


**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ**
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Воронежский государственный технический университет»

УТВЕРЖДАЮ
Декан факультета ФИТКБ

/Гусев П.Ю./
28.02.2023 г.



РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ
«Технологии и методы программирования»

Специальность 10.05.03 Информационная безопасность
автоматизированных систем

Специализация специализация N 7 "Анализ безопасности информационных
систем"

Квалификация выпускника специалист по защите информации

Нормативный период обучения 5 лет и 6 м.

Форма обучения очная

Год начала подготовки 2023

Автор программы _____ А.И. Мордовин

Заведующий кафедрой
Систем информационной
безопасности _____ А.Г. Остапенко

Руководитель ОПОП _____ А.Г. Остапенко

Воронеж 2023

1. ЦЕЛИ И ЗАДАЧИ ДИСЦИПЛИНЫ

1.1. Цели дисциплины - изучение основных методов и алгоритмов программирования, их применение для решения прикладных задач, связанных с поиском и сортировкой информации, формирование навыков к анализу эффективности функционирования алгоритмов.

1.2. Задачи освоения дисциплины

- изучить исходные понятия и формализации в языках программирования;
- освоить процессы представления, анализа работы языков программирования;
- сформировать умения использовать нормативную, правовую, справочную документацию и специальную литературу.

2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОПОП

Дисциплина «Технологии и методы программирования» относится к дисциплинам блока Б1.

3. ПЕРЕЧЕНЬ ПЛАНИРУЕМЫХ РЕЗУЛЬТАТОВ ОБУЧЕНИЯ ПО ДИСЦИПЛИНЕ

Процесс изучения дисциплины «Технологии и методы программирования» направлен на формирование следующих компетенций:

ОПК-7 - способен создавать программы на языке высокого уровня, применять существующие реализации структур данных и алгоритмов.

| Компетенция | Результаты обучения, характеризующие сформированность компетенции |
|-------------|---|
| ОПК-7 | <p>Знать:</p> <ul style="list-style-type: none">- современные технологии и методы программирования;- основные алгоритмы сортировки и поиска данных, комбинаторные и теоретико-графовые алгоритмы;- алгоритмы эффективного поиска информации в массивах и строках данных. <p>Уметь:</p> <ul style="list-style-type: none">- проектировать структуру и архитектуру программного обеспечения с использованием современных методологий и средств автоматизации проектирования программного обеспечения;- производить анализ и выбор эффективного |

| |
|---|
| <p>алгоритма поиска и сортировки информации в различных массивах и строках данных с переменными любого типа;</p> <p>- оценивать технические возможности и вырабатывать рекомендации по разработке программного кода для эффективного выполнения проекта.</p> |
| <p>Владеть:</p> <p>- навыками проектирования блок-схем, функциональных схем и алгоритмов реализации программного кода;</p> <p>- основами построения блок-схем, алгоритмов выполнения программ с выбранным методом сортировки или поиска информации;</p> <p>- методами оценки эффективности алгоритмов сортировки или поиска информации в различных массивах и строках данных с переменными любого типа</p> |

4. ОБЪЕМ ДИСЦИПЛИНЫ

Общая трудоемкость дисциплины «Технологии и методы программирования» составляет 8 з.е.

Распределение трудоемкости дисциплины по видам занятий

| Виды учебной работы | Всего часов | Семестры | |
|---|-------------|----------|----|
| | | 5 | 6 |
| Контактная работа по видам занятий (всего) | 162 | 108 | 54 |
| В том числе: | | | |
| Лекции | 54 | 36 | 18 |
| Лабораторные работы (ЛР) | 108 | 72 | 36 |
| Самостоятельная работа | 54 | 36 | 18 |
| Курсовой проект | | + | - |
| Часы на контроль | 72 | 36 | 36 |
| Виды промежуточной аттестации - экзамен | | + | + |

| | | | |
|--|----------|----------|----------|
| Общая трудоемкость: академические часы зач.ед. | 288 8 | 180 5 | 108 3 |
|--|----------|----------|----------|

5. СОДЕРЖАНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)

5.1 Содержание разделов дисциплины и распределение трудоемкости по видам занятий

| № п/п | Наименование темы | Содержание раздела | Лекц | Лабор. работа | СРС | Всего, час |
|-------|---|--|------|---------------|-----|------------|
| 1 | Понятие сложности алгоритма. Элементарные структуры данных. | Пространственная и временная сложность. Классы сложности. О-сложность алгоритмов. Зависимость сложности алгоритма и сложности данных. Определение сложности в О-обозначениях. Массивы. Список. Односвязный список. Двусвязный список. Кольцевой список | 4 | 8 | 4 | 19 |
| 2 | Абстрактные типы данных. Рекурсия и деревья | Очередь, стек. Очередь FIFO, очередь LIFO. Рекурсивные алгоритмы, функция вычисления факториала, сомнительная рекурсивная программа. Алгоритм Евклида, рекурсивная программа для оценки префиксных выражений. Деревья, обход дерева. Деревья с корнем, бинарные деревья. | 4 | 16 | 4 | 24 |
| 3 | Элементарные методы сортировки массивов данных | Сортировка выбором, сортировка вставками, пузырьковая сортировка. Характеристики производительности элементарных методов сортировки. Сортировка методом Шелла, быстрая сортировка. Характеристики производительности быстрой сортировки. Сортировка слиянием. Двухпутевое слияние. | 8 | 8 | 8 | 24 |
| 4 | Элементарные методы поиска информации в массивах и строках данных | Поиск с использованием индексации по ключам. Таблица символов, основывающаяся на индексированном по ключам массиве. Последовательный поиск. Таблица символов с использованием массива. Таблица символов с использованием связного списка. Бинарный поиск в таблице символов, основанной на массиве. | 4 | 16 | 4 | 24 |
| 5 | Бинарные деревья поиска | Таблица символов на базе дерева бинарного поиска. Создание дерева бинарного поиска. Сортировка с помощью BST-дерева. Вставка в BST – дерево (нерекурсивная). Производительность деревьев поиска. Поиск произвольного ключа в дереве. Необходимость балансировки деревьев. Рандомизированные BST-деревья. Балансировка BST-дерева. Удаление в рандомизованном BST-дерева. | 4 | 8 | 6 | 18 |
| 6 | Красно-черные деревья | Свойства красно-черных деревьев. Пример красно-черного дерева. Повороты. Вставка. Вставка узла в красно-черное дерево с n узлами. Удаление узла красно-черного дерева. | 8 | 16 | 8 | 32 |
| 7 | Точный поиск подстроки в строке | Задача поиска подстрок. Алгоритмы поиска подстрок. Время предварительной обработки и сравнения алгоритмов поиска. Простейший алгоритм поиска подстрок. Алгоритм Рабина-Карпа. | 6 | 8 | 6 | 20 |
| 8 | Алгоритмы на графах | Свойства и типы графов. Глоссарий. Полные графы. Двухдольный граф. АТД графа. Интерфейс АТД графа. Пример клиентской функции обработки графов. Клиентская функция печати графа. Интерфейс ввода/вывода для функций обработки графов. | 8 | 16 | 6 | 30 |

| | | | | | | |
|--------------|---|---|-----------|------------|-----------|------------|
| | | Пример клиентской программы обработки графов. Алгоритмы обхода графа в глубину. Алгоритм обхода графа в ширину. Алгоритм Дейкстры. Алгоритм Флойда. Практическая реализация алгоритма Флойда. | | | | |
| 9 | Алгоритмы нахождения минимального остовного дерева в графе. | Алгоритм Дейкстры-Прима. Принцип работы алгоритма Дейкстры-Прима. Алгоритм Крускала. Принцип работы алгоритма Крускала. Алгоритмы нахождения максимального потока транспортной сети. Остаточные сети. Увеличивающие пути. Разрезы транспортных сетей. | 8 | 12 | 8 | 28 |
| Итого | | | 54 | 108 | 54 | 288 |

5.2 Перечень лабораторных занятий

1. Определение сложности алгоритмов – 8 ч.
2. Сортировка выбором – 16 ч.
3. Сортировка вставками – 8 ч.
4. Пузырьковая сортировка – 16 ч.
5. Сортировка методом Шелла – 8 ч.
6. Быстрая сортировка – 16 ч.
7. Последовательный поиск – 8 ч.
8. Бинарный поиска – 16 ч.
9. Алгоритм Робина-Карпа – 12 ч.

6. ПРИМЕРНАЯ ТЕМАТИКА КУРСОВЫХ ПРОЕКТОВ (РАБОТ) И КОНТРОЛЬНЫХ РАБОТ

Учебным планом по дисциплине «Технологии и методы программирования» предусмотрено выполнение курсового проекта в 5-ом семестре. Примерные темы курсового проектирования:

- реализовать класс красно-черного дерева и проанализировать эффективность поиска в дереве;
- реализовать программу, использующую В-дерево для нахождения записей в файле по идентификатору. Оценить эффективность поиска записей;
- реализовать класс очереди с приоритетом и на его базе реализовать алгоритм пирамидальной сортировки;
- реализовать класс динамической хеш-таблицы, оценить эффективность поиска в ней по мере роста количества элементов;
- разработать программу, находящую максимальный пропускной поток транспортной системы, заданной графом, ребра которого имеют вес равный соответствующей пропускной способности отдельного участка.

Курсовой проект включает в себя решение практических задач из различных разделов курса «Технологии и методы программирования». Курсовой проект выполняется студентами в соответствии с заданным вариантом в соответствии с «Методическими указаниями к выполнению курсового проекта...».

7. ОЦЕНОЧНЫЕ МАТЕРИАЛЫ ДЛЯ ПРОВЕДЕНИЯ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ОБУЧАЮЩИХСЯ ПО

ДИСЦИПЛИНЕ

7.1. Описание показателей и критериев оценивания компетенций на различных этапах их формирования, описание шкал оценивания

7.1.1 Этап текущего контроля

Результаты текущего контроля знаний и межсессионной аттестации оцениваются по следующей системе:

«аттестован»;

«не аттестован».

| Компетенция | Результаты обучения, характеризующие сформированность компетенции | Критерии оценивания | Аттестован | Не аттестован |
|-------------|---|--|---|---|
| ОПК-7 | Знать: - современные технологии и методы программирования; - основные алгоритмы сортировки и поиска данных, комбинаторные и теоретико-графовые алгоритмы; - алгоритмы эффективного поиска информации в массивах и строках данных. | Активная работа на практических занятиях, отвечает на теоретические вопросы при защите курсового проекта | Выполнение работ в срок, предусмотренный в рабочих программах | Невыполнение работ в срок, предусмотренный в рабочих программах |
| | Уметь: - проектировать структуру и архитектуру программного обеспечения с использованием современных методологий и средств автоматизации проектирования программного обеспечения; - производить анализ и выбор эффективного алгоритма поиска и сортировки информации в различных массивах и строках данных с переменными любого типа; - оценивать технические возможности и вырабатывать рекомендации по разработке программного кода для эффективного выполнения проекта. | Решение стандартных практических задач, написание курсового проекта | Выполнение работ в срок, предусмотренный в рабочих программах | Невыполнение работ в срок, предусмотренный в рабочих программах |
| | Владеть: - навыками проектирования блок-схем, функциональных схем и алгоритмов реализации программного кода; - основами построения блок-схем, алгоритмов выполнения программ с выбранным методом сортировки или поиска информации; - методами оценки эффективности алгоритмов сортировки или поиска | Решение прикладных задач в конкретной предметной области, выполнение плана работ по разработке курсового проекта | Выполнение работ в срок, предусмотренный в рабочих программах | Невыполнение работ в срок, предусмотренный в рабочих программах |

| | | | | |
|--|--|--|--|--|
| | информации в различных массивах и строках данных с переменными любого типа | | | |
|--|--|--|--|--|

7.1.2 Этап промежуточного контроля знаний

Результаты промежуточного контроля знаний оцениваются в 5 и 6 семестрах по четырехбальной системе:

«отлично»;

«хорошо»;

«удовлетворительно»;

«неудовлетворительно».

| Компетенция | Результаты обучения, характеризующие сформированность компетенции | Критерии оценивания | Отлично | Хорошо | Удовл | Неудовл |
|-------------|---|--|--|---|---|--|
| ОПК-7 | <p>Знать:</p> <ul style="list-style-type: none"> - современные технологии и методы программирования; - основные алгоритмы сортировки и поиска данных, комбинаторные и теоретико-графовые алгоритмы; - алгоритмы эффективного поиска информации в массивах и строках данных <p>Уметь:</p> <ul style="list-style-type: none"> - проектировать структуру и архитектуру программного обеспечения с использованием современных методологий и средств автоматизации проектирования программного обеспечения; - производить анализ и выбор эффективного алгоритма поиска и сортировки информации в различных массивах и строках данных с переменными любого типа; - оценивать технические возможности и выработать рекомендации по | <p>знание учебного материала и использование учебного материала в процессе выполнения заданий</p> <p>умение использовать учебный материал в процессе выполнения практических работ</p> | <p>Студент демонстрирует полное понимание учебного материала.</p> <p>Студент демонстрирует ярко выраженную способность использовать знания, умения, навыки в процессе выполнения заданий</p> | <p>Студент демонстрирует значительное понимание материала.</p> <p>Студент демонстрирует способность использовать знания, умения, навыки в процессе выполнения заданий</p> | <p>Студент демонстрирует частичное понимание материала.</p> <p>Способность студента продемонстрировать знание, умение, навык выражена слабо</p> | <p>1. Студент демонстрирует незначительное понимание материала.</p> <p>2. Студент демонстрирует непонимание заданий.</p> <p>3. У студента нет ответа. Не было попытки выполнить задания.</p> |

| | | | | | | |
|--|---|--|--|--|--|--|
| | разработке программного кода для эффективного выполнения проекта. | | | | | |
| | Владеть: - навыками проектирования блок-схем, функциональных схем и алгоритмов реализации программного кода; - основами построения блок-схем, алгоритмов выполнения программ с выбранным методом сортировки или поиска информации; - методами оценки эффективности алгоритмов сортировки или поиска информации в различных массивах и строках данных с переменными любого типа | применение учебного материала при решении практических задач | | | | |

7.2 Примерный перечень оценочных средств (типовые контрольные задания или иные материалы, необходимые для оценки знаний, умений, навыков и (или) опыта деятельности)

7.2.1 Примерный перечень заданий для подготовки к тестированию (минимум 10 вопросов для тестирования с вариантами ответов)

1. Что НЕ является определением алгоритма:
 - а) Точное предписание, однозначно определяющее вычислительный процесс, ведущий от варьируемых начальных данных к искомому результату
 - б) Процедура, которая принимает любой из возможных входных экземпляров задачи и преобразует его в соответствии с требованиями, указанными в условии задачи
 - в) Параметр, характеризующий данные, называют иногда объемом данных или сложностью данных
 - г) Конечный набор правил, однозначно раскрывающих содержание и последовательность выполнения операций для систематического решения определенного класса задач за конечное число шагов
2. Рассчитать сложность алгоритма $O(2*N+N^2)$
 - а) $O(N)$
 - б) $O(1)$
 - в) $O(2N)$

г) $O(N^2)$

3. Рассчитать НОД (алгоритм Евклида) 2 чисел – 357444 и 2154:

а) 1

б) 6

в) 2

г) 3

4. Начальный элемент односвязанного списка:

а) head list;

б) prev list;

в) next list;

г) nil

5. Стек, функционирующий по принципу "последним пришел – первым вышел":

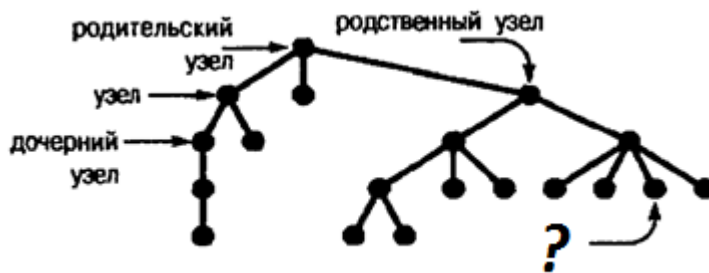
а) FIFO,

б) LIFO

в) FIOS

г) FILO

6. Назовите элемент дерева, обозначенный знаком вопроса



а) корень

б) узел

в) лист

г) дочерний узел

7. Какое из выражений не является условием, чтобы граф считался деревом

а) Граф имеет N ребер и один цикл

б) Граф имеет $N - 1$ ребер и является связным

в) Только один простой путь соединяет каждую пару вершин в графе

г) Граф является связным, но перестает быть таковым при удалении любого ребра

8. Определить тип сортировки массива

A S O R T I N G E X A M P L E
 A S O R T I N G E X A M P L E
 A A O R T I N G E X S M P L E
 A A E R T I N G O X S M P L E
 A A E E T I N G O X S M P L R
 A A E E G I N T O X S M P L R
 A A E E G I N T O X S M P L R
 A A E E G I L T O X S M P N R
 A A E E G I L M O X S T P N R
 A A E E G I L M N X S T P O R
 A A E E G I L M N O S T P X R
 A A E E G I L M N O P T S X R
 A A E E G I L M N O P R S X T
 A A E E G I L M N O P R S X T
 A A E E G I L M N O P R S T X
 A A E E G I L M N O P R S T X

- а) пузырьковая
- б) вставками
- в) выбором
- г) шейкер

9. Вставьте пропущенные значения (вместо знаков вопроса)

Пузырьковая сортировка производит в среднем примерно ? операций сравнения и ? операций обмена как в среднем, так и в наихудшем случаях:

- а) $N^2/2$, $N^2/2$
- б) $N/2$, $N^2/2$
- в) $N/2$, $N/2$
- г) N^2 , N^2

10. Определить, каким видом сортировки является часть программы:

```

template <class Item>
void sort(Item a[] , int l, int r)
{
    if (r <= l) return;
    int i = partition (a, l, r);
    sort(a, l, i-1);
    sort(a, i+1, r);
}
  
```

- а) сортировка методом Шелла
- б) быстрая сортировка
- в) пузырьковая
- г) вставками

7.2.4 Примерный перечень вопросов для подготовки к экзамену

1. Понятие сложности алгоритма. Пространственная и временная сложность.
2. Классы сложности. O-сложность алгоритмов. Зависимость сложности алгоритма и сложности данных.
3. Массивы. Список. Односвязный список. Двусвязный список. Кольцевой список

4. Очередь, стек. Очередь FIFO, очередь LIFO.
5. Рекурсивные алгоритмы, функция вычисления факториала, сомнительная рекурсивная программа.
6. Алгоритм Евклида, рекурсивная программа для оценки префиксных выражений..
7. Деревья, обход дерева. Деревья с корнем, бинарные деревья.
8. Сортировка выбором, сортировка вставками, пузырьковая сортировка.
9. Характеристики производительности элементарных методов сортировки. Сортировка методом Шелла, быстрая сортировка.
10. Характеристики производительности быстрой сортировки. Сортировка слиянием. Двухпутевое слияние.
11. Поиск с использованием индексации по ключам. Таблица символов, основывающаяся на индексированном по ключам массиве. Последовательный поиск.
12. Таблица символов с использованием массива. Таблица символов с использованием связного списка. Бинарный поиск в таблице символов, основанной на массиве.
13. Таблица символов на базе дерева бинарного поиска. Создание дерева бинарного поиска. Сортировка с помощью BST-дерева.
14. Вставка в BST – дерево (нерекурсивная). Производительность деревьев поиска.
15. Поиск произвольного ключа в дереве. Необходимость балансировки деревьев.
16. Рандомизированные BST-деревья. Балансировка BST-дерева. Удаление в рандомизованном BST-дереве.
17. Свойства красно-черных деревьев. Пример красно-черного дерева.
18. Повороты. Вставка. Вставка узла в красно-черное дерево с n узлами.
19. Удаление узла красно-черного дерева.
20. Задача поиска подстроки. Алгоритмы поиска подстроки.
21. Время предварительной обработки и сравнения алгоритмов поиска.
22. Простейший алгоритм поиска подстроки. Алгоритм Рабина-Карпа.
23. Свойства и типы графов. Глоссарий. Полные графы.
24. Двухдольный граф. АТД графа.
25. Интерфейс АТД графа. Пример клиентской функции обработки графов.
26. Клиентская функция печати графа. Интерфейс ввода/вывода для функций обработки графов.
27. Пример клиентской программы обработки графов.
28. Алгоритмы обхода графа в глубину.
29. Алгоритм обхода графа в ширину.
30. Алгоритм Дейкстры.
31. Алгоритм Флойда. Практическая реализация алгоритма Флойда.
32. Алгоритм Дейкстры-Прима. Принцип работы алгоритма Дейкстры-Прима.

33. Алгоритм Крускала. Принцип работы алгоритма Крускала..
34. Алгоритмы нахождения максимального потока транспортной сети.
35. Остаточные сети. Увеличивающие пути. Разрезы транспортных сетей.

7.2.7 Паспорт оценочных материалов

| № п/п | Контролируемые разделы (темы) дисциплины | Код контролируемой компетенции | Наименование оценочного средства |
|-------|---|--------------------------------|--|
| 1 | Понятие сложности алгоритма. Элементарные структуры данных | ОПК-7 | Тест, решение практических задач, выполнение курсового проекта |
| 2 | Абстрактные типы данных. Рекурсия и деревья | ОПК-7 | Тест, решение практических задач, выполнение курсового проекта |
| 3 | Элементарные методы сортировки массивов данных | ОПК-7 | Тест, решение практических задач, выполнение курсового проекта |
| 4 | Элементарные методы поиска информации в массивах и строках данных | ОПК-7 | Тест, решение практических задач, выполнение курсового проекта |
| 5 | Бинарные деревья поиска | ОПК-7 | Тест, решение практических задач, выполнение курсового проекта |
| 6 | Красно-черные деревья | ОПК-7 | Тест, решение практических задач, выполнение курсового проекта |
| 7 | Точный поиск подстроки в строке | ОПК-7 | Тест, решение практических задач, выполнение курсового проекта |
| 8 | Алгоритмы на графах | ОПК-7 | Тест, решение практических задач, выполнение курсового проекта |
| 9 | Алгоритмы нахождения минимального остовного дерева в графе. | ОПК-7 | Тест, решение практических задач, выполнение курсового проекта |

7.3. Методические материалы, определяющие процедуры оценивания знаний, умений, навыков и (или) опыта деятельности

При преподавании дисциплины «Технологии и методы программирования» в качестве формы оценки знаний студентов используются: тесты, решение практических задач различной сложности, экзамен.

Тестирование осуществляется, либо при помощи компьютерной системы тестирования, либо с использованием выданных тест-заданий на бумажном носителе. Время тестирования 30 мин. Затем осуществляется проверка теста экзаменатором и выставляется оценка согласно методики выставления оценки при проведении промежуточной аттестации.

Решение стандартных задач осуществляется, либо при помощи компьютерной системы тестирования, либо с использованием выданных задач на бумажном носителе. Время решения задач 30 мин. Затем осуществляется проверка решения задач экзаменатором и выставляется оценка, согласно методики выставления оценки при проведении промежуточной аттестации.

Решение прикладных задач осуществляется, либо при помощи компьютерной системы тестирования, либо с использованием выданных задач на бумажном носителе. Время решения задач 30 мин. Затем осуществляется проверка решения задач экзаменатором и выставляется оценка, согласно методики выставления оценки при проведении промежуточной аттестации.

8 УЧЕБНО МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

8.1 Перечень учебной литературы, необходимой для освоения дисциплины

Основная:

1. Пантелеев Е.Р. Алгоритмы и структуры данных: учебное пособие / Е.Р. Пантелеев, А.Л. Алыкова. — Иваново: ИГЭУ, 2018. — 142 с. — Текст: электронный // Лань: электронно-библиотечная система
2. Окулов С.М. Абстрактные типы данных: учебное пособие / С.М. Окулов. - 3-е изд. - Москва: Лаборатория знаний, 2020. - 253 с. - ISBN 978-5-00101-891-9. - Текст: электронный // Лань: электронно-библиотечная система
3. Самойленко А.П. Оптимизационные методы синтеза графовых структур топологий телекоммуникационных систем: учебное пособие / А.П. Самойленко, О.А. Усенко. - Ростов-на-Дону: ЮФУ, 2016. - 241 с. - ISBN 978-5-9275-2089-3. - Текст: электронный // Лань: электронно-библиотечная система
4. Симонова Е.В. Структуры данных в C#: линейные и нелинейные динамические структуры: учебное пособие / Е.В. Симонова. - Санкт-Петербург: Лань, 2021. - 152 с. - ISBN 978-5-8114-3098-7. - Текст: электронный // Лань: электронно-библиотечная система.
5. Павлов Л. А. Структуры и алгоритмы обработки данных: учебник для

вузов / Л.А. Павлов, Н.В. Первова. - 3-е изд., стер. - Санкт-Петербург: Лань, 2021. - 256 с. - ISBN 978-5-8114-7259-8. - Текст: электронный // Лань: электронно-библиотечная система.

Дополнительная:

1. Роберт Седжвик. Фундаментальные алгоритмы на С. Части 1 - 5. Анализ. Структуры данных. Сортировка. Поиск. Алгоритмы на графах. 2003. — 1136 с.
2. Кормен, Томас Х., Лейзерсон, Чарльз И., Ривест, Рональд Л., Штайн, Клиффорд. Алгоритмы: построение и анализ, 2-е издание. — М.:Издат. дом "Вильямс", 2005. —1296с.
3. Седжвик Роберт. Фундаментальные алгоритмы на С++. Часть 5: Алгоритмы на графах. — СПб.: ООО «ДиаСофтЮП», 2002. — 496 с.
4. Быков А.Ю. Решение задач на языках программирования Си и Си++: методические указания / А. Ю. Быков. - Москва: МГТУ им. Н.Э. Баумана, 2017. - 248 с. - ISBN 978-5-7038-4577-6.- Текст: электронный // Лань: электронно-библиотечная система.
5. Свердлов С.З. Языки программирования и методы трансляции: учебное пособие для вузов / С.З. Свердлов. - 2-е изд., стер. - Санкт-Петербург: Лань, 2021. - 564 с. - ISBN 978-5-8114-8195-8.
6. Непейвода Н.Н. Стили и методы программирования: учебное пособие / Н. Н. Непейвода. - 2-е изд. - Москва: ИНТУИТ, 2016. - 295 с. - ISBN 5-9556-0023-Х.
7. Струченков В.И. Методы оптимизации в прикладных задачах: учебное пособие / В.И. Струченков. - Москва: СОЛОН-Пресс, 2009. - 320 с. - ISBN 978-5-91359-061-9.
8. Окулов С. М. Динамическое программирование: учебное пособие / С.М. Окулов, О. А. Пестов. - 3-е изд. - Москва: Лаборатория знаний, 2020. - 299 с. - ISBN 978-5-00101-683-0. - Текст: электронный // Лань: электронно-библиотечная система.

Методические разработки:

1. Методические указания к выполнению курсового проекта по дисциплине «Методы программирования» для студентов специальности 10.05.02 «Безопасность телекоммуникационных систем» очной формы обучения / ФГБОУ ВПО «Воронежский государственный технический университет»; сост. Д.О. Карпеев. Воронеж, 2020. - 1 файл. - 00-00.
2. Методические указания к выполнению лабораторных работ по дисциплине «Методы программирования» для студентов специальности 10.05.02 «Безопасность телекоммуникационных систем» очной формы

обучения / ФГБОУ ВПО «Воронежский государственный технический университет»; сост. Д.О. Карпеев. Воронеж, 2020. - 1 файл. - 00-00.

8.2 Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине, включая перечень лицензионного программного обеспечения, ресурсов информационно-телекоммуникационной сети «Интернет», современных профессиональных баз данных и информационных справочных систем:

<http://www.eios.vorstu.ru> (электронная информационно-обучающая система ВГТУ)

<http://e.lanbook.com/> (ЭБС Лань)

<http://znanium.com/> (ЭБС Знаниум)

<http://IPRbookshop.ru/> (ЭБС IPRbooks (Айбукс))

<http://urait.ru/> (Образовательная платформа «Юрайт»)

9 МАТЕРИАЛЬНО-ТЕХНИЧЕСКАЯ БАЗА, НЕОБХОДИМАЯ ДЛЯ ОСУЩЕСТВЛЕНИЯ ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА

Специализированная лекционная аудитория, оснащенная оборудованием для лекционных демонстраций и проекционной аппаратурой.

10. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ОБУЧАЮЩИХСЯ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ (МОДУЛЯ)

По дисциплине «Технологии и методы программирования» читаются лекции, проводятся лабораторные работы, выполняется курсовой проект.

Основой изучения дисциплины являются лекции, на которых излагаются наиболее существенные и трудные вопросы, а также вопросы, не нашедшие отражения в учебной литературе.

На лабораторных работах проводится тестирование и решение задач в соответствии с темой занятия. Методики решения задач приведены в методических указаниях к практическим занятиям.

Большое значение по закреплению и совершенствованию знаний имеет самостоятельная работа студентов. Информацию о всех видах самостоятельной работы студенты получают на занятиях.

Контроль усвоения материала дисциплины производится проверкой курсового проекта, выполнения тестов, решения практической задач. Освоение дисциплины оценивается на экзамене.

| Вид учебных занятий | Деятельность студента |
|---------------------------------------|--|
| Лекция | Написание конспекта лекций: кратко, схематично, последовательно фиксировать основные положения, выводы, формулировки, обобщения; пометать важные мысли, выделять ключевые слова, термины. Проверка терминов, понятий с помощью энциклопедий, словарей, справочников с выписыванием толкований в тетрадь. Обозначение вопросов, терминов, материала, которые вызывают трудности, поиск ответов в рекомендуемой литературе. Если самостоятельно не удастся разобраться в материале, необходимо сформулировать вопрос и задать преподавателю на лекции или на практическом занятии. |
| Лабораторная работа | Лабораторные работы позволяют научиться применять теоретические знания, полученные на лекции при решении конкретных задач. Чтобы наиболее рационально и полно использовать все возможности лабораторных работ для подготовки к ним необходимо: следует разобрать лекцию по соответствующей теме, ознакомиться с соответствующим разделом учебного пособия по данной дисциплине, проработать дополнительную литературу и источники, решить задачи для самостоятельного решения из соответствующего раздела методических указаний к лабораторным работам. |
| Самостоятельная работа | Самостоятельная работа студентов способствует глубокому усвоению учебного материала и развитию навыков самообразования. Самостоятельная работа предполагает следующие составляющие: - работа с текстами: учебниками, справочниками, дополнительной литературой, а также проработка конспектов лекций; - выполнение домашних заданий и расчетов; - работа над темами для самостоятельного изучения; - участие в работе студенческих научных конференций, олимпиад; - подготовка к промежуточной аттестации. |
| Подготовка к промежуточной аттестации | Готовиться к промежуточной аттестации следует систематически, в течение всего семестра. Интенсивная подготовка должна начинаться не позднее, чем за месяц-полтора до промежуточной аттестации. Данные перед экзаменом три дня эффективнее всего использовать для повторения и систематизации материала. |