

Документ подписан простой электронной подписью

Информация о владельце:

ФИО: Таскаев Сергей Васильевич

Должность: Ректор

Дата подписания: 15.06.2026 12:16:04

Уникальный программный ключ:

04c19ed8bfb98f3b6cb77a48bb9a6788b8322523

МИНОБРАЗОВАНИЯ РОССИИ

Федеральное государственное бюджетное образовательное

учреждение высшего образования

«Челябинский государственный университет» (ФГБОУ ВО «ЧелГУ»)

Математический факультет

Кафедра вычислительной математики

Фонд оценочных средств по дисциплине "Фундаментальные алгоритмы на C++" по направлению подготовки (специальности) "01.03.02 ПРИКЛАДНАЯ МАТЕМАТИКА И ИНФОРМАТИКА" направленности (профилю)

«Прикладная информатика и искусственный интеллект» ФГБОУ ВО «ЧелГУ»

Версия документа - 1

стр. 1 из 18

Первый экземпляр _____

КОПИЯ № _____

**Фонд оценочных средств
для промежуточной аттестации
по дисциплине (модулю)
Фундаментальные алгоритмы на C++**

Направление подготовки (специальность)
01.03.02 ПРИКЛАДНАЯ МАТЕМАТИКА И ИНФОРМАТИКА

Направленность (профиль)
Прикладная математика и искусственный интеллект

Присваиваемая квалификация (степень)
бакалавр

Форма обучения
Очная

Год набора 2026

Челябинск 2026 г.



МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное образовательное
учреждение высшего образования
«Челябинский государственный университет» (ФГБОУ ВО «ЧелГУ»)
Математический факультет
Кафедра вычислительной математики

Фонд оценочных средств по дисциплине "Фундаментальные алгоритмы на C++" по направлению подготовки
(специальности) " 01.03.02 ПРИКЛАДНАЯ МАТЕМАТИКА И ИНФОРМАТИКА" направленности (профилю)
«Прикладная информатика и искусственный интеллект» ФГБОУ ВО «ЧелГУ»

Версия документа - 1

стр. 2 из 18

Первый экземпляр _____

КОПИЯ № _____

Содержание

1. Паспорт фонда оценочных средств
2. Перечень формируемых компетенций
 - 2.1. Компетенции, закрепленные за дисциплиной
3. Содержание оценочных средств по дисциплине
 - 3.1. Виды оценочных средств
 - 3.2. Содержание оценочных средств
4. Порядок проведения и критерии оценивания промежуточной аттестации
 - 4.1. Порядок проведения промежуточной аттестации
 - 4.2. Критерии оценивания промежуточной аттестации по видам оценочных средств
 - 4.3. Результаты промежуточной аттестации и уровни сформированности компетенций



МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное образовательное
учреждение высшего образования
«Челябинский государственный университет» (ФГБОУ ВО «ЧелГУ»)
Математический факультет
Кафедра вычислительной математики

Фонд оценочных средств по дисциплине "Фундаментальные алгоритмы на С++" по направлению подготовки (специальности) " 01.03.02 ПРИКЛАДНАЯ МАТЕМАТИКА И ИНФОРМАТИКА" направленности (профилю) «Прикладная информатика и искусственный интеллект» ФГБОУ ВО «ЧелГУ»

Версия документа - 1

стр. 3 из 18

Первый экземпляр _____

КОПИЯ № _____

1. ПАСПОРТ ФОНДА ОЦЕНОЧНЫХ СРЕДСТВ

Направление подготовки (специальность): 01.03.02 ПРИКЛАДНАЯ МАТЕМАТИКА И ИНФОРМАТИКА

Направленность (профиль): Прикладная математика и искусственный интеллект

Дисциплина: Фундаментальные алгоритмы на С++

Семестр (семестры) изучения: 5 семестр

Форма (формы) промежуточной аттестации: экзамен 5 семестр.

Используется балльно-рейтинговая система для оценивания результатов.

2. ПЕРЕЧЕНЬ ФОРМИРУЕМЫХ КОМПЕТЕНЦИЙ

2.1. Компетенции, закреплённые за дисциплиной

Изучение дисциплины «Фундаментальные алгоритмы на С++» направлено на формирование следующих компетенций:

Коды компетенции согласно ФГОС (ОПОП ВО)	Содержание компетенций согласно ФГОС (ОПОП ВО)	Индикаторы достижения компетенции согласно ОПОП	Перечень планируемых результатов обучения по дисциплине
1	2	3	4
ПК-2	Способен использовать базовые алгоритмы и средства проектирования программного обеспечения	ПК-2.1. Обладает знаниями о существующих типовых шаблонах проектирования программного обеспечения; о методах и средствах проектирования программного обеспечения, структур данных, баз данных. ПК-2.2. Демонстрирует умение: применять типо-	Знать: - основы разработки программного обеспечения (ПО) в рамках объектно-ориентированного подхода; - основы методов написания, оформления, отладки и тестирования ПО; Уметь: - реализовывать на языке С++ структуры данных, алгоритмы поиска и сортировки, различные



МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное образовательное
учреждение высшего образования
«Челябинский государственный университет» (ФГБОУ ВО «ЧелГУ»)
Математический факультет
Кафедра вычислительной математики

Фонд оценочных средств по дисциплине "Фундаментальные алгоритмы на С++" по направлению подготовки
(специальности) " 01.03.02 ПРИКЛАДНАЯ МАТЕМАТИКА И ИНФОРМАТИКА" направленности (профилю)
«Прикладная информатика и искусственный интеллект» ФГБОУ ВО «ЧелГУ»

Версия документа - 1

стр. 4 из 18

Первый экземпляр _____

КОПИЯ № _____

		<p>вые решения, библиотеки программных модулей, шаблоны, классы объектов, используемые при разработке программного обеспечения; применять методы и средства проектирования программного обеспечения, баз данных, структур данных.</p> <p>ПК-2.3. Имеет практический опыт (навыки): применения стандартных алгоритмов при проектирования программного обеспечения; разработки алгоритмов решения задач в соответствии с поставленными условиями; использования методов и приемов алгоритмизации поставленных задач.</p>	<p>ВЫЧИСЛИТЕЛЬНЫЕ алгоритмы</p> <p>Владеть:</p> <p>- навыками проектирования и анализа сложности алгоритмов, разрабатываемых в соответствии с поставленными условиями</p>
--	--	--	---



МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное образовательное
учреждение высшего образования
«Челябинский государственный университет» (ФГБОУ ВО «ЧелГУ»)
Математический факультет
Кафедра вычислительной математики

Фонд оценочных средств по дисциплине "Фундаментальные алгоритмы на С++" по направлению подготовки
(специальности) " 01.03.02 ПРИКЛАДНАЯ МАТЕМАТИКА И ИНФОРМАТИКА" направленности (профилю)
«Прикладная информатика и искусственный интеллект» ФГБОУ ВО «ЧелГУ»


Версия документа - 1

стр. 5 из 18

Первый экземпляр _____

КОПИЯ № _____

УК-4	Способен осуществлять деловую коммуникацию в устной и письменной формах на государственном языке Российской Федерации и иностранном(ых) языке(ах)	УК-4.1 Имеет представление о правилах и принципах деловой устной и письменной коммуникации на государственном языке Российской Федерации и иностранном(ых) языке(ах) УК-4.2 Демонстрирует умение осуществлять деловую коммуникацию в устной и письменной формах, использовать методы и навыки делового общения УК-4.3 Имеет навыки делового общения на государственном языке Российской Федерации и иностранном(ых) языке(ах)	Знать: - основные термины и речевые обороты, употребляющиеся в сфере компьютерных технологий, на русском и иностранном языке Уметь: - составлять тексты и сообщения с описанием технологических и программных характеристик разрабатываемых продуктов Владеть: - иметь навыки вербальной коммуникации на техническом иностранном языке
------	---	--	---

	МИНОБРНАУКИ РОССИИ Федеральное государственное бюджетное образовательное учреждение высшего образования «Челябинский государственный университет» (ФГБОУ ВО «ЧелГУ»)		
	Математический факультет Кафедра вычислительной математики		
Фонд оценочных средств по дисциплине "Фундаментальные алгоритмы на С++" по направлению подготовки (специальности) " 01.03.02 ПРИКЛАДНАЯ МАТЕМАТИКА И ИНФОРМАТИКА" направленности (профилю) «Прикладная информатика и искусственный интеллект» ФГБОУ ВО «ЧелГУ»			
Версия документа - 1	стр. 6 из 18	Первый экземпляр _____	КОПИЯ № _____

3. СОДЕРЖАНИЕ ОЦЕНОЧНЫХ СРЕДСТВ ПО ДИСЦИПЛИНЕ

3.1 Виды оценочных средств

№ п/п	Код компетенции/ планируемые результаты обучения	Контролируемые темы/ разделы	Наименование оценочного средства для текущего контроля	Наименование оценочного средства на промежуточной аттестации/№ задания
1	УК-4 ПК-2/ Имеет практический опыт применения стандартных алгоритмов при проектирования программного обеспечения;.	Основы современного языка С++	Домашняя работа	Теоретические вопросы к экзамену №1-11
2	ПК-2 УК-4/ имеет практический опыт в разработке алгоритмов решения задач в соответствии с поставленными условиями; использования методов и приемов алгоритмизации поставленных задач	Фундаментальные алгоритмы	Домашняя работа Семестровое задание	Теоретические вопросы к экзамену №12-33

Типовые задания, критерии и показатели оценивания в рамках текущего контроля представлены в рабочей программе дисциплины (модуля). Полные комплекты оценочных средств и контрольно-измерительных материалов хранятся на кафедре

3.2 Содержание оценочных средств

Темы типовых домашних заданий:

1. Шаблонные функции со стеками
2. Универсальные алгоритмы
3. Реализация операторов в классе матриц

Перечень домашних работ

1. Вводное задание



МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное образовательное
учреждение высшего образования
«Челябинский государственный университет» (ФГБОУ ВО «ЧелГУ»)
Математический факультет
Кафедра вычислительной математики

Фонд оценочных средств по дисциплине "Фундаментальные алгоритмы на C++" по направлению подготовки
(специальности) " 01.03.02 ПРИКЛАДНАЯ МАТЕМАТИКА И ИНФОРМАТИКА" направленности (профилю)
«Прикладная информатика и искусственный интеллект» ФГБОУ ВО «ЧелГУ»

Версия документа - 1

стр. 7 из 18

Первый экземпляр _____

КОПИЯ № _____

- a. Написать функцию, проверяющую, что три заданных числа являются сторонами некоторого треугольника.
 - b. Написать функцию, проверяющую, что заданный массив чисел является сторонами некоторого многоугольника.
 - c. Написать функцию, вычисляющую площадь многоугольника.
 - d. Написать функцию для нахождения палиндрома наибольшей длины, содержащегося в заданной строке символов.
2. Шаблонные функции с динамическими массивами
- a. Написать шаблонную функцию `AdjacentFind` для отыскания в заданном массиве пары одинаковых рядом стоящих значений.
 - b. Написать шаблонную функцию `Sorted`, проверяющую, что заданный массив является отсортированным.
 - c. Написать шаблонную функцию `ValueCount` для подсчёта количества вхождений в массив заданного значения.
 - d. Написать функцию `PermutatedArrays`, определяющую, являются ли два заданных массива перестановкой друг друга.
3. Шаблонные функции со стеками
- a. Написать шаблонную функцию `EqualStacks<Stack1,Stack2>`, проверяющую на совпадение содержимое двух стеков без их изменения.
 - b. Тестовые функции скорости работы:
 - i. `Test2` – кладёт в стек 1млн элемент и извлекает их, повторяет 100 раз.
 - ii. `Test3` – производит 200млн вероятностных операций: с вероятностью p кладёт элемент в стек и с вероятностью $1-p$ извлекает, если стек не пуст
 - c. Написать шаблонную функцию сортировки стека. Разрешается использовать только простые переменные, стеки того же типа и стандартные стековые операции.
4. Задачи со стеками
- a. Придумать тест для `quick_sort_stack`, на котором алгоритм падает или очень долго работает.
 - b. Найдите способ исправить `quick_sort_stack`.
 - c. Напишите шаблонную функцию, которая из двух данных отсортированных в порядке убывания стеков получает один с объединением их элементов.
 - d. Задача о затопленной брусчатке. Дорожка сложена из брусов единичной ширины и непостоянной высоты. После дождя в образовавшихся впадинах скопилась вода. Найти ее объем.
5. Задачи со стеками
- a. Реализовать алгоритм поиска эйлерова пути в графе.
 - b. Написать генератор случайного графа на N вершинах, содержащего эйлеров путь.
 - c. Реализовать функцию разделения строки с арифметическим выражением на термы.



МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное образовательное
учреждение высшего образования
«Челябинский государственный университет» (ФГБОУ ВО «ЧелГУ»)
Математический факультет
Кафедра вычислительной математики

Фонд оценочных средств по дисциплине "Фундаментальные алгоритмы на C++" по направлению подготовки
(специальности) " 01.03.02 ПРИКЛАДНАЯ МАТЕМАТИКА И ИНФОРМАТИКА" направленности (профилю)
«Прикладная информатика и искусственный интеллект» ФГБОУ ВО «ЧелГУ»

Версия документа - 1	стр. 8 из 18	Первый экземпляр _____	КОПИЯ № _____
----------------------	--------------	------------------------	---------------

- d. Дан набор отрезков на прямой, заданных началом и концом. Найти систему отрезков, являющуюся объединением данных.
6. Стек и не только
 - a. Дан набор отрезков на прямой, заданных началом и концом. Найти систему отрезков, являющуюся объединением данных.
 - b. Отладить и протестировать функции `GetEulerPath` и `GenerateEulerPath`.
 - c. Написать функцию, проверяющую является ли данное скобочное выражение правильным.
 - d. Дано скобочное выражение с одной «лишней» скобкой. Найти её индекс.
7. Универсальные алгоритмы
 - a. Написать функцию, преобразующую последовательность термов арифметического выражения в постфиксную запись.
 - b. Написать функцию, которая с помощью универсального алгоритма `count_if` находит в заданном целочисленном векторе количество элементов, являющихся простыми числами.
 - c. Написать функцию, которая для двух целочисленных векторов находит во втором минимальное число, отсутствующее в первом (использовать универсальные алгоритмы `find` и `find_if`).
 - d. Написать функцию, которая для заданного вектора точек на плоскости находит количество выпуклых многоугольников, на которое можно разбить это множество точек так, чтобы получилась кольцевая структура.
8. Вычисление арифметических выражений
 - a. В файле записано арифметическое выражение и значения переменных (через знак равно). Написать функцию, считывающую строку с арифметическим выражением и значения переменных в ассоциативный массив `map<string,double>`.
 - b. Имеется арифметическое выражение. Написать функцию, возвращающую номера избыточных скобок. Пара скобок называется избыточной, если её удаление не меняет порядок вычисления:
 $(a+b)+c$ – избыточная скобка;
 $a*(b*c)$ – нет.
 - c. Контрабандисты спрятали груз за рядом из стоящих вплотную друг к другу коробок. Написать функцию, которая по вектору размеров коробок устанавливает, какой максимальной площади контрабандную коробку можно спрятать позади них так, чтобы ее не было заметно.
9. Рекурсия, скобки и прочее
 - a. Реализовать рекурсивный алгоритм поиска максимальной контрабандной коробки.
 - b. Реализовать алгоритм поиска лишних скобок в арифметическом выражении с помощью сравнения постфиксных записей до и после удаления. Для этого осуществить следующие этапы:



МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное образовательное
учреждение высшего образования
«Челябинский государственный университет» (ФГБОУ ВО «ЧелГУ»)
Математический факультет
Кафедра вычислительной математики

Фонд оценочных средств по дисциплине "Фундаментальные алгоритмы на C++" по направлению подготовки
(специальности) " 01.03.02 ПРИКЛАДНАЯ МАТЕМАТИКА И ИНФОРМАТИКА" направленности (профилю)
«Прикладная информатика и искусственный интеллект» ФГБОУ ВО «ЧелГУ»

Версия документа - 1

стр. 9 из 18

Первый экземпляр _____

КОПИЯ № _____

- 1) поиск соответствия парных скобок;
 - 2) поиск лишних скобок.
 - c. Реализовать методы ДДП Find и Insert.
10. Реализация базовых операций ДДП
- a. Переписать функции Find и Insert с помощью _Find.
 - b. Закодировать метод Delete.
 - c. Реализовать рекурсивный деструктор ДДП.
 - d. Дана последовательность цифр. Между каждой парой соседних цифр нужно вставить знак + или * так, чтобы значение полученного выражения было
 - i. максимальным,
 - ii. минимальным.
11. Двоичное дерево поиска
- a. Найти и исправить ошибку в коде Find_next.
 - b. Написать нерекурсивную выгрузку ДДП в вектор в порядке возрастания ключей.
 - c. Для класса ДДП реализовать оператор ==, сравнивающий два дерева на полное совпадение данных и структуры.
 - d. Написать функцию, проверяющую, что данное ДДП является ДДП по полю данных.
12. Матрицы
- a. Реализовать операторы +, -, *, ==, !=, +=, *=, -= для матриц/матриц и матриц/чисел.
 - b. Написать рекурсивные функции вычисления алгебраического дополнения и определителя матрицы.
 - c. Реализовать по правилу Крамера функцию для решения СЛУ, принимающую на вход матрицу и вектор-столбец.
13. Матрицы и код Прюфера
- a. Реализовать оператор быстрого возведения матрицы в степень.
 - b. Реализовать методы матрицы inverse и determinant через метод исключения Гаусса.
 - c. Для заданного списками смежности дерева реализовать функцию получения его кода Прюфера.
 - d. Для заданного кода Прюфера реализовать функцию восстановления дерева.
14. Матрицы и графы
- a. Переписать методы матрицы inverse и determinant через метод исключения Гаусса с выбором ведущего элемента по столбцу.
 - b. Реализовать поиск в ширину и 5 его приложений: поиск длин кратчайших маршрутов, поиск самих маршрутов, определение связности графа, определение количества компонент связности, определение двудольности графа.
 - c. Решить на сайте acm.timus.ru задачу «Число Исенбаева» в разделе «Графовые задачи».
15. Графы и деревья



МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное образовательное
учреждение высшего образования
«Челябинский государственный университет» (ФГБОУ ВО «ЧелГУ»)
Математический факультет
Кафедра вычислительной математики

Фонд оценочных средств по дисциплине "Фундаментальные алгоритмы на C++" по направлению подготовки (специальности) " 01.03.02 ПРИКЛАДНАЯ МАТЕМАТИКА И ИНФОРМАТИКА" направленности (профилю) «Прикладная информатика и искусственный интеллект» ФГБОУ ВО «ЧелГУ»

Версия документа - 1

стр. 10 из 18

Первый экземпляр _____

КОПИЯ № _____

- a. Найти и исправить ошибку в коде функции `code2tree`, реализованной через контейнеры `set` и `multiset`.
- b. Решить на сайте `acm.timus.ru` задачи в разделе «Графовые задачи»:
 - a) «Две команды»;
 - б) «План электрификации».

Семестровое задание

Семестровое задание состоит из двух частей. Студентам выдается по одной задаче из каждой.

Часть I. Структуры данных, классы

1. Создать класс `polynomial`, реализующий многочлен одной переменной. Класс должен содержать набор подходящих конструкторов, перегрузку оператора `[]` для доступа к коэффициенту при соответствующей степени для чтения и для записи, оператор сложения `+` и вычитания `-`, а также метод, дающий по числу a разложение исходного многочлена по степеням $(x-a)$ (т.е. для многочлена $P(x)$ нужно сконструировать многочлен $Q(x)$ такой, что $P(x) = Q(x - a)$).
2. Создать класс `polynomial`, реализующий многочлен одной переменной. Класс должен содержать набор подходящих конструкторов, оператор умножения `*` и *быстрого* возведения в степень `^`, методы взятия производной и первообразной, а также метод вычисления дискриминанта многочлена. <https://ru.wikipedia.org/wiki/Дискриминант>
3. Создать класс `polynomial`, реализующий многочлен одной переменной. Класс должен содержать набор подходящих конструкторов, перегрузку оператора круглые скобки `()` для вычисления значения в точке, оператор деления `/` и деления с остатком `%`. Написать `friend`-функцию для вычисления унитарного наибольшего общего делителя двух переданных многочленов по алгоритму Евклида.
4. Создать класс `polynomial`, реализующий многочлен одной переменной. Класс должен содержать набор подходящих конструкторов, перегрузку операторов `*` для умножения на число справа и слева, оператор деления `/` на число, операторы сдвига `<<` и `>>`, повышающие и понижающие степени слагаемых многочлена, а также метод, вычисляющий оценку для количества корней на отрезке $[a,b]$ по теореме Фурье-Бюдана. Написать метод, определяющий все такие единичные отрезки $[n,n+1]$ ($n \in \mathbb{Z}$), для которых теорема Фурье-Бюдана предсказывает наибольшее количество корней. http://dic.academic.ru/dic.nsf/enc_mathematics/610/БЮДАНА
5. Создать класс `polynomial`, реализующий многочлен одной переменной. Класс должен содержать набор подходящих конструкторов, в том числе конструкторы, создающие мономы (т.е. многочлены вида ax^n). Помимо этого перегрузите операторы `+` (в том числе для сложения с числом) и `*`, а также напишите функцию, принимающую на вход два многочлена и возвращающую *результат* этих двух многочленов. <https://ru.wikipedia.org/wiki/Результант>



МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное образовательное
учреждение высшего образования
«Челябинский государственный университет» (ФГБОУ ВО «ЧелГУ»)
Математический факультет
Кафедра вычислительной математики

Фонд оценочных средств по дисциплине "Фундаментальные алгоритмы на C++" по направлению подготовки (специальности) " 01.03.02 ПРИКЛАДНАЯ МАТЕМАТИКА И ИНФОРМАТИКА" направленности (профилю) «Прикладная информатика и искусственный интеллект» ФГБОУ ВО «ЧелГУ»

Версия документа - 1

стр. 11 из 18

Первый экземпляр _____

КОПИЯ № _____

6. Создать класс `polynomial`, реализующий многочлен одной переменной. Класс должен содержать набор подходящих конструкторов, перегрузку оператора `[]` для доступа к коэффициентам для чтения и для записи, операторы `*`, `+` и `-`, а также статический метод, создающий унитарный многочлен с заданным списком корней, и статический метод, создающий многочлен Чебышева первого рода по его номеру.
https://ru.wikipedia.org/wiki/Многочлены_Чебышёва
7. Создать класс `polynomial`, реализующий многочлен одной переменной. Класс должен содержать набор подходящих конструкторов, перегрузку операторов `+` и `-`, а также перегрузку оператора `<<` для вывода многочлена в поток в *естественном* виде, как в примере: $x^5 - 5.4x^2 - x + 1$. А также перегрузку оператора `>>` для чтения из потока в описанном выше формате.
8. Создать класс `polynomial`, реализующий многочлен одной переменной с целочисленными коэффициентами. Класс должен содержать набор подходящих конструкторов, перегрузку оператора круглые скобки `()` для вычисления значения в заданной точке, а также метод для нахождения *всех* рациональных корней многочлена в виде несократимой дроби, хранимой в виде `pair<int,int>`.
9. Создать класс `polynomial`, реализующий многочлен одной переменной с целочисленными коэффициентами. Класс должен содержать набор подходящих конструкторов, перегрузку оператора умножения `*`, а также метод, проверяющий, является ли многочлен квадратом некоторого другого многочлена.
10. Написать для класса `Matrix` методы, проверяющие, является ли матрица нильпотентной или жордановой. Написать для класса `Matrix` функцию, определяющую являются ли две переданные матрицы жордановыми и подобными.
https://ru.wikipedia.org/wiki/Нильпотентная_матрица
https://ru.wikipedia.org/wiki/Жорданова_матрица
11. Перегрузить для класса `Matrix` функцию `exp`, *эффективно* вычисляющую экспоненту матрицы по указанной формуле. Суммирование вести до тех пор, пока элементы очередного слагаемого не станут меньше 10^{-6} . Написать для класса `Matrix` метод, проверяющий, что матрица записана в канонической форме Вейра.
$$\exp(A) = E + A + \frac{A^2}{2!} + \frac{A^3}{3!} + \dots$$

https://ru.wikipedia.org/wiki/Каноническая_форма_Вейра
12. Написать для класса `Matrix` метод, проверяющий, что заданная матрица является симметричной, а также метод, проверяющий что матрица симметричная и положительно определённая (с помощью критерия Сильвестра). Сложность полученного алгоритма должна быть $O(N^3)$ операций. https://ru.wikipedia.org/wiki/Критерий_Сильвестра
13. Написать для класса `Matrix` метод, вычисляющий дополнение Шура для разбиения на блоки заданного размера. Написать метод для восстановления ортогональной матрицы из матрицы, отличающейся от ортогональной главной диагональю.



МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное образовательное
учреждение высшего образования
«Челябинский государственный университет» (ФГБОУ ВО «ЧелГУ»)
Математический факультет
Кафедра вычислительной математики

Фонд оценочных средств по дисциплине "Фундаментальные алгоритмы на C++" по направлению подготовки (специальности) " 01.03.02 ПРИКЛАДНАЯ МАТЕМАТИКА И ИНФОРМАТИКА" направленности (профилю) «Прикладная информатика и искусственный интеллект» ФГБОУ ВО «ЧелГУ»

Версия документа - 1

стр. 12 из 18

Первый экземпляр _____

КОПИЯ № _____

https://ru.wikipedia.org/wiki/Дополнение_Шура
https://ru.wikipedia.org/wiki/Ортогональная_матрица

14. Написать для класса `BinarySearchTree` две версии метода `count_less_than(const KeyType &key)`, подсчитывающего количество ключей в дереве, меньших `key`:
а) с помощью метода `PrevKey`; б) алгоритмом с линейной (относительно количества искомых элементов и высоты дерева) сложностью.
15. Написать для класса `BinarySearchTree` две версии метода `symmetric_view()`, возвращающего двоичное дерево, симметричное исходному относительно вертикальной прямой, проходящей через корень дерева:
а) с помощью рекурсивного алгоритма; б) с помощью нерекурсивного алгоритма.
Замечание. В результате получится двоичное дерево поиска с противоположным порядком расположения ключей в поддеревьях.
16. Написать для класса `BinarySearchTree` две версии метода `is_subtree(const BinarySearchTree &T)`, проверяющего, что `T` является поддеревом `*this`:
а) с помощью рекурсивного алгоритма; б) с помощью нерекурсивного алгоритма.
17. Написать для класса `BinarySearchTree` две версии метода копирования `assign(const BinarySearchTree &T)`, создающего в текущем объекте копию дерева `T` (должны быть идентичны содержащиеся данные и структура связей между узлами):
а) с помощью рекурсивного алгоритма; б) с помощью нерекурсивного алгоритма.
18. Написать для класса `BinarySearchTree` две версии friend-функции `mutual_key(const BinarySearchTree &T1, const BinarySearchTree &T2, KeyType &key)`, отыскивающую в переменной `key` общий ключ (если он есть) деревьев `T1` и `T2`:
а) с помощью рекурсивного алгоритма; б) с помощью нерекурсивного алгоритма.
19. Написать для класса `BinarySearchTree` метод `min_childfree_key()`, отыскивающий в дереве минимальный ключ, узел которого не имеет потомков. Написать для класса `BinarySearchTree` метод `min_childfull_key()`, отыскивающий в дереве минимальный ключ, узел которого имеет обоих потомков.

Часть II. Прикладные алгоритмы

20. Написать шаблонную функцию вида `is_substack(stack &source, stack &sample)`, проверяющую, что элементы стека `sample` встречаются в каком-то месте стека `source`. При реализации разрешается использовать только стеки того-же типа и простые переменные (никаких векторов, массивов и т.п.).
21. Написать функцию, генерирующую для заданного количества вершин n такой полный неориентированный взвешенный граф, что при поиске алгоритмом Дейкстры кратчайших путей из 0-ой вершины на каждом шаге изменяются *все* глубины вершин, для которых кратчайший путь ещё в процессе определения. А заодно и алгоритм Дейкстры реализуйте.



МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное образовательное
учреждение высшего образования
«Челябинский государственный университет» (ФГБОУ ВО «ЧелГУ»)
Математический факультет
Кафедра вычислительной математики

Фонд оценочных средств по дисциплине "Фундаментальные алгоритмы на C++" по направлению подготовки (специальности) " 01.03.02 ПРИКЛАДНАЯ МАТЕМАТИКА И ИНФОРМАТИКА" направленности (профилю) «Прикладная информатика и искусственный интеллект» ФГБОУ ВО «ЧелГУ»

Версия документа - 1

стр. 13 из 18

Первый экземпляр _____

КОПИЯ № _____

22. Дан неориентированный граф, в каждой вершине которого записано натуральное число. Напишите функцию, находящую маршрут из одной вершины в другую такой, чтобы сумма чисел в этом маршруте оказалась наименьшей.
23. Напишите функцию, находящую в заданном ориентированном графе цикл наименьшей длины.
24. Написать функцию, которая для заданного неориентированного взвешенного графа и указанной вершины находит все вершины графа, в которые можно попасть из указанной, двигаясь по рёбрам с увеличивающимся весом. Вес рёбер - натуральное число, меньше 10^9 .
25. Написать функцию, которая по массиву вершин замкнутой ломаной, перечисленных в порядке обхода, определяет количество частей, на которые разбивается плоскость этой ломаной. Каждая вершина ломаной лежит только на двух рёбрах, исходящих из этой вершины.
26. Написать пару функций, которые для заданного множества различных точек в пространстве находят 1) количество различных проекций этих точек на плоскость в трёхмерном пространстве, заданную своим каноническим уравнением; 2) каноническое уравнение плоскости, для которой количество проекций исходных точек будет минимальным.
27. Написать функцию, которая по заданному множеству отрезков на плоскости определяет, является ли оно связным.
28. Написать функцию, которая определяет можно ли заданное множество точек на плоскости разделить на две части, чтобы точки в каждой из частей образовывали выпуклый многоугольник. Количество точек не более 20. Никакие три точки не лежат на одной прямой.
29. Написать функцию, которая по заданному множеству точек определяет, сколько различных выпуклых многоугольников с вершинами в некоторых из этих точек можно нарисовать. Никакие три точки не лежат на одной прямой. Количество точек не более 20.
30. Написать функцию, которая по массиву вершин многоугольника перечисленных в порядке обхода определяет, является ли он выпуклым, а если нет, то находит все вершины, лежащие строго внутри выпуклой оболочки.
31. Напишите функцию построения кода Прюфера вида `tree2code(BinarySearchTree &T)`, если известно, что ключи дерева T - это числа $0, 1, 2, \dots, n$.
32. Напишите функцию вида `paths_with_sum(const BinarySearchTree<int,double> &T, double sum)`, отыскивающую все пути от корня дерева T до его листьев, что сумма данных в этих путях одна и та же и равна sum.
33. Добавить в алгоритм вычисления символьных выражений операции сравнения `<` и `>`, операции битового сдвига `<<` и `>>`, а также обработку унарного минуса. Например, `a*(-b+(a>b+c)<<3)*d`.
34. Добавить в алгоритм вычисления символьных выражений операцию вычисления целой части числа через `[]`. Например, `a+e/x-[(3i-j)/3.1415]`.



МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное образовательное
учреждение высшего образования
«Челябинский государственный университет» (ФГБОУ ВО «ЧелГУ»)
Математический факультет
Кафедра вычислительной математики

Фонд оценочных средств по дисциплине "Фундаментальные алгоритмы на C++" по направлению подготовки
(специальности) " 01.03.02 ПРИКЛАДНАЯ МАТЕМАТИКА И ИНФОРМАТИКА" направленности (профилю)
«Прикладная информатика и искусственный интеллект» ФГБОУ ВО «ЧелГУ»

Версия документа - 1	стр. 14 из 18	Первый экземпляр _____	КОПИЯ № _____
----------------------	---------------	------------------------	---------------

35. Добавить в алгоритм вычисления символьных выражений операцию gcd, которая принимает на вход два параметра, округляет их до ближайшего целого и вычисляет наибольший общий делитель. Например, gcd(2.2,6.3) равен 2.
36. Добавить в алгоритм вычисления символьных выражений операцию вычисления факториала и отрицания. Например, !a + (c - d*e)!+c/e.
37. Добавить в алгоритм вычисления символьных выражений обработку сравнения >=, <=, == и !=. Например, a*((c==d)*b +c/d)>=c+b/a.
38. Написать функцию, которая по последовательности токенов, содержащей постфиксную запись, воссоздаёт исходное выражение в естественной форме записи. Предполагается, что все токены-функции – унарные, а операторы – только основные бинарные +, -, *, / и возведение в степень ^.

Перечень вопросов для экзамена

1. Шаблоны C++: определение, параметры шаблона, шаблонные функции и классы, примеры использования.
2. Классы C++: перегрузка стандартных операций для объектов классов с примерами.
3. Язык C++: циклы for, while, do while, операторы ветвления. Использование циклов для перечисления содержимого контейнеров.
4. Алгоритмы нахождения площади треугольника и многоугольника. Алгоритм Евклида, формула Пика и нахождение количества целых точек внутри многоугольника.
5. Универсальные алгоритмы STL: работа с итераторами, немодифицирующие алгоритмы, примеры использования.
6. Универсальные алгоритмы STL: сортировка и поиск min/max элементов, использование предикатов сравнения.
7. Универсальные алгоритмы STL: работа с итераторами, модифицирующие алгоритмы, примеры использования.
8. Контейнеры STL: вектор и ассоциативный массив, операции с ними, примеры использования
9. Контейнеры STL: сет, мультисет и операции с ними, примеры использования в алгоритмах кода Прюфера
10. Универсальные алгоритмы STL: алгоритмы проверки условия, алгоритмы поиска, алгоритмы модификации, примеры использования
11. Лямбда-выражения: особенности синтаксиса и использование
12. Класс Matrix: разработка интерфейса класса и реализация операторов
13. Класс Matrix: быстрое возведение в степень
14. Класс Matrix: нахождение обратной матрицы и определителя
15. Назначение и использование механизма RVO
16. Стеки: реализация на основе массива



МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное образовательное
учреждение высшего образования
«Челябинский государственный университет» (ФГБОУ ВО «ЧелГУ»)
Математический факультет
Кафедра вычислительной математики

Фонд оценочных средств по дисциплине "Фундаментальные алгоритмы на С++" по направлению подготовки
(специальности) " 01.03.02 ПРИКЛАДНАЯ МАТЕМАТИКА И ИНФОРМАТИКА" направленности (профилю)
«Прикладная информатика и искусственный интеллект» ФГБОУ ВО «ЧелГУ»

Версия документа - 1

стр. 15 из 18

Первый экземпляр _____

КОПИЯ № _____

17. Стеки: реализация на основе связного списка
18. Стеки: реализация идеи быстрой сортировки на стеках. Особенности функционирования в зависимости от используемой реализации контейнера.
19. Стеки: реализация идеи сортировки слиянием на стеках. Особенности функционирования в зависимости от используемой реализации контейнера.
20. Стеки: алгоритм построения постфиксной записи, типы данных и прототипы функций для его реализации.
21. Стеки: алгоритм Грэхема.
22. Стеки: вычисление выражения, записанного в постфиксной записи, типы данных и прототипы функций для его реализации.
23. Деревья/графы и их представление/хранение. Создание списков смежности, добавление и удаление ребра.
24. Двоичное дерево поиска: определение, операции поиска и вставки ключа, поиск следующего по значению.
25. Двоичное дерево поиска: определение, удаления ключа и нерекурсивное удаление поддеревя.
26. Код Прюфера. Построение и расшифровка через векторы, сложность алгоритма.
27. Поиск в ширину в графе: реализация, сложность алгоритма, получение оптимального маршрута от заданной вершины.
28. Приложения поиска в ширину: нахождение количества компонент связности и определение двудольности графа
29. Поиск кратчайших путей в графе: алгоритм Форда-Беллмана, реализация и сложность.
30. Поиск кратчайших путей в графе: алгоритм Флойда, реализация и сложность.
31. Поиск кратчайших путей в графе: алгоритм Дейкстры, реализация и сложность.
32. Двоичная куча. Операция вставки элемента и извлечения минимума.
33. Улучшенный алгоритм Дейкстры.

4. ПОРЯДОК ПРОВЕДЕНИЯ И КРИТЕРИИ ОЦЕНИВАНИЯ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ

4.1. Порядок проведения промежуточной аттестации

Балльно-рейтинговая система оценки знаний студента по дисциплине выстраивается на основе балловой оценки различных форм деятельности студентов. Для оценки экзамена суммируются баллы семестра и экзамена.

В течение семестра студентам выдается 15 домашних заданий, которые осуществляют срез знаний по основным понятиям, определениям и задачам. Оценка за домашние задания варьируется от 3 до 5 баллов (в зависимости от количества пунктов).

Максимальный балл в 5-м семестре за домашние задания – 60.

Также студенты получают семестровое задание, состоящее из двух задач из



МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное образовательное
учреждение высшего образования
«Челябинский государственный университет» (ФГБОУ ВО «ЧелГУ»)
Математический факультет
Кафедра вычислительной математики

Фонд оценочных средств по дисциплине "Фундаментальные алгоритмы на С++" по направлению подготовки (специальности) " 01.03.02 ПРИКЛАДНАЯ МАТЕМАТИКА И ИНФОРМАТИКА" направленности (профилю) «Прикладная информатика и искусственный интеллект» ФГБОУ ВО «ЧелГУ»

Версия документа - 1

стр. 16 из 18

Первый экземпляр _____

КОПИЯ № _____

разделов «Структуры данных» и «Фундаментальные алгоритмы». Выполнение каждого из этих заданий оценивается максимум в 15 баллов.

Максимальный балл за семестровое задание – 30.

Проведение экзамена:

На экзамене студенту предлагается письменно ответить на два теоретических вопроса, баллы за которые суммируются с баллами за работу в течение семестра.

Структура экзаменационного билета

1. Теоретический вопрос – 10 баллов
2. Теоретический вопрос – 10 баллов

При оценке знаний учитывается также выполнение домашних и семестрового заданий.

3. Домашние задания в течение семестра – 60 баллов
4. Семестровое задание в течение семестра – 30 баллов

Итоговое максимальное количество баллов – $10 \cdot 2 + 60 + 30 = 110$ баллов

Критерий оценивания результатов экзамена:

- 60 – 75 баллов – выставляется оценка “удовлетворительно”
- 76 – 89 баллов – выставляется оценка “хорошо”
- 90 – 110 баллов – выставляется оценка “отлично”

4.2. Критерии оценивания промежуточной аттестации по видам оценочных средств.

4.2.1 Критерии оценивания теоретического вопроса

Максимальный балл за ответ на теоретический вопрос — 10 баллов.

Отлично/ зачтено/ 9-10 баллов	Хорошо/ зачтено/ 7-8 баллов	Удовлетворительно/ зачтено/ 5-6 баллов	Неудовлетворительно/ незачтено/ 0-4 балла
Высокий уровень освоения проверяемых компетенций	Средний уровень освоения проверяемых компетенций	Базовый уровень освоения проверяемых компетенций	Недостаточный уровень освоения проверяемых компетенций
Обучающийся отлично знает материал, умеет анализировать	Обучающийся хорошо знает материал, понимает, как	Обучающийся знаком с материалом, владеет достаточным для высказывания	Обучающийся не знает основных положений вопроса, не ориентируется в



МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное образовательное
учреждение высшего образования
«Челябинский государственный университет» (ФГБОУ ВО «ЧелГУ»)
Математический факультет
Кафедра вычислительной математики

Фонд оценочных средств по дисциплине "Фундаментальные алгоритмы на C++" по направлению подготовки
(специальности) " 01.03.02 ПРИКЛАДНАЯ МАТЕМАТИКА И ИНФОРМАТИКА" направленности (профилю)
«Прикладная информатика и искусственный интеллект» ФГБОУ ВО «ЧелГУ»

Версия документа - 1

стр. 17 из 18

Первый экземпляр _____

КОПИЯ № _____

проблему и аргументировано изложить свою точку зрения, владеет достаточным для высказывания теоретическим запасом, грамотно изъясняется на профессиональном языке с использованием точных терминов и названий. Обучающийся практически не допускает ошибок.	анализировать проблему, аргументация может содержать мелкие пробелы, владеет достаточным для высказывания теоретическим запасом, хорошо изъясняется на профессиональном языке. Обучающийся допускает незначительные ошибки.	теоретическим запасом. Обучающийся допускает некоторые ошибки, плохо оперирует понятиями и фактами.	основных понятиях, излагает материал с трудом, с грубыми теоретическими ошибками, либо отказывается от ответов на вопросы.
---	---	---	--

4.3. Результаты промежуточной аттестации и уровни сформированности компетенций

Особенности проведения процедуры оценивания результатов обучения инвалидов и лиц с ограниченными возможностями здоровья обозначены в рабочей программе дисциплины (модуля).

Уровни сформированности компетенций определяется следующим образом:

1. Высокий уровень сформированности компетенций соответствует оценке отлично:
 - предполагает формирование компетенций на высоком уровне, готовность к самостоятельной профессиональной деятельности: формируются навыки составления и анализа алгоритмов, использования современных технологий программирования
 - студент способен аргументировать собственную точку зрения по дискуссионным вопросам дисциплины, решать ситуационные задачи, критически оценивать информацию о реализации тех или иных аспектов вычислительных и информационных алгоритмов.
2. Средний уровень соответствует оценке хорошо:
 - предполагает формирование компетенций на более высоком уровне: формируется комплексное знание особенностей применения и понимания тех или иных алго-



МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное образовательное
учреждение высшего образования
«Челябинский государственный университет» (ФГБОУ ВО «ЧелГУ»)
Математический факультет
Кафедра вычислительной математики

Фонд оценочных средств по дисциплине "Фундаментальные алгоритмы на С++" по направлению подготовки
(специальности) " 01.03.02 ПРИКЛАДНАЯ МАТЕМАТИКА И ИНФОРМАТИКА" направленности (профилю)
«Прикладная информатика и искусственный интеллект» ФГБОУ ВО «ЧелГУ»

Версия документа - 1

стр. 18 из 18

Первый экземпляр _____

КОПИЯ № _____

- ритмов и структур данных в прикладных задачах, умение сбора, анализа и обработки данных, необходимых для решения конкретных алгоритмических задач;
- студент способен давать развернутые ответы на теоретические вопросы дисциплины на уровне не ниже оценки «удовлетворительно».
3. Базовый уровень соответствует оценке удовлетворительно:
- предполагает формирование компетенций на начальном уровне: знание основных структур данных и алгоритмов с применением этих структур;
4. Низкий уровень соответствует оценке неудовлетворительно.

