

Министерство науки и высшего образования Российской Федерации
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ (НИ ТГУ)

Радиофизический факультет

УТВЕРЖДЕНО:

Декан

А. Г. Коротаев

Оценочные материалы по дисциплине

Программирование на C++ часть 2

по направлению подготовки / специальности

11.05.01 Радиоэлектронные системы и комплексы

Направленность (профиль) подготовки/ специализация:
Программное обеспечение микропроцессорных систем

Форма обучения

Очная

Квалификация

Инженер-программист

Год приема

2024

СОГЛАСОВАНО:

Руководитель ОП

С.Н. Торгаев

Председатель УМК

А.П. Коханенко

Томск – 2025

1. Компетенции и индикаторы их достижения, проверяемые данными оценочными материалами

Целью освоения дисциплины является формирование следующих компетенций:

ОПК-4 Способен проводить экспериментальные исследования и владеть основными приемами обработки и представления экспериментальных данных.

ОПК-5 Способен выполнять опытно-конструкторские работы с учетом требований нормативных документов в области радиоэлектронной техники и информационно-коммуникационных технологий.

ОПК-7 Способен понимать принципы работы современных информационных технологий и использовать их для решения задач профессиональной деятельности.

ОПК-8 Способен использовать современные программные и инструментальные средства компьютерного моделирования для решения различных исследовательских и профессиональных задач.

ОПК-9 Способен разрабатывать алгоритмы и компьютерные программы, пригодные для практического применения.

ПК-6 Способен применять методы искусственного интеллекта и машинного обучения в задачах обработки сигналов, анализа результатов и управления параметров систем связи.

Результатами освоения дисциплины являются следующие индикаторы достижения компетенций:

РООПК 4.3 Владеет способами обработки и представления полученных экспериментальных результатов

РООПК 5.3 Владеет современными компьютерными системами проектирования

РООПК 7.1 Знает современные информационно-коммуникационные технологии для обработки, анализа и представления в требуемом формате информации

РООПК 7.2 Умеет решать информационно-коммуникационные задачи с помощью современных систем автоматизации

РООПК 8.1 Знает современные информационные технологии и программное обеспечение при решении задач профессиональной деятельности

РООПК 8.2 Умеет использовать компьютерные системы поиска, хранения, обработки, анализа и представления информации

РООПК 9.1 Знает современные инструментальные системы программирования и компьютерного моделирования при решении прикладных задач.

РООПК 9.2 Владеет навыками работы в компьютерной среде.

РОПК 6.1 Знает основы машинного обучения и искусственного интеллекта

РОПК 6.2 Умеет применять методы машинного обучения в различных задачах и необходимое программное обеспечение для реализации методов искусственного интеллекта и машинного обучения

2. Оценочные материалы текущего контроля и критерии оценивания

Элементы текущего контроля:

- тесты;
- лабораторные работы.

Типовые тестовые вопросы (правильные ответы выделены курсивом).

№	Вопрос	Варианты ответа
1	Пусть объявлен булев вектор <code>unsigned short a;</code> Как эффективно реализовать операцию	a) <i><code>a = a >> k;</code></i> б) <i><code>a = a << k;</code></i> в) <i><code>a = 2 >> k;</code></i> г) <i><code>a = k << 2;</code></i>

	умножения переменной a на 2^k ? (РООПК 4.3, РООПК 5.3, РООПК 8.2, РООПК 9.2, РОПК 6.2)	
2	Как эффективно проверить, является ли значение переменной a чётным? (РООПК 4.3, РООПК 5.3, РООПК 8.2, РООПК 9.2, РОПК 6.2)	а) <code>if ((a & 1) == 0) {...}</code> б) <code>if ((a 1) == 0) {...}</code> в) <code>if ((a && 1) == 0) {...}</code> г) <code>if ((a 1) == 0) {...}</code>
3	Что представляет собой булев вектор <code>mask</code> , объявленный и проинициализированный в следующем фрагменте программного кода: <code>const int N = 16; unsigned short mask; mask = (~0) >> (N-k);</code> (РООПК 4.3, РООПК 5.3, РООПК 8.2, РООПК 9.2, РОПК 6.2)	а) Вектор, содержащий нули в k правых разрядах, а в остальных разрядах единицы; б) Вектор, содержащий единицы в k правых разрядах, а в остальных разрядах нули; в) Вектор, содержащий нули в $(N-k)$ правых разрядах, а в остальных разрядах единицы; г) Вектор, состоящий из всех единиц.
4.	Как можно иначе записать следующий фрагмент программы? <code>int a[100]; cin >> *((&a[2])-2);</code> (РООПК 4.3, РООПК 5.3, РООПК 8.2, РООПК 9.2, РОПК 6.2)	а) <code>int a[100]; cin >> a[2];</code> б) <code>int a[100]; cin >> a[0];</code> в) <code>int a[100]; cin >> &a[2];</code>
5	Сколько раз выполнится цикл? <code>for (int i = 0; i <=5; i+=3);</code> (РООПК 4.3, РООПК 5.3, РООПК 8.2, РООПК 9.2, РОПК 6.2)	а) ни одного раза; б) 3 раз; в) 1 раз; г) 2 раза.
6	Какова трудоемкость добавления элемента в список (без поиска позиции)? (РООПК 7.1-2, РООПК 8.1, РООПК 9.1, РОПК 6.1)	а) $T(n) \sim C$ б) $T(n) \sim n$
7	Какова трудоемкость поиска элемента в списке? (РООПК 7.1-2, РООПК 8.1, РООПК 9.1, РОПК 6.1)	а) $T(n) \sim C$ б) $T(n) \sim n$
8	Какое дерево называется двоичным (бинарным)? (РООПК 7.1-2, РООПК 8.1, РООПК 9.1, РОПК 6.1)	а) дерево, высота которого не превосходит 2; б) дерево, степень каждой вершины которого не превосходит 2;
9	Какое дерево называется идеально сбалансированным? (РООПК 7.1-2, РООПК 8.1, РООПК 9.1, РОПК 6.1)	а) дерево, в котором на всех уровнях (кроме последнего) помещается максимальное количество вершин;

		б) дерево, для любой вершины t которого все ключи левого поддеревя $\leq t.key$ ($< t.key$), а все ключи правого поддеревя $\geq t.key$ ($> t.key$); в) дерево, у каждой вершины которого высота левого поддеревя отличается от высоты правого поддеревя не более, чем на 1;
10	Какова трудоемкость поиска в дереве поиска с n вершинами? (РООПК 7.1-2, РООПК 8.1, РООПК 9.1, РОПК 6.1)	а) в лучшем случае – $\lceil \log_2 n \rceil + 1$, в худшем случае – n ; б) всегда C ; в) всегда $\lceil \log_2 n \rceil + 1$;

За тест выставляется оценка «зачтено», если дано не менее 70% правильных ответов, в противном случае, выставляется оценка «не зачтено».

Примеры задач для лабораторных работ (РООПК 4.3, РООПК 5.3, РООПК 8.2, РООПК 9.2, РОПК 6.2)

1. Сформировать битовую матрицу, в которой i -й столбец заполнен единицами, остальные элементы равны нулю.

2. Построить битовую матрицу, в которой единица в позиции (i, j) , остальные нули.

3. Построить битовую матрицу, в которой ноль в позиции (i, j) , остальные единицы.

4. Для заданной битовой матрицы определить значение элемента (i, j) .

5. Для заданной битовой матрицы построить инвертированную матрицу.

6. Сформировать битовую матрицу, главная диагональ которой заполнена нулями, остальные элементы – единицы.

7. Проверить, является ли заданная битовая матрица симметричной относительно главной диагонали.

8. Подсчитать количество единиц в битовой матрице.

9. Подсчитать количество нулей в битовой матрице.

10. Вывести номера столбцов битовой матрицы, в которых содержится хотя бы одна единица.

11. Вывести номера столбцов битовой матрицы, в которых содержатся только единицы.

12. Вывести количество столбцов битовой матрицы, в которых содержится хотя бы одна единица

13. Вывести количество столбцов битовой матрицы, в которых содержатся только единицы.

14. Написать программу для представления набора целых чисел в виде упорядоченного замкнутого списка (кольца). Реализовать операции вставки и удаления элементов списка (с сохранением порядка), ввода и вывода списка, уничтожения списка.

15. Написать программу для представления набора целых чисел в виде упорядоченного двусвязного списка. Реализовать операции вставки элементов списка (с сохранением порядка), ввода и вывода списка, уничтожения списка.

16. Написать программу для представления конечного автомата в виде односвязного списка, где элемент списка представляет собой переход в автомате, т.е. четверку <вход, начальное состояние перехода, финальное состояние перехода, выход>. Реализовать операции создания списка для заданного в файле автомата, поиска элемента списка по заданным входу и начальному состоянию перехода, вывода списка, уничтожения списка.

17. Написать программу построения частотного словаря слов заданного текста (файла). Для представления словаря использовать упорядоченный односвязный список,

каждый элемент которого служит для представления одного слова и содержит его (это слово) и количество его вхождений в текст. Реализовать операции построения списка, вывода списка и его уничтожения.

18. Написать программу для выполнения операций над многочленами любой степени от одной переменной. Для представления многочлена использовать односвязный список, каждый элемент которого представляет одночлен и содержит коэффициент и степень переменной. Реализовать операции ввода и вывода многочленов, сложения многочленов и вычисления значения многочлена при заданном значении переменной.

19. Написать программу построения частотного словаря слов некоторого текста в виде дерева поиска. Каждая вершина дерева служит для представления одного слова и содержит его (это слово) и количество его вхождений в текст. Значение любой вершины больше значений всех вершин ее левого поддерева и меньше значений всех вершин ее правого поддерева (упорядоченность по словам!). Реализовать операции создания дерева для заданного текста, вывода дерева на экран, поиска слова в дереве и уничтожения дерева.

20. Написать программу для представления набора целых чисел в виде AVL-дерева. Каждая вершина дерева служит для представления одного числа. Значение любой вершины больше значений всех вершин ее левого поддерева и меньше значений всех вершин ее правого поддерева. Реализовать операции вставки нового числа в дерево, вывода дерева на экран и уничтожения дерева.

21. Написать программу построения частотного словаря слов некоторого текста в виде AVL-дерева. Каждая вершина дерева служит для представления одного слова и содержит его (это слово) и количество его вхождений в текст. Значение любой вершины больше значений всех вершин ее левого поддерева и меньше значений всех вершин ее правого поддерева (упорядоченность по словам). Реализовать операции вставки нового узла в дерево, вывода дерева на экран и уничтожения дерева.

Критерии оценивания:

Результаты выполнения лабораторной работы определяются оценками «зачтено», «не зачтено».

Оценка «зачтено» выставляется, если разработанное программное обеспечение соответствует поставленному заданию, т. е. реализует заявленные функции (1), использует требуемые структуры данных (2) и не содержит избыточных элементов в программном коде (3).

Оценка «не зачтено» выставляется, если не выполнено хотя бы одно из условий 1, 2 и 3 предыдущего пункта. В таком случае, программная реализация отправляется на доработку.

3. Оценочные материалы итогового контроля (промежуточной аттестации) и критерии оценивания

Для получения зачёта необходимо выполнить три условия.

1) Получить оценку зачтено по трём лабораторным (по одной на каждый раздел курса) для подтверждения овладения компетенциями РООПК 4.3, РООПК 5.3, РООПК 8.2, РООПК 9.2, РОПК 6.2.

2) Получить оценку зачтено за тест (РООПК 4.3, РООПК 5.3, РООПК 7.1-2, РООПК 8.1-2, РООПК 9.1-2, РОПК 6.1-2).

3) Получить оценку зачтено на устном зачёте. Ответить на два теоретических вопроса (список приведён ниже) по двум темам дисциплины и два дополнительных вопроса для подтверждения овладения компетенциями РООПК 7.1-2, РООПК 8.1, РООПК 9.1, РОПК 6.1.

Вопросы к устному зачёту

1. Представление битовых векторов. Стандартные операции над битовыми векторами. Векторы-маски.
2. Указатели. Операции над указателями. Динамическое управление памятью.
3. Списковые информационные структуры. Операции над списками и их трудоемкость.
4. Древовидные информационные структуры.
5. Бинарные деревья поиска.
6. AVL-деревья. Балансировка.
7. Красно-чёрные деревья. Балансировка.
8. Операции над деревьями и их трудоемкость в зависимости от типа дерева.
9. Структура C++ программы. Функции: объявление, определение и вызов.
10. Пошаговая отладка программ.

Дополнительные вопросы

1. Битовые векторы и стандартные операции над ними.
2. Представление булевых векторов.
3. Реализация операций над булевыми векторами.
4. Векторы-маски.
5. Указатели. Операции над указателями.
6. Динамическое управление памятью.
7. Списковые информационные структуры.
8. Операции над списками и их трудоемкость.
9. Поиск элементов в массивах и связанных списках.
10. Древовидные информационные структуры.
11. Операции над деревьями и их трудоемкость.

Критерии оценивания:

Результаты устного опроса определяются оценками «зачтено», «не зачтено».

Оценка «зачтено» выставляется, если выполнены два из трёх перечисленных ниже условий. Даны правильные определения для фигурирующих в вопросе терминов (1). Раскрыт механизм функционирования для затрагиваемых в вопросе технологий (2). Приведены примеры использования обсуждаемых структур данных и алгоритмов (3).

Оценка «не зачтено» выставляется, если выполнено менее двух условий 1, 2 и 3 предыдущего пункта.

4. Оценочные материалы для проверки остаточных знаний (сформированности компетенций)

Тест (РООПК 4.3, РООПК 5.3, РООПК 8.2, РООПК 9.2, РОПК 6.2).

Правильные ответы выделены курсивом.

№	Вопрос	Варианты ответа
1	<p>Пусть объявлен булев вектор</p> <pre>unsigned short a;</pre> <p>Как эффективно реализовать операцию умножения переменной a на 2^k?</p>	<p>a) $a = a \gg k;$</p> <p><i>б) $a = a \ll k;$</i></p> <p>в) $a = 2 \gg k;$</p> <p>г) $a = k \ll 2;$</p>

2	Как эффективно проверить, является ли значение переменной <i>a</i> чётным?	<p>а) <code>if ((a & 1) == 0) {...}</code> б) <code>if ((a 1) == 0) {...}</code> в) <code>if ((a && 1) == 0) {...}</code> г) <code>if ((a 1) == 0) {...}</code></p>
3	<p>Что представляет собой булев вектор <i>mask</i>, объявленный и проинициализированный в следующем фрагменте программного кода:</p> <pre>const int N = 16; unsigned short mask; mask = (~0) >> (N-k);</pre>	<p>а) Вектор, содержащий нули в <i>k</i> правых разрядах, а в остальных разрядах единицы; б) Вектор, содержащий единицы в <i>k</i> правых разрядах, а в остальных разрядах нули; в) Вектор, содержащий нули в $(N-k)$ правых разрядах, а в остальных разрядах единицы; г) Вектор, состоящий из всех единиц.</p>
4.	<p>Как можно иначе записать следующий фрагмент программы?</p> <pre>int a[100]; cin >> *((&a[2])-2);</pre>	<p>а) <code>int a[100];</code> <code>cin >> a[2];</code> б) <code>int a[100];</code> <code>cin >> a[0];</code> в) <code>int a[100];</code> <code>cin >> &a[2];</code></p>
5	<p>Сколько раз выполнится цикл?</p> <pre>for (int i = 0; i <=5; i+=3);</pre>	<p>а) ни одного раза; б) 3 раз; в) 1 раз; г) 2 раза.</p>
6	Какое дерево называется двоичным (бинарным)?	<p>а) дерево, высота которого не превосходит 2; б) дерево, степень каждой вершины которого не превосходит 2;</p>
7	Какое дерево называется идеально сбалансированным?	<p>а) дерево, в котором на всех уровнях (кроме последнего) помещается максимальное количество вершин; б) дерево, для любой вершины <i>t</i> которого все ключи левого поддерева $\leq t.key$ ($< t.key$), а все ключи правого поддерева $\geq t.key$ ($> t.key$); в) дерево, у каждой вершины которого высота левого поддерева отличается от высоты правого поддерева не более, чем на 1;</p>

Теоретические вопросы (РООПК 7.1-2, РООПК 8.1, РООПК 9.1, РОПК 6.1).

1. Битовые векторы и стандартные операции над ними.
2. Представление булевых векторов.
3. AVL-деревья.
4. Красно-чёрные деревья.

5. Указатели. Операции над указателями.
6. Динамическое управление памятью.
7. Списковые информационные структуры.
8. Операции над списками и их трудоемкость.
9. Поиск элементов в массивах и связанных списках.
10. Древовидные информационные структуры.
11. Операции над деревьями и их трудоемкость.

Теоретические вопросы для проверки остаточных знаний предполагают краткое определение данного понятия и приведение примера его реализации в языке программирования C++.

Информация о разработчиках

Твардовский Александр Сергеевич, канд. физ.-мат. наук, кафедра компьютерной безопасности, доцент.