетов по итогам выполнения лабораторных работ, проверки выполнения домашнего задания, опросов на лекционных занятиях.

При изучении дисциплины в каждом разделе дисциплины выделяются наиболее важные моменты:

в 6 семестре акцентируется внимание на особенностях использования различных структур данных в зависимости от свойств информации, способах ее кодирования, передачи и обработки;

во 7 семестре при изучении алгоритмов сортировок приводится классификация алгоритмов сортировки и больше внимания уделяется сравнительному анализу сложности их работы;

при изучении алгоритмов поиска показываются возможные подходы к классификации алгоритмов поиска, заостряется внимание на необходимости комплексного подхода к выбору структуры данных при реализации конкретного алгоритма;

при изучении алгоритмов на графах подробнее рассматриваются элементарные понятия теории графов и сосредотачивается внимание на широком применении алгоритмов на графах в науке и технике;

при изучении комбинаторных алгоритмов подчеркивается, какую важную роль играют в вопросах защиты информации алгоритмы, порождающие псевдослучайные последовательности, перестановки на множестве элементов и коды Грея.

2. Перечень тем лабораторных занятий

1. Стеки, деки, очереди, линейные списки.
2. Двухнаправленные списки.
3. Потоки.
4. Разреженные матрицы.
5. Деревья. Двоичные деревья поиска.
6. Алгоритмы поиска в деревьях.
7. AVL-деревья.
8. Красно-черные деревья.
9. Алгоритмы сортировки сравнениями. Вывод нижней оценки для трудоемкости работы алгоритмов данного типа. Сравнение эмпирических оценок вычислительной сложности работы алгоритма с теоретическими.
10. Алгоритм сортировки вставками и вывод оценки его трудоемкости. Сортировка Шелла. Сравнение эмпирических оценок вычислительной сложности работы алгоритма с теоретическими.
11. Обменная сортировка. Сортировка расческой. Сравнение эмпирических оценок вычислительной сложности работы алгоритма с теоретическими.
12. Алгоритм быстрой сортировки. Сравнение эмпирических оценок вычислительной сложности работы алгоритма с теоретическими.
13. Сортировка выбором. Пирамидальная сортировка. Сравнение эмпирических оценок вычислительной сложности работы алгоритма с теоретическими.
14. Сортировки слиянием, блочная и распределяющие сортировки. Сравнение эмпирических оценок вычислительной сложности работы алгоритма с теоретическими.

15. Целочисленные сортировки. Сортировка подсчетом и поразрядная сортировка. Сравнение эмпирических оценок вычислительной сложности работы алгоритма с теоретическими.
16. Внешние сортировки слиянием.
17. Поиск в массивах. Алгоритмы исчерпывающего поиска. Бинарный, интерполяционный и следящий поиск.
18. Поиск в тексте. Алгоритмы Кнута - Морриса - Пратта и Боуэра - Мура.
19. Хеширование и способы разрешения коллизий.
20. Способы задания графа. Остовное дерево. Алгоритмы поиска в глубину и в ширину.
21. Поиск кратчайшего пути в графе. Алгоритмы Дейкстры и Флойда.
22. Алгоритм построения минимального остовного дерева. Алгоритмы Прима и Крускала.
23. Нахождение связных и двусвязных компонент графа.
24. Алгоритм поиска сильно связных компонент графа.
25. Генерация псевдо - случайных чисел. Метод середины квадрата, линейный и квадратичный конгруэнтные методы, метод Фибоначчи
26. Алгоритмы порождения перестановок в лексикографическом порядке и порядке минимального изменения.
27. Построение кода Грея. Рекуррентный и не рекуррентный алгоритмы.

3. Перечень тем индивидуальных заданий

1. Сложение деревьев поиска.
2. Обменная сортировка со слиянием, алгоритм Бэтчера.
3. Лексикографическая сортировка.
4. Турнирная сортировка.
5. Алгоритмы поиска с возвращениями.
6. Индексно-последовательный поиск.
7. Алгоритм поиска подстрок с помощью построения конечного автомата.
8. Алгоритм Рабина-Карпа поиска подстрок.
9. Построение оптимальных бинарных деревьев поиска.

- 10.Балансировка деревьев по весу.
- 11.Цифровой поиск.
- 12.Алгоритм транзитивного замыкания.
- 13.Композиции и разбиения целых чисел и алгоритмы их порождения.
- 14.Алгоритм Беллмана-Форда нахождения кратчайших расстояний между вершинами.
- 15.Потоки в сетях.
- 16.Алгоритм Эдмондса-Карпа поиска максимального потока в сети.

4. Примерный перечень тем домашних заданий

1. Реализация структур данных стек, очередь, двоичное дерево поиска.
2. Напишите программу, реализующую один из алгоритмов поиска.
3. Реализация алгоритмов турнирной и пирамидальной сортировок. Сравнение эмпирических оценок вычислительной сложности работы алгоритмов.
4. Реализация быстрой сортировки, с учётом возможных модификаций. Сравнение эмпирических оценок вычислительной сложности работы алгоритма с теоретическими.
5. Реализация алгоритма лексикографической сортировки.
6. Реализация алгоритма построения оптимального двоичного дерева поиска.
7. Реализация алгоритма балансировки AVL-дерева при вставке элементов.
8. Реализация алгоритма балансировки красно-чёрного дерева при вставке элементов.
9. Напишите программу, реализующую один из алгоритмов сортировки.
- 10.Получить эмпирические оценки трудоемкости алгоритма сортировки или поиска.
- 11.Произвести сравнительный анализ трудоемкости работы нескольких алгоритмов.

12. Напишите программу, реализующую алгоритм поиска сильно связанных компонент в графе.
13. Напишите программу, реализующую алгоритм поиска двусвязных компонент в графе.
14. Напишите программу, реализующую алгоритм порождения перестановок в лексикографическом порядке.
15. Напишите программу, реализующую алгоритм порождения перестановок циклическим сдвигом.
16. Реализация алгоритма построения DST и Trie деревьев.
17. Реализация алгоритма идеального хеширования.
18. Сравнительный анализ вычислительной трудоёмкости алгоритмов Кнута – Морриса – Пратта и Боуэра – Мура.
19. Реализация алгоритма нахождения максимального потока в сети.
20. Реализация алгоритма генерации кодов Грея.

5. Темы рефератов по разделу Технологии программирования

Тема 1. Жизненный цикл программного обеспечения.

1. Программные системы. Основные проблемы разработки сложных программных систем. Жизненный цикл ПО. Стандартизация жизненного цикла в системе государственных стандартов ЕСПД и ГОСТ Р ИСО/МЭК 12207.
2. Модели жизненного цикла (каскадная, спиральная, формальные и др.). Гибкие методологии разработки (XP, Scrum и др.).

Тема 2. Качество программных систем

3. Критерии оценки качества программных систем, характеристики качества и метрики качества. Методы контроля качества.
4. Модель процесса оценивания качества. ГОСТ Р ИСО 9000 и ГОСТ Р ИСО 9126. Инструментальные системы оценки качества программных систем.

Тема 3. Анализ и разработка требований

5. Функциональные и нефункциональные требования. Разработка требований. Спецификация требований к программному обеспечению, техническое задание (ГОСТ 19.201-78, ГОСТ 34.602-89).
6. Методы первичного сбора требований. Модели системы. Языки спецификации требований (сценарии, PDL, потоковые диаграммы и др.). Оценка требований (ГОСТ Р ИСО/МЭК 12207).
7. Методы аттестации требований (обзор, протипирование, тестовые сценарии). Управление требованиями. Инструментальные средства отслеживания и контроля требований.

Тема 4. Проектирование архитектуры программных систем

8. Архитектурное и детальное проектирование. Основные принципы проектирования. Сквозная функциональность. Сцепление и связность.
9. Архитектурные стили (парадигмы: объектно-ориентированная, компонентная, многослойная, DDD, клиент-сервер, многоуровневая, SOA и др.).
10. Принципы объектно-ориентированного проектирования (SOLID).
11. Паттерны проектирования.
12. Унифицированный язык моделирования (UML).
13. Унифицированный язык моделирования (UML).
14. Кодогенерация. Документирование архитектуры.

Тема 5. Аттестация и верификация

15. Понятие аттестации и верификации. Методы инспектирования. Инструментальные средства инспектирования и оценки качества.
16. Тестирование ПО. Неразрешимость проблемы тестирования. Виды и уровни тестирования. Стратегии восходящего и нисходящего тестирования. Методы "белого" и "черного" ящика. Автоматизированное и ручное тестирование. Разработка через тестирование (TDD). Непрерывная интеграция. Покрытие кода тестами.

17. Методы отладки программ. Инструментальные средства поддержки тестирования и отладки.

Тема 6. Управление проектами

18. Группы процессов и процедуры управления проектом. План управления проектом. Инструментальные средства управления проектами.

19. Управление персоналом. Права и обязанности членов коллектива. Мотивация. Организация совместной работы коллектива.

20. Управление рисками. Модель зрелости возможностей. Инструментальные средства планирования и управления процессом разработки

Литература

а) обязательная литература:

1. Методы программирования : учебное пособие / Ю. Ю. Громов, О. Г. Иванова, Ю. В. Кулаков [и др.]. — Тамбов : Тамбовский государственный технический университет, ЭБС АСВ, 2012. — 144 с. — ISBN 978-5-8265-1076-6. — Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. — URL: <https://www.iprbookshop.ru/63867.html>

2. Ковалевская, Е. В. Методы программирования : учебное пособие / Е. В. Ковалевская, Н. В. Комлева. — Москва : Евразийский открытый институт, 2011. — 320 с. — ISBN 978-5-374-00356-7. — Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. — URL: <https://www.iprbookshop.ru/10784.html>

б) дополнительная литература:

1. Макарова, Н. П. Методы программирования и информатика : лабораторный практикум : в 2 частях : [16+] / Н. П. Макарова. — Гродно : Гродненский Государственный университет имени Янки Купалы, 2012. — Часть 1. — 54 с. — Режим доступа: по подписке. — URL: <https://biblioclub.ru/index.php?page=book&id=134239>

IV. Фонды оценочных средств

1. Типовые контрольные задания для проверки уровня сформированности компетенции ОПК-10 «Способность к самостоятельному построению алгоритма, проведению его анализа и реализации в современных программных комплексах»

| Этап формирования компетенции, в котором участвует дисциплина | Типовые контрольные задания для оценки знаний, умений, навыков (2-3 примера) | Показатели и критерии оценивания компетенции, шкала оценивания |
|---|--|--|
| Базовый владеть | Бинарные деревья поиска. Основные операции над ними. | <p>Владеет навыками разработки алгоритмов решения типовых профессиональных задач – 85-100%.</p> <p>Владеет навыками разработки алгоритмов решения типовых профессиональных задач, но показывает недостаточность опыта в их реализации – 65-84%.</p> <p>Знает основные алгоритмы решения типовых задач, но не все может реализовать – 45-64%.</p> <p>Имеет лишь представление о бинарных деревьях и основных операциях над ними – 20-44%.</p> <p>Не способен – 0-19%.</p> |
| Базовый уметь | Обменные сортировки. Шейкерная сортировка. Их вычислительная сложность | <p>Умеет разрабатывать программы реализации в информационно-аналитических системах алгоритмов решения типовых задач обработки информации, проводить оценку сложности алгоритмов; разрабатывать эффективные алгоритмы и программы – 85-100%.</p> <p>Умеет разрабатывать программы реализации в информационно-</p> |

| | | |
|---------------------------------|---|---|
| | | <p>аналитических системах алгоритмов решения типовых задач обработки информации, но не умеет проводить оценку сложности алгоритмов – 65-84%.</p> <p>Умеет разрабатывать программы реализации в информационно-аналитических системах алгоритмов решения типовых задач обработки информации, но не эффективно и не умеет проводить оценку сложности алгоритмов – 65-84%.</p> <p>Знает основные алгоритмы решения типовых задач, но не все может реализовать программно – 45-64%.</p> <p>Имеет лишь представление об алгоритмах сортировок – 20-44%.</p> <p>Не способен – 0-19%.</p> |
| <p>Базовый знать</p> | <p>Стеки и очереди. Основные операции над ними.</p> | <p>Знает современные технологии программирования, в частности, объектно-ориентированного программирования, базовые структуры данных – 85-100%.</p> <p>Знает базовые структуры данных, но не достаточно хорошо ориентируется в технологиях программирования – 65-84%.</p> <p>Знает базовые</p> |

| | | |
|-----------------------------------|--|---|
| | | <p>структуры данных, но плохо ориентируется в технологиях программирования – 45-64%</p> <p>Имеет лишь представление о базовых структурах данных – 20-44%</p> <p>Не способен – 0-19%</p> |
| <p>Продвинутый владеть</p> | <p>Опишите основные разделы, которые должна содержать проектная документация программного обеспечения, и содержание этих разделов</p> | <p>Владеет навыками документирования программного обеспечения (ПО) – 85-100%.</p> <p>Владеет принципами составления документации к ПО, но не достаточно хорошо ориентируется в содержании разделов – 65-84%.</p> <p>Владеет принципами составления документации к ПО, но плохо ориентируется в содержании разделов – 45-64%</p> <p>Имеет лишь представление о документировании ПО – 20-44%</p> <p>Не способен – 0-19%</p> |
| <p>Продвинутый уметь</p> | <p>Сформулируйте основные принципы объектно-ориентированного проектирования (ООП) классов и поясните суть каждого из этих принципов.</p> | <p>Умеет строить алгоритмы решения типовых задач анализа информации в ИАС и создавать программы их реализации, планировать разработку сложного программного обеспечения; оценивать качество готового</p> |

| | | |
|-------------------------------------|---|--|
| | | <p>программного обеспечения – 85-100%. Владеет принципами ООП, но показывает недостаточность опыта их использования – 65-84%</p> <p>Не достаточно владеет принципами ООП, плохо разбирается в их сути – 45-64%</p> <p>Имеет лишь представление о принципах ООП – 20-44%</p> <p>Не способен – 0-19%</p> |
| <p>Продвинутый знать</p> | <p>Алгоритм Дейкстры нахождения кратчайших расстояний от источника до всех остальных вершин графа</p> | <p>Знает показатели качества программного обеспечения, основные комбинаторные и теоретико-графовые алгоритмы, а также способы их эффективной реализации и оценки сложности – 85-100%.</p> <p>Знает алгоритм Дейкстры, но не достаточно хорошо ориентируется в способах его эффективной реализации и оценке сложности – 65-84%.</p> <p>Знает алгоритм Дейкстры, но плохо ориентируется в его реализации и оценке сложности – 45-64%</p> <p>Имеет лишь представление об алгоритме Дейкстры – 20-44%</p> <p>Не способен</p> |

2. Текущий контроль успеваемости

Задания для самостоятельной работы

1. Стеки, деки, очереди, линейные списки.
2. Двухнаправленные списки.
3. Потoki.
4. Разреженные матрицы.
5. Деревья. Двоичные деревья поиска.
6. Алгоритмы поиска в деревьях.
7. AVL-деревья.
8. Красно-черные деревья.
9. Алгоритмы сортировки сравнениями. Вывод нижней оценки для трудоемкости работы алгоритмов данного типа. Сравнение эмпирических оценок вычислительной сложности работы алгоритма с теоретическими.
10. Алгоритм сортировки вставками и вывод оценки его трудоемкости. Сортировка Шелла. Сравнение эмпирических оценок вычислительной сложности работы алгоритма с теоретическими.
11. Обменная сортировка. Сортировка расческой. Сравнение эмпирических оценок вычислительной сложности работы алгоритма с теоретическими.
12. Алгоритм быстрой сортировки. Сравнение эмпирических оценок вычислительной сложности работы алгоритма с теоретическими.
13. Сортировка выбором. Пирамидальная сортировка. Сравнение эмпирических оценок вычислительной сложности работы алгоритма с теоретическими.
14. Сортировки слиянием, блочная и распределяющие сортировки. Сравнение эмпирических оценок вычислительной сложности работы алгоритма с теоретическими.

15. Целочисленные сортировки. Сортировка подсчетом и поразрядная сортировки. Сравнение эмпирических оценок вычислительной сложности работы алгоритма с теоретическими.
16. Внешние сортировки слиянием.
17. Поиск в массивах. Алгоритмы исчерпывающего поиска. Бинарный, интерполяционный и следящий поиск.
18. Поиск в тексте. Алгоритмы Кнута - Морриса - Пратта и Боуэра - Мура.
19. Хеширование и способы разрешения коллизий.
20. Способы задания графа. Остовное дерево. Алгоритмы поиска в глубину и в ширину.
21. Поиск кратчайшего пути в графе. Алгоритмы Дейкстры и Флойда.
22. Алгоритм построения минимального остовного дерева. Алгоритмы Прима и Крускала.
23. Нахождение связных и двусвязных компонент графа.
24. Алгоритм поиска сильно связных компонент графа.
25. Генерация псевдо - случайных чисел. Метод середины квадрата, линейный и квадратичный конгруэнтные методы, метод Фибоначчи
26. Алгоритмы порождения перестановок в лексикографическом порядке и порядке минимального изменения.
27. Построение кода Грея. Рекуррентный и не рекуррентный алгоритмы.

3. Промежуточная аттестация

Вопросы к зачету

1. Стеки, деки, очереди, линейные списки.
2. Двухнаправленные списки.
3. Потоки.
4. Разреженные матрицы.
5. Деревья. Двоичные деревья поиска.
6. Алгоритмы поиска в деревьях.

7. AVL-деревья.
8. Красно-черные деревья.
9. Программное обеспечение, стадии жизненного цикла: понятие жизненного цикла.
10. Программное обеспечение, стадии жизненного цикла: водопадная (каскадная) модель.
11. Программное обеспечение, стадии жизненного цикла: итерационная модель.
12. Программное обеспечение, стадии жизненного цикла: итерационный подход, прототипирование.
13. Анализ требований: техническое задание и спецификация требований, три подхода к разработке ТЗ.
14. Анализ требований: свойства требований. 7. Анализ требований: способы выражения (записи) требований в ТЗ.
15. Анализ требований: требования заказчика и детальные требования.
16. Анализ требований: способы организации детальных требований.
17. Характеристики качества ПО: удобство применения, универсальность.
18. Характеристики качества ПО: надежность, эффективность.
19. Характеристики качества ПО: сопровождаемость, корректность.
20. Обеспечение качества ПО: обеспечение завершенности и надежности.
21. Обеспечение качества ПО: обеспечение эффективности и легкости применения.
22. Обеспечение качества ПО: обеспечение универсальности.
23. Проектирование: декомпозиция, классификация архитектур.
24. Проектирование: архитектура потоков данных и независимые компоненты.
25. Проектирование: декомпозиция, классификация архитектур.

Вопросы к экзамену

1. Определение сложности алгоритма. Общие функции оценки сложности. Правила вычисления сложности алгоритма. Анализ скорости выполнения алгоритмов.
2. Списки, их представление, операции со списками. Списки с потоками.
3. Стеки и очереди. Основные операции. Очереди с приоритетами. Основные операции.
4. Разреженные матрицы, способы представления. Основные операции.
5. Деревья. Полные деревья. Обход бинарного дерева: прямой, симметричный, обратный, в ширину.
6. Бинарные деревья поиска. Основные операции. Алгоритм сортировки с помощью дерева.
7. AVL-деревья. Балансировка деревьев по высоте.
8. Красно-черные деревья. Балансировка деревьев по высоте.
9. Сортировки вставками. Сортировка Шелла. Их вычислительная сложность.
10. Обменные сортировки. Шейкерная сортировка. Сортировка прочесыванием. Их вычислительная сложность.
11. Сортировка выбором. Пирамидальная сортировка. Их вычислительная сложность
12. Сортировки, не основанные на сравнениях. Блочная сортировка. Поразрядная сортировка. Сортировка подсчетом. Их вычислительная сложность.
13. Быстрая сортировка. Сортировки слиянием. Их вычислительная сложность.
14. Основные понятия внешней сортировки. Алгоритмы естественного и сбалансированного слияния.
15. Внешние сортировки. Алгоритмы многофазного и каскадного слияния.
16. Поиск в массивах: полный перебор, двоичный и интерполяционный поиск. Следящий поиск.
17. Поиск в тексте. Алгоритмы Кнута – Морриса – Пратта и Боуэра – Мура.

18. Хеш-функция. Виды хеш-функций. Хеш-таблица. Внешнее хеширование (метод цепочек). Блочное хеширование.
19. Закрытое хеширование. Способы разрешения коллизий.
20. Способы представления графов в ЭВМ. Алгоритм поиска в глубину.
Алгоритм поиска в ширину. Реализации алгоритмов.
21. Нахождения транзитивного замыкания в графе.
22. Алгоритм Дейкстры нахождения кратчайших расстояний от источника до всех остальных вершин графа.
23. Алгоритм Флойда нахождения кратчайших расстояний между всеми парами вершин графа.
24. Остовные деревья. Алгоритмы нахождения остовного дерева минимального веса Прима и Крускала.
25. Связные компоненты в неориентированных графах. Алгоритм, основанный на разделенных множествах. Двусвязные компоненты в неориентированных графах. Алгоритм, основанный на поиске точек сочленения.
26. Сильно связные компоненты в ориентированных графах. Алгоритм Косарайю.
27. Перестановки. Алгоритмы порождения перестановок в лексикографическом порядке и в порядке минимального изменения.
28. Коды Грея. Алгоритмы порождения кодов Грея: рекурсивный и не рекурсивный.
29. Источники и генераторы случайных и псевдослучайных чисел. Алгоритмы генерации псевдослучайных последовательностей. Проверка качества работы генератора.
30. Методы моделирования равномерно распределенных случайных величин. Моделирование дискретных и непрерывных случайных величин.

4. Пример одной из контрольных работ

Вариант

1. Опишите необходимые типы, переменные и функцию вставки в связанный линейный однонаправленный список (с полем данных типа integer) нового элемента с адресом P1 после элемента с адресом P (предполагается, что элемент с адресом P в списке существует). Не используйте никаких других дополнительных переменных-указателей.

2. Опишите необходимые типы, переменные и последовательность операций присваивания, формирующих связанный список с двумя потоками, упорядочивающими элементы списка по возрастанию значений поля данных поле1 и по возрастанию значений поля данных поле2 (поле1 – типа char, поле2 – типа integer). Входные данные содержат три набора (значение поля1, значение поля2, указатель на выделенный под элемент участок памяти): ('f', 5, p1), ('w', 1, p2), ('m', 10, p3). Изобразите список графически.

3. В рекурсивном алгоритме, в котором при каждом рекурсивном вызове производится $2n$ операций и параметр n (размер задачи) уменьшается на единицу, возникает следующее рекуррентное соотношение для числа выполняемых операций:

$$C(n) = C(n-1) + 2n, \text{ где } n > 1 \text{ и } C(1) = 1.$$

Постепенно раскрывая рекурсию, определите временную сложность данного алгоритма.

4. Дайте определение глубины узла и глубины дерева. Приведите пример.

5. Двоичное дерево задано как статическая структура данных в виде одномерного массива:

| 1 | 2 | 3 | 4 | 5 | 6 | X | 7 | X | X | X | 8 | 9 | X | X |, где символ X означает отсутствие значения.

Изобразите его графически: с помощью списочного представления (списков дочерних узлов) и с помощью метода полных узлов (в виде графа с выделенной вершиной – корнем).

6. Выпишите линейные последовательности, получающиеся в результате прямого, симметричного и обратного обхода дерева в задании 6, а также его обхода по ширине.

7. Даны два списка s_1 и s_2 , организованных в виде стека и упорядоченных по возрастанию значений информационных полей. Создать упорядоченный по возрастанию список в виде стека s_3 , являющийся объединением списков s_1 и s_2 , т.е. $s_3 = s_1 \cup s_2$. Напишите алгоритм или программный код.

V. Перечень основной и дополнительной учебной литературы,

необходимой для освоения дисциплины

а) обязательная литература:

1. Методы программирования : учебное пособие / Ю. Ю. Громов, О. Г. Иванова, Ю. В. Кулаков [и др.]. — Тамбов : Тамбовский государственный технический университет, ЭБС АСВ, 2012. — 144 с. — ISBN 978-5-8265-1076-6. — Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. — URL: <https://www.iprbookshop.ru/63867.html>
2. Ковалевская, Е. В. Методы программирования : учебное пособие / Е. В. Ковалевская, Н. В. Комлева. — Москва : Евразийский открытый институт, 2011. — 320 с. — ISBN 978-5-374-00356-7. — Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. — URL: <https://www.iprbookshop.ru/10784.html>

а) дополнительная литература:

1. Макарова, Н. П. Методы программирования и информатика : лабораторный практикум : в 2 частях : [16+] / Н. П. Макарова. — Гродно : Гродненский Государственный университет имени Янки Купалы, 2012. — Часть 1. — 54 с. — Режим доступа: по подписке. — URL: <https://biblioclub.ru/index.php?page=book&id=134239>

VI. Перечень ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины (или модуля)

1. ЭБС Лань <https://e.lanbook.com/> Договор № 4-е/23 от 02.08.2023 г.
2. ЭБС Znanium.com <https://znanium.com/> Договор № 1106 эбс от 02.08.2023 г.
3. ЭБС Университетская библиотека online <https://biblioclub.ru> Договор № 02-06/2023 от 02.08.2023 г.
4. ЭБС ЮРАЙТ <https://urait.ru/> Договор № 5-е/23 от 02.08.2023 г.
5. ЭБС IPR SMART <https://www.iprbookshop.ru/> Договор № 3-е/23К от 02.08.2023 г.
6. <https://cyberleninka.ru/> научная электронная библиотека «Киберленинка».
7. Научная электронная библиотека eLIBRARY.RU (подписка на журналы) https://elibrary.ru/projects/subscription/rus_titles_open.asp;
8. Репозиторий ТвГУ <http://eprints.tversu.ru>

VII. Методические указания для обучающихся по освоению дисциплины (или модуля)

Организуя свою учебную работу, студенты должны:

Во-первых, выявить рекомендуемый режим и характер учебной работы по изучению теоретического курса, практическому применению изученного материала, по выполнению заданий для самостоятельной работы, по использованию информационных технологий и т.д.

Во-вторых, ознакомиться с указанным в методическом материале по дисциплине (модулю) перечнем учебно-методических изданий, рекомендуемых студентам для подготовки к занятиям и выполнения самостоятельной работы, а также с методическими материалами на бумажных и/или электронных носителях, выпущенных кафедрой своими силами и предоставляемые студентам во время занятий.

Самостоятельная работа студентов, предусмотренная учебным планом должна соответствовать более глубокому усвоению изучаемого курса, формировать навыки исследовательской работы и ориентировать студентов на умение применять теоретические знания на практике.

1. Работа с учебными пособиями. Для полноценного усвоения курса студент должен, прежде всего, овладеть основными понятиями этой дисциплины. Необходимо усвоить определения и понятия, уметь приводить их точные формулировки, приводить примеры объектов, удовлетворяющих этому определению. Кроме того, необходимо знать круг фактов, связанных с данным понятием. Требуется также знать связи между понятиями, уметь устанавливать соотношения между классами объектов, описываемых различными понятиями.

2. Самостоятельное изучение тем. Самостоятельная работа студента является важным видом деятельности, позволяющим хорошо усвоить изучаемый предмет и одним из условий достижения необходимого качества подготовки и профессиональной переподготовки специалистов. Она предполагает самостоятельное изучение студентом рекомендованной учебно-методической литературы, различных справочных материалов, написание рефератов, выступление с докладом, подготовку к лекционным и практическим занятиям, подготовку к зачёту и экзамену.

3. Подготовка к практическим занятиям. При подготовке к практическим занятиям студентам рекомендуется следовать методическим рекомендациям по работе с учебными пособиями, приведенным выше.

4. Составление глоссария. В глоссарий должны быть включены основные понятия, которые студенты изучают в ходе самостоятельной работы. Для полноты исследования рекомендуется вписывать в глоссарий и те термины, которые студентам будут раскрыты в ходе лекционных занятий.

5. Составление конспектов. В конспекте отражены основные понятия темы. Для наглядности и удобства запоминания использованы схемы и таблицы.

6. Подготовка к зачету / экзамену. При подготовке к зачету / экзамену студенты должны использовать как самостоятельно подготовленные конспекты, так и материалы, полученные в ходе лекций. Для получения зачета по дисциплине необходимо решить минимум 51% тестовых заданий (минимальная оценка – удовлетворительно), при решении меньшего количества заданий зачет считается не сданным. Экзамен студенты могут сдавать в виде теста, контрольной работы или устного ответа по вопросам, представленным в данной программе. Для получения положительной оценки на экзамене необходимо продемонстрировать знания, не ниже базового (минимального) уровня.

Процедура оценивания знаний, умений, владений (умений применять) и (или) опыта деятельности обучающихся по дисциплине (модулю) производится в рамках балльно-рейтинговой системы, включая рубежную и текущую аттестации.

Согласно подходам балльно-рейтинговой системы в рамках оценки знаний, умений, владений (умений применять) и (или) опыта деятельности дисциплины (модуля) установлены следующие аспекты:

- Содержание учебной дисциплины в рамках одного семестра делится на два модуля (периода обучения). По окончании модуля (периода обучения) осуществляется рейтинговый контроль успеваемости знаний студентов.

- Сроки проведения рейтингового контроля:

осенний семестр – I рейтинговый контроль успеваемости проводится на 9-10 учебной неделе по графику учебного процесса, II рейтинговый контроль успеваемости - две последние недели фактического завершения семестра по графику учебного процесса;

весенний семестр – I рейтинговый контроль успеваемости проводится на 32-33 учебной неделе по графику учебного процесса, II рейтинговый контроль успеваемости - две последние недели фактического завершения семестра по графику учебного процесса.

Максимальное количество баллов, которое может быть получено в результате освоения дисциплины составляет 100 баллов, из них 60 баллов отводится на текущий контроль (например, по 30 баллов на каждый модуль) и 40 баллов на промежуточную аттестацию.

Максимальная сумма рейтинговых баллов по учебной дисциплине, заканчивающейся *зачетом /экзаменом*, по итогам **промежуточной аттестации** в форме теста составляет 40 баллов, при этом начисление баллов производится следующим образом:

Самостоятельно выполнено верно 85 - 100 % заданий – 40 баллов;

Самостоятельно выполнено верно 75 - 84% заданий – 30 баллов;

Самостоятельно выполнено верно 50 - 74% заданий – 20 баллов;

Выполнено верно менее 50% заданий – 0 баллов.

VIII. Перечень педагогических и информационных технологий, используемых при осуществлении образовательного процесса по дисциплине

Преподавание учебной дисциплины строится на сочетании лекций, лабораторных занятий и различных форм самостоятельной работы студентов.

В процессе освоения дисциплины используются следующие образовательные технологии, способы и методы формирования компетенций.

Традиционные лекции и лабораторные занятия, выполнение расчетно-графических работ и семестровых заданий, упражнения, моделирование,

составление различных видов алгоритмов и таблиц, программирование на языках высокого уровня.

Также на занятиях практикуется самостоятельная работа студентов, выполнение заданий в малых группах, письменные работы, моделирование дискуссионных ситуаций, работа с раздаточным материалом, привлекаются ресурсы сети INTERNET. Курс предусматривает выполнение контрольных и самостоятельных работ, письменных домашних заданий. В качестве форм контроля используются различные варианты взаимопроверки и взаимоконтроля.

Программное обеспечение:

| | |
|---|--|
| Adobe Acrobat Reader DC - Russian | бесплатно Государственный контракт на поставку лицензионных программных продуктов 103 - ГК/09 от 15.06.2009 |
| Cadence SPB/OrCAD 16.6 | бесплатно |
| Google Chrome | бесплатно |
| Java SE Development Kit 8 Update 45 (64-bit) | бесплатно |
| Kaspersky Endpoint Security 10 для Windows | Акт на передачу прав ПК545 от 16.12.2022 |
| Lazarus 1.4.0 | бесплатно Акт предоставления прав ИС00000027 от 16.09.2011; |
| Mathcad 15 M010 | Акт предоставления прав № Us000311 от 25.09.2012; |
| MATLAB R2012b | бесплатно |
| Mercurial 3.7.3 | бесплатно |
| Microsoft SQL Server 2012 Express LocalDB | бесплатно |
| Microsoft Web Deploy 3.5 | бесплатно |
| MiKTeX 2.9 | бесплатно |
| MSXML 4.0 SP2 Parser and SDK | бесплатно |
| MySQL Workbench 6.3 CE | бесплатно |
| NetBeans IDE 8.0.2 | бесплатно |
| Notepad++ | бесплатно договор №13918/M41 от 24.09.2009 с ЗАО «СофтЛайн Трейд»; |
| Origin 8.1 Sr2 | бесплатно |
| Python 3.4.3 | бесплатно |
| WinDjView 2.1 | бесплатно |
| WCF RIA Services V1.0 SP2 | бесплатно |
| Многофункциональный редактор ONLYOFFICE бесплатное ПО | бесплатно |
| ОС Linux Ubuntu бесплатное ПО | бесплатно |

IX. Материально-техническое обеспечение дисциплины

Учебная аудитория с мультимедийной установкой (Ноутбук, проектор, колонки), наличие классной доски.. Класс ПЭВМ.

Х. Сведения об обновлении рабочей программы дисциплины

| №п.п. | Обновленный раздел рабочей программы дисциплины (модуля) | Описание внесенных изменений | Дата и № протокола заседания кафедры, утвердившего изменения |
|--------------|---|--|---|
| 1. | Вся рабочая программа | Приведена в соответствие с новым стандартом и новым шаблоном | |
| 2. | | | |