

**БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ФАКУЛЬТЕТ РАДИОФИЗИКИ И КОМПЬЮТЕРНЫХ ТЕХНОЛОГИЙ
КАФЕДРА ИНФОРМАТИКИ И КОМПЬЮТЕРНЫХ СИСТЕМ**

**Н.В ЛЕВКОВИЧ
Н. В. СЕРИКОВА**

**ЗАДАНИЯ ПО КУРСУ
«ПРОГРАММИРОВАНИЕ»**

ВАРИАНТ В

**2020
МИНСК**

ОГЛАВЛЕНИЕ

2 семестр	3
5. Функции	4
5.4. Возврат ссылок	4
5.5. Вычисление корня уравнения. Передача имени функции в качестве параметра. Аргументы по умолчанию	6
5.6. Вычисление интеграла. Передача имени функции в качестве параметра	8
5.7. Сортировка массивов	10
5.8. Сортировка слиянием	10
5.9. Перебор с возвратом	11
6. Файлы	12
6.1. Строки string	12
6.2. Файлы чисел	13
6.3. Файлы записей	14
6.4. Использование структур с битовыми полями	15
7. Динамические структуры данных	16
7.1. Динамическое выделение памяти для одномерных массивов	16
7.2. Динамическое выделение памяти для матриц	17
7.3. Линейный список структур	18
7.4. Линейные списки как динамическая структура данных	19
7.5. Линейные списки как массивы данных	19
7.6. Двусвязные списки как динамические структуры данных	20
7.7. Бинарное дерево поиска	21
7.8. Графы	22
8. Классы	23
8.1. Класс «Вектор»	23
8.2. Инкапсуляция	24
8.3. Класс «Многоразрядное число»	26
8.4. Класс «Линейный список»	28
8.5. Класс «Двусвязный список»	28
8.6. Класс «Бинарное дерево»	29
8.7. Класс «Граф»	30
8.8. Класс «Матрица»	31
9. Библиотека шаблонов STL	32
9.1. Векторы	32
9.2. Списки	32
9.3. Вычисление выражений с использованием обратной польской записи	32
9.4. Ассоциативные списки	33
10. Наследование. Полиморфизм	34
10.1. Наследование	34
10.2. Полиморфизм. Виртуальные методы	47
10.3. «Фабрика класса»	47
10.4. Полиморфизм. Виртуальные методы	47

2 СЕМЕСТР

29 занятий

оценка	количество задач
7	22
8	30
9	35
10	37

№	тема	№ задач			
		7	8	9	10
1	5. Функции	5.1			
2		5.2			
3		5.3			
4		5.4	5.6		
5		5.5		5.8	
6		5.7			5.9
7	6. Структуры. Файлы	6.1			
8		6.2			
9		6.3			
10			6.4		
11	7. Динамические структуры данных	7.1			
12		7.2	7.5		
13		7.3			
14		7.4	7.7		
15		7.6		7.8	
16	8. Классы	8.1			
17		8.2			
18		8.4			
19		8.5	8.3		
20			8.6		
21				8.7	
22				8.8	
23	9. Библиотека STL	9.1			
24		9.2			
25				9.3	
26					9.4
27	10. Наследование	10.1			
28		10.2	10.3		
29			10.4		
30	Зачет				

5. ФУНКЦИИ

5.4. ВОЗВРАТ ССЫЛОК

Напишите функцию, возвращающую ссылку.

1. Напишите функцию, возвращающую ссылку на минимальное число, встречающееся в заданном массиве произвольного размера (аргумент функции) ровно один раз. Если такого числа нет, то вернуть ссылку на любое из минимальных чисел массива. Замените этот элемент нулевым значением.

2. Напишите функцию, возвращающую ссылку на минимальное число, встречающееся в заданном массиве произвольного размера (аргумент функции) больше одного раза. Если такого числа нет, то вернуть ссылку на минимальное число в массиве. Замените этот элемент значением k .

3. Напишите функцию, возвращающую ссылку на максимальное число, встречающееся в заданном массиве произвольного размера (аргумент функции) ровно 2 раза. Если такого числа нет, то вернуть ссылку на любое из максимальных чисел массива. Замените этот элемент значением k .

4. Напишите функцию, возвращающую ссылку на число, встречающееся в заданном массиве произвольного размера (аргумент функции) максимальное количество раз. Если таких чисел несколько, то выбрать минимальное из них. Заменить этот элемент значением k .

5. Напишите функцию, возвращающую ссылку на максимальное число, встречающееся в заданном массиве произвольного размера (аргумент функции) более двух раз. Если такого числа нет, то вернуть ссылку на любое из максимальных чисел массива. Замените этот элемент значением k .

6. Напишите функцию, возвращающую ссылку на максимальное число, встречающееся в заданном массиве произвольного размера (аргумент функции), являющееся числом Фибоначчи. Если такого числа нет, то вернуть ссылку на любое из максимальных чисел массива. Заменить значение этого элемента нулевым значением.

7. Напишите функцию, возвращающую ссылку на максимальное число, встречающееся в заданном массиве произвольного размера (аргумент функции), являющееся степенью числа n . Если такого числа нет, то вернуть ссылку на любое из максимальных чисел массива. Заменить значение этого элемента нулевым значением.

8. Напишите функцию, возвращающую ссылку на максимальное число, встречающееся в заданном массиве произвольного размера (аргумент функции) ровно 1 раз. Если такого числа нет, то вернуть ссылку на любое из максимальных чисел массива. Заменить значение этого элемента нулевым значением.

9. Напишите функцию, возвращающую ссылку на максимальное число, встречающееся в заданном массиве произвольного размера (аргумент функции)

больше одного раза. Если такого числа нет, то вернуть ссылку на максимальное число массива. Замените значение этого элемента нулевым значением.

10. Напишите функцию, возвращающую ссылку на число, встречающееся в заданном массиве произвольного размера (аргумент функции) минимальное количество раз. Если таких чисел несколько, то выбрать максимальное из них. Замените этот элемент значением k .

11. Напишите функцию, возвращающую ссылку на минимальное число, встречающееся в заданном массиве произвольного размера (аргумент функции), являющееся полным квадратом. Если такого числа нет, то вернуть ссылку на любое из максимальных чисел массива. Замените этот элемент значением k .

12. Напишите функцию, возвращающую ссылку на число, встречающееся в заданном массиве произвольного размера (аргумент функции) чаще остальных. Если таких чисел несколько, то выбрать минимальное из них. Замените этот элемент значением k .

5.5. ВЫЧИСЛЕНИЕ КОРНЯ УРАВНЕНИЯ. ПЕРЕДАЧА ИМЕНИ ФУНКЦИИ В КАЧЕСТВЕ ПАРАМЕТРА. АРГУМЕНТЫ ПО УМОЛЧАНИЮ

Вычислить корень уравнения $f(x) = 0$ на отрезке $[a; b]$ с точностью $\varepsilon = 10^{-6}$, используя заданный метод ($M = 1$ – метод половинного деления, $M = 2$ – метод касательных, $M = 3$ – метод хорд) для заданных функций. Вычисление корня уравнения оформить в виде функции с функциональным параметром, параметры a, b, ε, s – в виде аргументов по умолчанию.

Результат представить в виде таблицы (s – значение параметра, x – вычисленный корень уравнения, $f(x)$ – значение функции в найденной точке x , k_iter – количество итераций цикла для получения корня с заданной точностью):

S	X	$F(x)$	k_iter
S
$S+\Delta s$
...

1. а) $f(x) = x^2 - 3$ $a = 1; \quad b = 3;$
 б) $f(x) = e^{-sx} - 2 + x^2$ $a = 0; \quad b = 1,5; s \in [0,1; 1,3]; \Delta s = 0,3. M=1.$
2. а) $f(x) = x^3 - 3$ $a = 1; \quad b = 4;$
 б) $f(x) = \sqrt[5]{x} - 2 \cos^2(\pi x / 2)$ $a = 0; \quad b = 4,5; s \in [0,5; 2], \Delta s = 0,5. M=1.$
3. а) $f(x) = (x-1)^2 - 3$ $a = 1; \quad b = 4;$
 б) $f(x) = e^{(x-s)} - \sqrt{x+1}$ $a = 0; \quad b = 2; s \in [0,3; 0,7]; \Delta s = 0,1. M=1.$
4. а) $f(x) = (x-1)^2 - 3$ $a = -2; \quad b = 1;$
 б) $f(x) = \cos^2(x) - \sqrt[5]{x}$ $a = 0; \quad b = 1; s \in [1,95; 2], \Delta s = 0,01. M=1.$
5. а) $f(x) = (x-1)^2 - 5$ $a = -3; \quad b = 0;$
 б) $f(x) = x^2 - \sin(5x^s)$ $a = 0,5; \quad b = 0,8; s \in [0,7; 1,6], \Delta s = 0,3; M=2.$
6. $f(x) = (x-1)^3 - 8$ $a = 1; \quad b = 4;$
 б) $f(x) = s \cos^2(\pi x) - \sqrt{x}$ $a = 0; \quad b = 1,5; s \in [0,95; 1,2], \Delta s = 0,05; M=2.$
7. а) $f(x) = (x+3)^3 - 8$ $a = -2; \quad b = 1;$
 б) $f(x) = \cos(\pi x) - x^s$ $a = 0; \quad b = 2; s \in [2,8; 3,2], \Delta s = 0,1; M=2.$
8. а) $f(x) = (x-1)^3 - 1$ $a = 0; \quad b = 3;$
 б) $f(x) = sx - \cos^2(\pi x)$ $a = -1; \quad b = 0,7; s \in [1; 3], \Delta s = 1; M=1.$

9. а) $f(x) = (x-1)^2 - 5$ $a = 2; \quad b = 4;$
 б) $f(x) = (x-s)^2 - e^{-x}$ $a = -1,5; \quad b = 2; \quad s \in [0,7; 1,6], \quad \Delta s = 0,3; \quad M=3.$
10. а) $f(x) = (x+1)^2 - 5$ $a = 0; \quad b = 2;$
 б) $f(x) = x^2 - e^x - 1,5s$ $a = -1,5; \quad b = 1; \quad s \in [0,9; 1,1], \quad \Delta s = 0,05; \quad M=3.$
11. а) $f(x) = (x+1)^2 - 4$ $a = 0; \quad b = 3;$
 б) $f(x) = \cos^2(\pi x) + x^2 - 1,5s$ $a = 0; \quad b = 2; \quad s \in [0,9; 1,2], \quad \Delta s = 0,02; \quad M=3.$
12. а) $f(x) = (x+1)^2 - 9$ $a = 1; \quad b = 4;$
 б) $f(x) = \cos^2(\pi x) - e^{x^s} + 1$ $a = 0; \quad b = 1; \quad s \in [0,96; 1,02], \quad \Delta s = 0,02; \quad M=3.$

5.6. ВЫЧИСЛЕНИЕ ИНТЕГРАЛА. ПЕРЕДАЧА ИМЕНИ ФУНКЦИИ В КАЧЕСТВЕ ПАРАМЕТРА

Вычислить следующие интегралы заданным методом ($M = 1$ – метод левых прямоугольников, $M = 2$ – метод правых прямоугольников, $M = 3$ – метод средних прямоугольников, $M = 4$ – метод трапеции, $M = 5$ – метод Симпсона.), воспользовавшись критерием двойного пересчета, с точностью $\varepsilon = 10^{-6}$.

Значения параметра s : $s_i \in [s1; s2]$, $s_i = s1 + i\Delta s$, $i = 0, 1, \dots$.

Значения параметра t : $t_i \in [t1; t2]$, $t_i = t1 + i\Delta t$, $i = 0, 1, \dots$.

Вычисление значения функции в точке оформить в виде функции, вычисление интеграла – в виде функции с параметром функционального типа для задания функции. Процедуры оформить в виде отдельного файла.

Результаты представить в виде таблицы (s, t – соответствующие значения параметров, Int – вычисленное значение интеграла, k_iter – количество пересчетов значений интеграла для получения заданной точности):

s	t	Int	k_iter
$s1$	$t1$
$s1$	$t1+\Delta t$
...
$s1+\Delta s$	$t1$
...

$$1. \int_a^b t \frac{\arctg(\sqrt[5]{x})}{x^2 + s \sin(x)} dx$$

$$2. \int_a^b \sqrt[s]{t + x^3} dx$$

$$3. \int_a^b \frac{t \sqrt{1+x^2}}{e^{\sin(x)+s}} dx$$

$$4. \int_a^b s \frac{\cos^3(x^2)}{t \sqrt{x}} dx$$

$$5. \int_a^b s \frac{x^2}{\lg(x^t) + \cos^2(\sqrt[3]{x})} dx$$

$$6. \int_a^b \frac{1 + x^t + x^2}{\sqrt{x^3 + 1}} (1 - \sin(sx)) dx$$

$$7. \int_a^b \frac{\lg(x^2) + \sqrt{x}}{t \lg(x+s)} dx$$

$$8. \int_a^b \frac{t}{\sqrt{x^3 + 1} + s} dx$$

$$9. \int_a^b t \frac{3x^s}{\sqrt{1+x^3}} dx$$

$$10. \int_a^b \frac{\sin^2(x^t)}{\sqrt{1+x^3}} dx$$

$$11. \int_a^b t \frac{1 + (sx)^2}{1 + x^3} dx$$

$$12. \int_a^b t \frac{e^{0,5x}}{\sqrt{x+s}} dx$$

Таблица значений параметров

№ зада- ния	s1	s2	Δs	t1	t2	Δt	a	b	M
1	0	0,5	0,1	0,6	1,8	0,4	0,1	1,2	1
2	1,5	2,5	0,5	0	4	1	0,81	1,762	1
3	0	2	0,5	2	5	1	-1	1	1
4	0	5	1	2	3	0,2	0,1	0,7	2
5	1	10	3	1	3	1	2,63	3,1	2
6	0	0,5	0,1	0,5	2,5	0,5	1	2	3
7	1	4	0,5	0,4	1,2	0,2	0,4	1,2	3
8	0	0,5	0,1	1	3	0,5	1,3	2,621	4
9	1	4	1	0,2	1,2	0,2	0	1,075	4
10	0	1	0,2	1	4	1	s	1,234	5
11	1	5	1	0,5	2,5	0,5	3	4,254	5
12	0,5	2	0,5	0,5	4	0,5	0	1,047	5

5.7. СОРТИРОВКА МАССИВОВ

Дан массив чисел произвольной длины. Отсортировать массив заданными методами:

1. сортировкой вставками и обменом;
2. сортировкой выбором и вставками;
3. сортировкой выбором и обменом;
4. сортировкой простыми и бинарными вставками;
5. быстрой сортировкой и сортировкой вставками;
6. быстрой сортировкой и сортировкой обменами;
7. быстрой сортировкой и сортировкой выбором;
8. быстрой сортировкой и сортировкой бинарными вставками;
9. сортировкой бинарными вставками и сортировкой обменами;
10. сортировкой бинарными вставками и сортировкой выбором;
11. сортировкой вставками и быстрой сортировкой;
12. сортировкой простыми и бинарными вставками.

Для тестирования программы заполнять массив значениями тремя способами: по возрастанию, по убыванию, случайным образом.

Каждый метод сортировки, каждый способ заполнения массивов оформить отдельными функциями. Функции оформить в виде отдельного файла.

Для каждого метода сортировки определить число сравнений и перемещений (перестановок с одного места на другое) элементов в процессе выполнения программы.

Сравнить экспериментальные результаты с известными теоретическими оценками этих показателей для заданных методов сортировки.

Обеспечить перегрузку и шаблоны необходимых функций для выполнения задания с типами элементов массивов `char`, `int`, `float`, `double`.

5.8. СОРТИРОВКА СЛИЯНИЕМ

Дан массив чисел произвольной длины. Отсортировать массив, используя сортировку слияниями.

Создать шаблоны функций для сортировок, заполнения массивов, вывода результатов. Сравнение элементов массива оформить отдельной функцией. Перегрузить функцию сравнения элементов для массива `C-строк`.

Выполнить сортировки массивов для данных разных типов: целых, вещественных, символьных, `C-строк`, используя написанные функции.

5.9. ПЕРЕБОР С ВОЗВРАТОМ

Написать функции для выполнения следующих заданий.

1. Получить все расстановки восьми ладей на шахматной доске, при которых ни одна ладья не угрожает другой.
2. Задача о восьми слонах: на шахматной доске расставить восемь слонов так, чтобы каждое поле находилось под ударом одного из них.
3. Дано натуральное число m . Получить m расстановок восьми ферзей на шахматной доске, при которых ни один из ферзей не угрожает другому. Если m больше общего числа таких расстановок, то получить все расстановки.
4. Найти расстановку пяти ферзей на шахматной доске, при которой каждое поле будет находиться под ударом одного из них.
5. На одной из клеток шахматной доски стоит конь. Требуется выполнить обход конем шахматной доски. Ни одну из клеток конь не может проходить дважды, но каждой клетке он обязан побывать. Выдать сообщение, если обхода не существует.
6. Найти расстановку двенадцати коней на шахматной доске, при которой каждое свободное поле будет находиться под ударом как минимум одного из них.

6. ФАЙЛЫ

6.1. СТРОКИ STRING

Выполнить задания 4.1, 4.2, 4.3, используя string-строки для представления данных. Задания выполнить через функции.

6.2. ФАЙЛЫ ЧИСЕЛ

Выполнить задания для **текстового и бинарного** файлов. Задания выполнять через функции. **Размер файлов $\leq 64\text{GiB}$.**

1. В файлах f и g содержатся вещественные числа, упорядоченные по возрастанию. Получить за один проход файл чисел, упорядоченный по возрастанию, путем слияния файлов f и g .

2. Компоненты файла f – целые числа. Получить файл g из чисел исходного файла, в котором записаны сначала все положительные числа, затем все отрицательные.

3. Компоненты файла f – целые числа, четных чисел столько же, сколько нечетных. Получить файл g из чисел исходного файла, в котором не было бы двух соседних чисел одинаковой четности.

4. В файлах f и g содержатся вещественные числа, упорядоченные по убыванию. Получить за один проход файл чисел, упорядоченный по убыванию, путем слияния файлов f и g .

5. Компоненты файла f – целые числа, положительных чисел столько же, сколько отрицательных. Получить файл g из чисел исходного файла, в котором не было бы двух соседних чисел с одинаковым знаком.

6. Компоненты файла f – целые числа, причем положительных чисел столько же, сколько отрицательных. Получить файл g из чисел исходного файла, в котором записаны 2 положительных числа, затем 2 отрицательных и т. д.

7. Компоненты файла f – целые числа, причем четных чисел столько же, сколько нечетных. Получить файл g из чисел исходного файла, в котором записаны 2 четных числа, затем 2 нечетных т. д.

8. В файлах f и g содержатся вещественные числа, упорядоченные по возрастанию. Получить за один проход файл чисел, встречающихся в обоих исходных файлах без повторений.

9. Компоненты файла f – целые числа. Получить файл g из чисел исходного файла, в котором записаны сначала все четные числа, затем все нечетные числа.

10. Компоненты файла f – положительные вещественные числа. Получить файл g из чисел исходного файла, в котором записаны сначала все числа больше 1, затем остальные числа.

11. В файлах f и g содержатся вещественные числа, упорядоченные по возрастанию. Получить за один проход файл чисел, встречающихся в файле f и не встречающихся в файле g .

12. Компоненты файла f – положительные вещественные числа, причем чисел $> \sqrt{2}$ столько же, сколько и меньших этого значения. Получить файл g из чисел исходного файла, в котором не было бы двух соседних чисел $> \sqrt{2}$.

6.3. ФАЙЛЫ ЗАПИСЕЙ

Написать программу для работы с базой данных, содержащей записи со сведениями о студентах: ФИО, возраст, пол, курс, успеваемость, в которой должны выполняться следующие действия:

- *создание файла*
- *просмотр файла*

*Выполнить задания для **текстового и бинарного** файлов.*

1. По исходному файлу определить ФИО самого младшего студента на каждом курсе. Сформировать файл из этих студентов.
2. По исходному файлу определить ФИО самого старшего студента на каждом курсе. Сформировать файл из этих студентов.
3. По исходному файлу определить всех студентов, ФИО которых начинается на заданную букву. Сформировать файл из этих студентов.
4. По исходному списку определить всех студентов, возраст которых превышает заданный. Сформировать файл из этих студентов.
5. По исходному файлу определить всех студентов n курса. Сформировать файл из этих студентов.
6. По исходному файлу определить всех отличников. Сформировать файл из этих студентов.
7. По исходному файлу определить всех отличников n курса. Сформировать файл из этих студентов.
8. По исходному файлу определить всех неуспевающих студентов. Сформировать файл из этих студентов.
9. По исходному файлу определить всех неуспевающих студентов n курса. Сформировать файл из этих студентов.
10. По исходному файлу определить студентов, имеющих средний бал успеваемости выше общего среднего бала. Сформировать файл из этих студентов.
11. По исходному файлу определить студентов на n курсе, имеющих средний бал успеваемости выше среднего бала по его курсу. Сформировать файл из этих студентов.
12. По исходному файлу определить студентов на n курсе, имеющих средний бал успеваемости выше среднего бала по его курсу. Сформировать файл из этих студентов.

6.4. ИСПОЛЬЗОВАНИЕ СТРУКТУР С БИТОВЫМИ ПОЛЯМИ

Определить объединение с битовыми полями для представления заданной информации. Описать функцию вывода полей на экран. Сгенерировать случайный массив таких структур. Отсортировать элементы массива стандартной функцией `std::sort`. Найти повторяющиеся элементы и вывести их на экран.

1. Объединяющий тип `float`, битовые поля: знак, порядок, мантисса.
2. Объединяющий тип `double`, битовые поля: знак, порядок, мантисса.
3. Объединяющий тип `unsigned short` для хранения положения стрелок часов, битовые поля: час (0..11), минута (0..59), секунда (0..59).
4. Объединяющий тип `unsigned short`, битовые поля: день(1..31), месяц(1..12), год (0..99).
5. Объединяющий тип `unsigned int`, битовые поля: 4 разряда IPv4 адреса (0..255).
6. Объединяющий тип `unsigned int` для хранения номеров автомобилей, битовые поля: 4 десятичных разряда (0..9999), две заглавных латинских буквы, цифра кода области (1..7).
7. Объединяющий тип `unsigned int` для хранения места в поезде, битовые поля: номер поезда (число от 1 до 1000 и символ-буква), номер вагона (1..32), номер места (1..54).
8. Объединяющий тип `unsigned int` для хранения места в самолете, битовые поля: направление полета (две заглавных латинских буквы), номер рейса (1..1000), ряд места (число 1..200), номер места (латинская буква А..Н).
9. Объединяющий тип `unsigned char` для хранения номера аудитории в учебном корпусе, битовые поля: этаж (1..8), номер аудитории (1..32).
10. Объединяющий тип `unsigned short`, битовые поля: номер школы (1..277), класс (число 1..11 и русская буква А..З).
11. Объединяющий тип `unsigned int` для хранения номера мобильного телефона, битовые поля: код оператора (двузначное число), номер телефона (7 разрядов).
12. Объединяющий тип `unsigned short` для хранения информации о студенте факультета, битовые поля: курс (1..4), номер группы (1..10), номер в списке студентов группы (1..32).

7. ДИНАМИЧЕСКИЕ СТРУКТУРЫ ДАННЫХ

7.1. ДИНАМИЧЕСКОЕ ВЫДЕЛЕНИЕ ПАМЯТИ ДЛЯ ОДНОМЕРНЫХ МАССИВОВ

Выполнить задание, используя динамическое выделение памяти для одномерного массива. Дано (в текстовом файле) $n > 10^9$ натуральных чисел (каждое число $\leq 10^4$). Ограничение на объем используемой памяти 1 Мб. Получить массив за один просмотр элементов файла:

1. в котором все числа исходного файла расположены по возрастанию значений (без повторений);
2. в котором все числа исходного файла расположены по убыванию значений (без повторений);
3. образованный из натуральных чисел $\leq 10^4$, не встречающихся в исходном файле, по возрастанию;
4. образованный из натуральных чисел $\leq 10^4$, не встречающихся в исходном файле, по убыванию;
5. образованный из исходного исключением чисел, которые встречаются в исходном файле ровно один раз, по возрастанию значений (без повторений);
6. образованный из чисел, встречающихся в исходном файле более одного раза, по возрастанию (без повторений);
7. образованный из чисел, встречающихся в исходном файле более одного раза, по убыванию без повторений;
8. образованный из чисел, встречающихся в исходном файле ровно два раза, по возрастанию (без повторений);
9. образованный из чисел, встречающихся в исходном файле ровно два раза, по убыванию (без повторений);
10. образованный из исходного исключением повторных вхождений одного и того же числа по возрастанию значений;
11. в котором все числа исходного файла расположены по возрастанию (без повторений);
12. в котором все числа исходного файла расположены по убыванию (без повторений).

7.2. ДИНАМИЧЕСКОЕ ВЫДЕЛЕНИЕ ПАМЯТИ ДЛЯ МАТРИЦ

Выполнить задание, используя динамическое выделение памяти для двумерного массива, двумя способами:

- *описывая двумерный массив как одномерный, с расчетом смещения элемента массива по линейной формуле;*
- *описывая двумерный массив как указатель на массив указателей.*

1. Дана вещественная квадратная матрица A порядка n , векторы x, y с n элементами. Составить функцию, которая позволяет получить вектор $c = A \cdot (x + y)$.
2. Даны вещественные квадратные матрицы A, B, C порядка n . Составить функцию для получения матрицы $D = A \cdot (B + C)$
3. Дана вещественная квадратная матрица A порядка n , вектор b с n элементами. Составить функцию, которая позволяет получить вектор $c = A^2 \cdot b$.
4. Составить функцию, которая позволяет по заданной квадратной матрице размером $n \times n$ построить вектор длиной $2n - 1$, элементы которого – максимумы элементов главной диагонали и диагоналей, параллельных ей.
5. Составить функцию, которая позволяет по заданной квадратной матрице размером $n \times n$ построить вектор длиной $2n - 1$, элементы которого – минимумы элементов побочной диагонали и диагоналей, параллельных ей.
6. Составить функцию, которая позволяет возвести вещественную матрицу произвольного размера в степень n , где n – натуральное заданное число.
7. Даны вещественные квадратные матрицы A, B порядка n . Составить функцию для получения матрицы $C = A \cdot B + B \cdot A$.
8. Дана вещественная квадратная матрица A порядка n . Составить функцию для получения матрицы $B = E + A + A^2 + \dots + A^m$, где m – заданное натуральное число.
9. Дана вещественная квадратная матрица A порядка n . Составить функцию для получения матрицы $B = A + A^2 + A^4 + A^8$.
10. Следом матрицы называется сумма элементов, расположенных на главной диагонали. Даны вещественная квадратная матрица A порядка m , натуральное число n . Составить функцию для получения суммы следов матриц A, A^2, \dots, A^n .
11. Составить функцию, которая позволяет возводить в квадрат комплексную квадратную матрицу произвольного размера.
12. Составить функцию, которая позволяет возводить в восьмую степень комплексную квадратную матрицу произвольного размера.

7.3. ЛИНЕЙНЫЙ СПИСОК СТРУКТУР

Создать связанную структуру данных – список. Разработать функции для выполнения базовых операций над линейными связанными структурами, содержащими записи со сведениями о студентах: ФИО, дата рождения, курс, успеваемость. При добавлении элемента данных в список обеспечить упорядоченность по алфавиту фамилий. Оценить асимптотическую сложность алгоритма.

1. По исходному списку определить ФИО самого младшего студента на каждом курсе. Сформировать список из этих студентов, удалив их из исходного списка.

2. По исходному списку определить ФИО самого старшего студента на каждом курсе. Сформировать список из этих студентов, удалив их из исходного списка.

3. По исходному списку определить всех студентов, ФИО которых начинается на заданную букву. Сформировать список из этих студентов, удалив их из исходного списка.

4. По исходному списку определить всех студентов, возраст которых превышает заданный. Сформировать список из этих студентов, удалив их из исходного списка.

5. По исходному списку определить всех студентов n курса. Сформировать список из этих студентов, удалив их из исходного списка.

6. По исходному списку определить всех отличников. Сформировать список из этих студентов, удалив их из исходного списка.

7. По исходному списку определить всех отличников n курса. Сформировать список из этих студентов, удалив их из исходного списка.

8. По исходному списку определить всех неуспевающих студентов. Сформировать список из этих студентов, удалив их из исходного списка.

9. По исходному списку определить всех неуспевающих студентов n курса. Сформировать список из этих студентов, удалив их из исходного списка.

10. По исходному списку определить студентов, имеющих средний бал успеваемости выше общего среднего бала. Сформировать список из этих студентов, удалив их из исходного списка.

11. По исходному списку определить студентов на n курсе, имеющих средний бал успеваемости выше среднего бала по его курсу. Сформировать список из этих студентов, удалив их из исходного списка.

12. По исходному списку определить студентов на n курсе, имеющих средний бал успеваемости выше среднего бала по его курсу. Сформировать список из этих студентов, удалив их из исходного списка.

7.4. ЛИНЕЙНЫЕ СПИСКИ КАК ДИНАМИЧЕСКАЯ СТРУКТУРА ДАННЫХ

Выполнить задания, используя связанные динамические структуры данных в виде связанных компонент. Оценить асимптотическую сложность алгоритма.

1. Описать функцию, которая формирует список символов L_2 , включив в него те символы из L_1 , которые входят в него по одному разу.
2. Описать функцию, которая формирует список символов L_2 , включив в него те символы из L_1 , которые входят в него более одного раза.
3. Описать функцию, которая формирует список символов L_2 , включив в него те символы из L_1 , которые входят в него ровно два раза.
4. Описать функцию, которая формирует список символов-букв L_2 , включив в него те символы-буквы, которые не входят в список L_1 .
5. Описать функцию, которая формирует список символов L , включив в него по одному разу элементы, которые входят хотя бы в один из списков L_1 и L_2 .
6. Описать функцию, которая формирует список символов L , включив в него по одному разу элементы, которые входят одновременно в оба списка L_1 и L_2 .
7. Описать функцию, которая формирует список символов L , включив в него по одному разу элементы, которые входят в один из списков L_1 и L_2 , но в то же время не входят в другой из них.
8. Описать функцию, которая формирует список символов L , включив в него по одному разу элементы, которые входят в список L_1 и не входят в список L_2 .
9. Описать функцию, которая из списка символов L удаляет все вхождения списка L_1 (если такой есть).
10. Описать функцию, которая в списке символов L заменяет первое вхождение списка L_1 (если такой есть) на список L_2 .
11. Описать функцию, которая в списке символов L заменяет все вхождения списка L_1 (если такой есть) на список L_2 .
12. Описать функцию, которая в списке символов L заменяет последнее вхождение списка L_1 (если такой есть) на список L_2 .

7.5. ЛИНЕЙНЫЕ СПИСКИ КАК МАССИВЫ ДАННЫХ

Выполнить задания 7.4, используя связанные динамические структуры данных в виде массивов. Оценить асимптотическую сложность алгоритма.

7.6. ДВУСВЯЗНЫЕ СПИСКИ КАК ДИНАМИЧЕСКИЕ СТРУКТУРЫ ДАННЫХ

Выполнить задания, используя двусвязные динамические структуры данных в виде связанных компонент. Оценить асимптотическую сложность алгоритма.

1. Даны натуральное число n , действительные числа x_1, x_2, \dots, x_n . Разработать программу вычисления значения выражения следующего вида:
$$x_1 \cdot x_n + x_2 \cdot x_{n-1} + \dots + x_n x_1.$$
2. Даны натуральное число n , действительные числа x_1, x_2, \dots, x_n . Разработать программу вычисления значения выражения следующего вида:
$$(x_1 + x_n) \cdot (x_2 + x_{n-1}) \cdot \dots \cdot (x_n + x_1).$$
3. Даны натуральное число n , действительные числа x_1, x_2, \dots, x_n . Разработать программу вычисления значения выражения следующего вида:
$$(x_1 + x_2 + nx_n) + (x_2 + x_3 + (n-1)x_{n-1}) + \dots + (x_{n-1} + x_n + 2x_2).$$
4. Даны натуральное число n , действительные числа x_1, x_2, \dots, x_n . Разработать программу вычисления значения выражения следующего вида:
$$(x_1 + x_2 + 2x_n) + (x_2 + x_3 + 2x_{n-1}) + \dots + (x_{n-1} + x_n + 2x_2).$$
5. Даны натуральное число n , действительные числа a_1, a_2, \dots, a_{2n} . Получить
$$(a_1 - a_{2n}) \cdot (a_3 - a_{2n-2}) \cdot (a_5 - a_{2n-4}) \cdot \dots \cdot (a_{2n-1} - a_2).$$
6. Даны натуральное число n , действительные числа a_1, a_2, \dots, a_{2n} . Получить
$$(a_n - a_{n+1}) \cdot (a_{n-1} - a_{n+2}) \cdot (a_{n-2} - a_{n+3}) \cdot \dots \cdot (a_1 - a_{2n}).$$
7. Даны натуральное число n , действительные числа a_1, a_2, \dots, a_{2n} . Получить
$$a_1 a_{2n} + a_2 a_{2n-1} + \dots + a_n a_{n+1}.$$
8. Даны натуральное число n , действительные числа a_1, a_2, \dots, a_{2n} . Получить
$$a_n a_{n+1} + a_{n-1} a_{n+2} + \dots + a_1 a_{2n}.$$
9. Даны натуральное число n , действительные числа a_1, a_2, \dots, a_{2n} . Получить
$$\min(a_1 + a_{2n}, a_2 + a_{2n-1}, \dots, a_n + a_{n+1}).$$
10. Даны натуральное число n , действительные числа a_1, a_2, \dots, a_{2n} . Получить
$$\max(a_1 + a_{2n}, a_2 + a_{2n-1}, \dots, a_n + a_{n+1}).$$
11. Даны натуральное число n , действительные числа a_1, a_2, \dots, a_{2n} . Получить
$$\max(\min(a_1, a_{2n}), \min(a_2, a_{2n-1}), \dots, \min(a_n, a_{n+1})).$$
12. Даны натуральное число n , действительные числа a_1, a_2, \dots, a_{2n} . Получить
$$\min(\max(a_1, a_{2n}), \max(a_2, a_{2n-1}), \dots, \max(a_n, a_{n+1})).$$

7.7. БИНАРНОЕ ДЕРЕВО ПОИСКА

По заданной последовательности различных целых чисел построить соответствующее бинарное дерево поиска T как динамическую структуру данных. Выполнить следующие задания и вывести элементы дерева на экран. Оценить асимптотическую сложность алгоритма.

1. Определяет значение самого левого листа дерева.
2. Определить число листьев дерева.
3. Удалить вершину с минимальным значением элементов.
4. Удалить вершину с максимальным значением элементов.
5. Определить число элементов k -ого уровня.
6. Вывести на экран все листья дерева.
7. Определить номер уровня, в котором содержится максимальное количество вершин.
8. Определить максимальную глубину дерева.
9. Удалить все листья дерева.
10. Определить число ветвей от корня до вершины с заданным элементом. Вывести часть дерева от вершины до данного элемента на экран.
11. Определить число ветвей n -го уровня этого дерева.
12. Дополнить полученное дерево новыми вершинами так, чтобы каждая вершина, которая не является листом, имела ровно двух преемников, значения элементов в дополнительных вершинах задавать равными значениям их предшественников.

7.8. ГРАФЫ

Создать подходящий тип данных для представления графа. Написать и протестировать процедуры, обеспечивающие добавление вершины в граф, вывода элементов графа на экран, выполнить следующие задания. Количество вершин графа ≤ 65000 , среднее число связей на вершину 6, ограничение по памяти 64 MiB. Оценить асимптотическую сложность алгоритма.

1. Найти все вершины заданного графа, недостижимые от заданной его вершины.
2. Определить, что для любой пары вершин заданного орграфа одна из этих вершин достижима от другой.
3. Определить, является ли связным заданный граф.
4. Найти длину кратчайшего цикла в графе.
5. Для двух выделенных вершин графа найти соединяющий их простой путь.
6. Найти самый длинный простой путь в графе.
7. Найти все вершины графа, к которым существует путь заданной длины от выделенной его вершины.
8. Найти все вершины орграфа, от которых существует путь заданной длины к выделенной вершине.
9. Источником орграфа назовем вершину, от которой достижимы все другие вершины; стоком – вершину, достижимую от всех других вершин. Найти все источники и стоки данного орграфа.
10. Найти такую вершину заданного графа, которая принадлежит каждому пути между двумя выделенными (различными) вершинами и отлична от каждой из них.
11. Найти диаметр графа, т. е. максимум расстояний между всевозможными парами его вершин.
12. Найти медиану графа, т. е. такую его вершину, что сумма расстояний от нее до остальных вершин минимальна.

8. КЛАССЫ

8.1. КЛАСС «ВЕКТОР»

Разработать класс «вектор», моделирующий математическое понятие «одномерный массив» произвольного размера с возможностью изменения числа элементов.

Реализовать три вида конструктора (без аргументов, инициализации, копирования), деструктор.

Определить методы:

- *задания размерности вектора,*
- *определения количества элементов заданного вектора;*
- *заполнения элементов вектора случайными числами.*

Реализовать:

- *перегрузку операторов индексирования [], возвращающие значения L-value и R-value с контролем выхода за пределы размерности вектора (для получения элемента вектора по заданному индексу, занесения значения элемента вектора по заданному индексу),*
- *перегрузку операторов ввода/вывода значений элементов вектора,*

Разместить описание класса в заголовочном файле, а определения методов и главную функцию программы – в отдельных файлах.

Используя объекты описанного класса «Вектор», реализовать алгоритм своего варианта задания 7.1.

8.2. ИНКАПСУЛЯЦИЯ

Объявить класс для решения задачи и определить его методы.

Реализовать:

- *три вида конструктора (без аргументов, инициализации, копирования),*
- *деструктор,*
- *перезгрузить операции ввода/вывода данных,*
- *указанные в задании операции реализовать посредством перегрузки операций.*

Объявление класса разместить в заголовочном файле, определения методов – во втором файле, определение главной функций программы – в третьем файле. Продемонстрировать корректную работу всех операций и методов реализованного класса.

1. Определить класс для работы с объектом «комплексное число», которое задается вещественной и мнимой частями. Реализовать: сложение, вычитание, произведение, деление двух чисел, возведение в целочисленную степень, извлечение квадратного корня.

2. Определить класс для работы с объектом «строка символов». Реализовать: сравнение строк, удаление из строки указанного символа. Удалить из заданной строки все символы, встречающиеся во второй заданной строке.

3. Определить класс для работы с объектом «рациональная дробь» (вида m/n). Реализовать: сложение, вычитание, умножение, деление двух дробей, приведение дроби к несократимому виду, вывод дроби в виде m/n , сравнение двух дробей.

4. Определить класс для работы с объектом «полином». Ввод осуществить по возрастанию степеней членов полинома. Реализовать: сложение, вычитание, умножение, деление двух полиномов, умножение полинома на число, вычисление значения полинома в заданной точке, дифференцирование полинома.

5. Определить класс для работы с объектом «комплексное число», которое задается своим модулем и углом. Реализовать: сложение, вычитание, произведение, деление двух чисел, возведение в целочисленную степень, извлечение квадратного корня.

6. Определить класс для работы с объектом «строка символов». Реализовать: вставку в строку указанного символа в заданную позицию, удаление ведущих пробелов. Добавить в заданную строку отсутствующие символы из второй заданной строки, на позиции с такими же номерами.

7. Определить класс для работы с объектом «множество целых чисел». Реализовать: объединение двух множеств, пересечение двух множеств, разность двух множеств, добавление элемента во множество, удаление элемента из множества.

8. Определить класс для работы с объектом «строка символов». Реализовать: замену подстроки на подстроку, конкатенацию строк. Заменить в заданной строке заданную подстроку на другую заданную подстроку (столько раз, сколько она встречается в тексте).

9. Определить класс для работы с объектом «множество действительных чисел». Реализовать: объединение двух множеств, пересечение двух множеств, разность двух множеств, добавление элемента во множество, удаление элемента из множества.

10. Определить класс для работы с объектом «время», которое задается в формате час.минута.секунда. Реализовать (учесть переход через 24 ч.): добавления ко времени заданного количества секунд, вычитания из времени заданного количества секунд, сложения двух моментов времени, вычитание из одного момента времени другого, подсчет числа секунд между двумя моментами времени, лежащими в пределах одних суток.

11. Определить класс для работы с объектом «строка символов». Реализовать: удаление подстроки из строки, проверку, является ли строка палиндромом. Удалить из заданной строки заданную подстроку (столько раз, сколько она встречается в тексте).

12. Определить класс для работы с объектом «дата», которое задается в формате число.месяц.год. Реализовать (учесть високосные года): сложение даты с заданным количеством дней, вычитания из даты заданного количества дней, вычисление числа дней, прошедших между двумя датами.

8.3. КЛАСС «МНОГОРАЗРЯДНОЕ ЧИСЛО»

Объявить класс «многоразрядное целое число». Количество разрядов числа хранить в поле класса, значения разрядов числа хранить в динамической памяти, основание системы счисления задавать константой.

Реализовать:

- *три вида конструктора (без аргументов, инициализации, копирования),*
- *деструктор,*
- *перегрузить операции ввода/вывода данных.*
- *перегрузить оператор индексирования [] для доступа к разрядам числа,*
- *операции в задании реализовать посредством перегрузки.*

Объявление класса разместить в заголовочном файле, определения методов – во втором файле, определение главной функций программы – в третьем файле.

При переполнении старшего разряда лишние биты теряются.

Примечание.

Оператор умножения многоразрядных чисел удобно реализовать через более простые перегруженные операторы: умножение многоразрядного числа на одноразрядное число, сдвиг многоразрядного числа на целое число разрядов, сложение многоразрядных чисел.

Оператор деления многоразрядных чисел удобно реализовать через более простые перегруженные операторы: сдвиг многоразрядного числа на целое число разрядов, сравнение и вычитание многоразрядных чисел.

Выполнить задание, используя объекты этого класса.

1. Объявить класс «32-разрядное целое положительное десятичное число». Реализовать операции: сложения, умножения этих чисел. Вычислить $(1787109376_{10})^3$.
2. Объявить класс «64-разрядное целое двоичное число со знаком» (использовать дополнительный двоичный код). Реализовать операции: сложения, умножения, операцию возведения в степень. Вычислить $3^{12} + (-3)^{11} + 3^{10} + (-3)^9$.
3. Объявить класс «256-разрядное целое положительное двоичное число». Реализовать операции: вычитания, сдвигов на n разрядов, целочисленного деления, получения остатка от деления. Создать метод для перевода «256-разрядного целого двоичного числа» в десятичное число (представленное в виде строки). Перевести в десятичную систему счисления число 2^{200} .
4. Объявить класс «64-разрядное целое положительное двоичное число». Реализовать операции: сложения, умножения, операцию возведения в степень. Вычислить $3^{12} + 3^{11} + 3^{10} + 3^9$.
5. Объявить класс «64-разрядное целое положительное десятичное число». Реализовать операции: сложения, умножения этих чисел. Вычислить $(8212890625_{10})^5$.
6. Объявить класс «256-разрядное целое положительное двоичное число». Реализовать операции: вычитания, сдвигов на n разрядов, целочисленного деления, получения остатка от деления. Найти НОД чисел $2^{249} - 2^6 + 7$ и $2^{108} - 1$.
7. Объявить класс « n -разрядное целое положительное число в 10^9 -ричной с/с» (каждый разряд числа представляет девять десятичных разрядов). Реализовать операции: сложения, умножения. Вычислить $100_{10}!$.
8. Объявить класс « n -разрядное целое положительное число в 2^{32} -ричной с/с» (каждый разряд числа представляет тридцать два двоичных разряда(int)). Реализовать операции: сложения, умножения. Вычислить $(5_{10})^{100}$.
9. Объявить класс « n -разрядное целое положительное число в 100-ричной с/с» (каждый разряд числа представляет два десятичных разряда). Реализовать операции: сложения, умножения. Вычислить $100_{10}!$.
10. Объявить класс « n -разрядное целое положительное число в 10-ричной с/с». Реализовать операции: сложения, умножения. Вычислить $100_{10}!$.
11. Объявить класс « n -разрядное целое положительное число в 16-ричной с/с». Реализовать операции: сложения, умножения. Вычислить $(5_{10})^{100}$.
12. Объявить класс « n -разрядное целое положительное число в 256-ричной с/с» (каждый разряд числа представляет два шестнадцатеричных разряда). Реализовать операции: сложения, умножения. Вычислить $(5_{10})^{100}$.

8.4. КЛАСС «ЛИНЕЙНЫЙ СПИСОК»

Объявить шаблон класса «Линейный список». Определить структуру класса и его методы. В классе должен быть конструктор по умолчанию, конструктор с формальными параметрами, конструктор копирования и деструктор.

Определить методы:

- *инициализация первого элемента структуры;*
- *добавления в структуру нового элемента;*
- *удаление элемента;*
- *поиск элемента по заданному ключу поиска (значению информационного поля);*
- *вывод элементов списка на экран;*
- *удаления списка.*

Используя объекты класса «Линейный список», реализовать алгоритмы своих вариантов заданий 7.4.

8.5. КЛАСС «ДВУСВЯЗНЫЙ СПИСОК»

Объявить шаблон класса «Двусвязный линейный список». Определить структуру класса и его методы. В классе должен быть конструктор по умолчанию, конструктор с формальными параметрами, конструктор копирования и деструктор.

Определить методы:

- *инициализация первого элемента структуры;*
- *запись в структуру нового элемента;*
- *удаление элемента;*
- *поиск элемента по заданному ключу поиска (значению информационного поля);*
- *вывод элементов списка на экран;*
- *удаления всех элементов списка.*

Используя объекты класса «Двусвязный линейный список», реализовать алгоритм своего варианта задания 7.6.

8.6. КЛАСС «БИНАРНОЕ ДЕРЕВО»

Объявить шаблон класса «Бинарное дерево». Определить структуру класса и его методы. В классе должен быть конструктор по умолчанию, конструктор с формальными параметрами, конструктор копирования и деструктор.

Определить методы:

- *инициализация первого элемента структуры;*
- *запись в структуру нового элемента;*
- *удаление элемента (листа);*
- *поиск элемента по заданному ключу поиска (значению информационного поля);*
- *вывод элементов дерева на экран;*
- *удаления всех элементов дерева.*

Используя объекты класса «Бинарное дерево», реализовать алгоритм своего варианта задания 7.7.

8.7. КЛАСС «ГРАФ»

Объявить шаблон класса «Граф». Определить структуру класса и его методы. В классе должен быть конструктор по умолчанию, конструктор с формальными параметрами, конструктор копирования и деструктор.

Определить методы:

- *инициализация первого элемента структуры;*
- *запись в структуру нового элемента;*
- *удаление элемента;*
- *вывод элементов графа на экран;*
- *удаления всех элементов графа.*

.

Используя объекты класса «Граф», реализовать алгоритм своего варианта задания 7.8.

8.8. КЛАСС «МАТРИЦА»

Объявить класс «Матрица» и определить его методы.

Обеспечить представление матрицы произвольного размера с возможностью изменения числа строк и столбцов. Каждая строка матрицы – объект класса «вектор» из задания 8.7.

Обеспечить в конструкторе автоматическое генерирование имени матрицы («матрица 1», «матрица 2» и т.д.), используя для этого статическое поле – счетчик объектов класса.

Реализовать в классе метод вывода матрицы на экран с одновременным выводом ее имени. Алгоритм обработки, реализующий условие задачи, определить как функцию-член класса или как дружественную функцию класса.

*Перегрузить оператор **индексирования** [] для обращения к элементам матрицы с контролем выхода за пределы размерностей.*

*В случае выхода за пределы массива генерировать исключения с помощью ключевого слова **throw**.*

Класс может реализовывать следующие операции над матрицами (в зависимости от условия задания):

- сложение, вычитание, умножение, деление на число
- сложение, вычитание, умножение на другую матрицу
- комбинированные операции присваивания ($+=$, $-=$, $*=$, $/=$);
- операции сравнения на равенство/неравенство;
- операции вычисления транспонированной матрицы;
- операцию возведения в степень;
- методы, реализующие проверку типа матрицы (квадратная, диагональная, нулевая, единичная, симметричная, верхняя треугольная, нижняя треугольная);
- операции ввода/вывода матрицы в стандартные потоки.

Обеспечить работу с безопасным массивом, т.е. контролировать выход индекса элемента массива за допустимый описанием объекта диапазон.

Используя объекты описанного класса «Матрица», реализовать алгоритм своего варианта задания 7.2.

9. БИБЛИОТЕКА ШАБЛОНОВ STL

9.1. ВЕКТОРЫ

1. Реализовать задание 7.1, используя объявление и методы соответствующего параметризованного класса `vector` из стандартной библиотеки шаблонов STL.
2. Реализовать задание 5.3, используя объявление и методы соответствующего параметризованного класса `vector` из стандартной библиотеки шаблонов STL.
3. Реализовать задание 7.2, используя объявление и методы соответствующего параметризованного класса `vector` из стандартной библиотеки шаблонов STL.

9.2. СПИСКИ

Реализовать решение задачи с использованием параметризованных классов `vector` и `list` (указать *лучший* и объяснить почему).

1. Реализовать варианты заданий 7.3, используя объявление и методы соответствующего параметризованного класса из стандартной библиотеки шаблонов STL.
2. Реализовать вариант заданий 7.4, используя объявление и методы соответствующего параметризованного класса из стандартной библиотеки шаблонов STL.
3. Реализовать вариант заданий 7.6, используя объявление и методы соответствующего параметризованного класса из стандартной библиотеки шаблонов STL.

9.3. ВЫЧИСЛЕНИЕ ВЫРАЖЕНИЙ С ИСПОЛЬЗОВАНИЕМ ОБРАТНОЙ ПОЛЬСКОЙ ЗАПИСИ

Выполнить задания с использованием подходящих связанных динамических структур данных. Оценить асимптотическую сложность алгоритма.

1. Написать программу, которая по заданному алгебраическому выражению (операнды – целые числа; операции: $() - + * / \%$) вычисляет результат.
2. Написать программу, которая по заданному алгебраическому выражению (операнды – вещественные числа; операции: $() - + * /$) вычисляет результат.
3. Написать программу, которая по заданному логическому выражению (операнды – целые числа и булевские константы; операции отношения: $> < == != >= <=$, логические операции: $\&\& || !$ и скобки $()$) вычисляет результат.
4. Написать программу, которая по заданному выражению вычисляет результат (операнды – целые числа; битовые операции: $\& | ^ \sim >> <<$ и скобки $()$).

9.4. АССОЦИАТИВНЫЕ СПИСКИ

Дана программа на C++. Проанализировать программу за один просмотр и записать в выходной файл построчно:

1. все ключевые слова, встречающиеся в этой программе, указав количество появлений для каждого ключевого слова и номера строк программы, в которых они встречаются.
2. все идентификаторы, встречающиеся в этой программе, указав количество появлений для каждого идентификатора и номера строк программы, в которых они встречаются.
3. все неименованные константы, встречающиеся в этой программе, указав количество появлений для каждой константы и номера строк программы, в которых они встречаются.
4. имена всех реализованных в этой программе функций, указав для каждой из них, из каких функций она вызывается.

10. НАСЛЕДОВАНИЕ. ПОЛИМОРФИЗМ

10.1. НАСЛЕДОВАНИЕ

В предложенных ниже вариантах описывается конкретная задача. Для неё:

1. Определить и описать иерархию классов: набор схожих по какому-либо критерию объектов, объединённых в подгруппы (родительские объекты).

Для каждого класса описать поля, характеризующие объект, и методы, позволяющие заполнять и получать значения каждого из полей, выводить значения полей на экран, а также необходимые конструкторы. Общие свойства хранить в родительских классах.

- Конструкторы и методы должны проверять параметры на допустимость.

- Вывод свойств родительских классов осуществить с использованием наследования и полиморфизма (виртуальный метод `Print()`, который вызывается из `operator<<` базового класса). Каждый класс должен уметь выводить в виде текста свой тип и набор своих свойств.

2. Обсудить схему с преподавателем.

3. Реализовать иерархию в коде.

4. Создать программу для тестирования описанных объектов, создающую и заполняющую свойства нескольких отдельных объектов, а затем выводящую свойства всех созданных объектов на экран. Первичную инициализацию свойств объектов осуществите с помощью конструкторов.

Примечание 1: чтобы позднее такую иерархию классов было удобнее расширять, начинайте её с абстрактного базового класса.

Примечание 2: во всех вариантах предполагается не менее 3 уровней в иерархии наследования. Где удобно - добавляйте в иерархию не указанные в задании абстрактные или обычные классы.

Вариант 1

Вам заказали спроектировать программу для учёта банковских вкладов.

Разработайте иерархию наследования классов, позволяющую хранить информацию о всех депозитных счетах и их свойствах в программе:

1. Вклад до востребования (бессрочный вклад)

- номер счёта
- дата создания вклада
- годовая процентная ставка
- текущая сумма
- дата последнего начисления процентов

2. Условный вклад (отличается от вклада «до востребования» наличием определённого условия для его закрытия)

- номер счёта
- дата создания вклада
- годовая процентная ставка
- текущая сумма
- дата последнего начисления процентов
- условие закрытия вклада (срока, например, "до вступления в брак", "до достижения совершеннолетия", или другие условия из завещаний)

3. Срочный безотзывной вклад

- номер счёта
- дата создания вклада
- годовая процентная ставка
- текущая сумма
- дата последнего начисления процентов
- срок вклада в месяцах
- период в течении которого допускаются дополнительные взносы во вклад (количество месяцев)

4. Срочный отзывной вклад

- номер счёта
- дата создания вклада
- годовая процентная ставка
- текущая сумма
- дата последнего начисления процентов
- срок вклада в месяцах
- годовая процентная ставка(пониженная) в случае досрочного расторжения договора
- - начальная сумма (для пересчёта процентов по пониженной ставке)

Вариант 2

Вы проектируете многопользовательскую 2D RPG игру с видом от третьего лица. В одной локации может находиться множество неподвижных предметов и игроков. Для реализации прототипа игры вам надо разработать иерархию наследования классов, позволяющую хранить все эти сущности и их свойства в программе. А именно:

1. Камень

- координаты X и Y (целые)
- размеры - ширина и высота (целые беззнаковые)
- идентификатор картинки, отображаемой на месте предмета (число от 1 до 10000)

2. Сундук

- координаты X и Y (целые)
- размеры - ширина и высота (целые беззнаковые)
- количество золотых монет в сундуке

3. Торговец

- текущие координаты X и Y (целые)
- имя
- код типа продаваемых товаров (число от 1 до 100)

4. Мирный житель

- текущие координаты X и Y (целые)
- имя

5. Игроки

- текущие координаты X и Y (целые)
- имя
- цвет одежды

Вариант 3

Вы проектируете 3D RPG игру на космическую тему.

В на игровой карте могут находиться одновременно разные сущности. Для реализации первой версии игры вам необходимо разработать иерархию наследования классов для представления в программе следующего минимального набора сущностей и их свойств:

1. Неподвижные (система координат в каждый момент времени отсчитывается относительно звезды или ближайшей планеты): звезда, планета
 - название
 - температура
 - наличие твёрдой поверхности, на которую можно высадиться
2. Астероид - подвижный мелкий объект.
 - координаты (X, Y, Z)
 - скорость (V_x , V_y , V_z)
 - максимальный размер в поперечнике
3. Космический корабль (под управлением игрока, или враждебный)
 - имя
 - координаты (X, Y, Z)
 - скорость (V_x , V_y , V_z)
 - максимальная скорость V_{\max}
 - боезапас (сколько осталось выстрелов)
4. Ракета выпущенная одним из кораблей
 - координаты (X, Y, Z)
 - скорость (V_x , V_y , V_z)
 - мощность взрывчатки
 - время, на которое остался запас топлива (в секундах)

Вариант 4

Вы проектируете программу для оформления бухгалтерской отчётности факультета РФиКТ. Разработайте иерархию наследования классов, объединяющую всех людей на факультете, и позволяющую хранить всю необходимую информацию о них.

Категории сущностей и свойства, которые должны присутствовать в программе:

1. Студент

- ФИО
- год обучения
- номер студенческого билета (число до 9 цифр)
- средний балл

2. Магистрант

- ФИО
- год обучения
- номер студенческого билета (число до 9 цифр)
- средний балл

3. Аспирант

- ФИО
- год обучения
- номер студенческого билета (число до 9 цифр)
- средний балл

4. Преподаватель

- ФИО
- должность (строка)
- учёная степень (строка)

5. Учебно-вспомогательный персонал

- ФИО
- должность (строка)
- номер лаборатории (число от 1 до 100)

Вариант 5

Вы проектируете графический пользовательский интерфейс для новой операционной системы. Разработайте иерархию наследования классов для представления элементов управления и их свойств. Минимальный набор элементов, необходимый для реализации прототипа системы:

1. **Ellipse** – рисует вписанный в рамки элемента управления эллипс
 - **Visible** - отображается элемент управления или скрыт (bool)
 - **FillColor** – цвет заливки
 - **BorderColor** – цвет контура
 - Координаты X и Y
 - Размеры высота и ширина элемента управления
2. **HyperlinkLabel** – при нажатии мышкой обеспечивает переход по гиперссылке
 - **Text** - текст в текстовом поле
 - **Color** - цвет текста элемента управления
 - **Visible** - отображается элемент управления или скрыт (bool)
 - Координаты X и Y
 - Размеры высота и ширина элемента управления
 - **URL** – адрес ссылки для перехода (строка)
3. **TextBox** - текстовое поле, позволяет пользователю вводить текст
 - **Text** - текущий текст в текстовом поле
 - **Color** - цвет текста элемента управления
 - **Readonly** – запрещает редактирование текста пользователем (bool)
 - **Visible** - отображается элемент управления или скрыт (bool)
 - Координаты X и Y
 - Размеры высота и ширина элемента управления
4. **Button** – кнопка
 - **Text** - текущий текст на кнопке
 - **ToolTipText** - текст, появляющийся в подсказке при наведении мышки на кнопку
 - **Visible** - отображается элемент управления или скрыт (bool)
 - Координаты X и Y
 - Размеры высота и ширина элемента управления

Вариант 6

Вас пригласили поучаствовать в разработке системы видеонаблюдения для национального аэропорта «Минск». Система будет обладать элементами искусственного интеллекта, а именно: она будет выделять для каждого кадра в видеопотоке отдельные сущности и классифицировать их. В данный момент вам необходимо разработать иерархию наследования классов для представления результатов работы модуля распознавания – набора обнаруженных в кадре сущностей. Далее приведены сущности (и их свойства), которые выдаёт первая версия программы:

1. Человек

- координаты области в кадре (X_{min} , Y_{min} , X_{max} , Y_{max})
- пол (М, Ж, не удалось определить)
- ребёнок (да, нет, не удалось определить)
- наличие очков (есть, нет, ракурс не позволяет определить)
- наличие бороды (есть, нет, ракурс не позволяет определить)

2. Велосипедист

- координаты области в кадре (X_{min} , Y_{min} , X_{max} , Y_{max})
- пол (М, Ж, не удалось определить)
- ребёнок (да, нет, не удалось определить)
- наличие очков (есть, нет, ракурс не позволяет определить)
- наличие бороды (есть, нет, ракурс не позволяет определить)
- цвет велосипеда (строка)

3. Автомобиль

- координаты области в кадре (X_{min} , Y_{min} , X_{max} , Y_{max})
- цвет
- номер (строка)
- такси (да, нет)

4. Бесхозная сумка

- координаты области в кадре (X_{min} , Y_{min} , X_{max} , Y_{max})
- время суток, когда была первый раз замечена

Вариант 7

Краеведческий музей в деревне Простоквашино нуждается в современной системе учёта экспонатов. Разработайте иерархию наследования классов, которая будет использована в такой системе.

Следующие виды экспонатов хранятся в настоящий момент в музее (также приведены важные для учёта свойства):

1. Монеты

- наименование (строка)
- номинал (может быть как целым, так и долями от целого)
- годы использования (диапазон лет – два целочисленных года)

2. Холодное оружие

- наименование (строка)
- приблизительная датировка (диапазон лет – два целочисленных года)

3. Картины

- что изображено (строка)
- автор (строка)
- время создания (год)

4. Статуэтки

- что изображено (строка)
- автор (строка)
- время создания (год)

5. Фотографии

- что изображено (строка)
- время создания (год)

Вариант 8

Компания, имеющая большой товарный оборот на собственных складских площадях, и, ведущая вручную весь учёт, озабочена автоматизацией процесса движения товара на складах. Вам необходимо разработать иерархию наследования классов, объединяющую все товары, хранимые на складе. Эти классы будут использованы для реализации функции складского учёта (постановка на учёт прибывающих товаров и снятие товаров с учёта при отправке заказов)

Виды товара, хранящиеся на складе в настоящий момент, и свойства, которые необходимо хранить в системе учёта:

1. Бытовая техника

- Наименование (строка)
- производитель (строка)
- цена
- остаток на складе (штук)

2. Верхняя одежда

- наименование (строка)
- размер
- материал (строка)
- сезон (строка)
- цена
- остаток на складе (штук)

3. Спортивная одежда

- наименование (строка)
- размер
- материал (строка)
- цена
- остаток на складе (штук)

4. Нижнее бельё

- наименование (строка)
- размер
- материал (строка)
- цена
- остаток на складе (штук)

Вариант 9

Одна известная транспортная компания планирует внедрить у себя на сайте функционал по построению маршрута из любой точки Минска в любую. Вам поручено разработать иерархию наследования классов для представления в программе сегментов такого маршрута. Каждый сегмент представляет часть маршрута, преодолеваемую на одном транспорте без пересадок. Виды транспорта и дополнительная информация, которую необходимо хранить в первой версии программы:

1. Автомобиль

- координаты GPS точки начала сегмента маршрута (float долгота и широта)
- координаты GPS точки конца сегмента маршрута (float долгота и широта)
- приблизительное время поездки в минутах
- стоимость аренды в рублях
- ссылка на страницу в интернет с арендой авто (строка)

2. Автобус

- координаты GPS точки начала сегмента маршрута (float долгота и широта)
- координаты GPS точки конца сегмента маршрута (float долгота и широта)
- приблизительное время поездки в минутах
- стоимость проезда

3. Троллейбус

- координаты GPS точки начала сегмента маршрута (float долгота и широта)
- координаты GPS точки конца сегмента маршрута (float долгота и широта)
- приблизительное время поездки в минутах
- стоимость проезда

5. Метро

- координаты GPS точки начала сегмента маршрута (float долгота и широта)
- координаты GPS точки конца сегмента маршрута (float долгота и широта)
- приблизительное время поездки в минутах
- стоимость проезда

6. Маршрутное такси

- координаты GPS точки начала сегмента маршрута (float долгота и широта)
- координаты GPS точки конца сегмента маршрута (float долгота и широта)
- приблизительное время поездки в минутах
- стоимость проезда

7. Пешком

- координаты GPS точки начала сегмента маршрута (float долгота и широта)
- координаты GPS точки конца сегмента маршрута (float долгота и широта)
- приблизительное время перемещения в минутах.

Вариант 10

Вас пригласили поучаствовать в создании программы 2D редактора векторной графики. Вам необходимо разработать иерархию наследования классов для представления в программе набора графических примитивов. Для первой версии продукта был отобран следующий минимальный набор фигур и их свойств:

1. Прямоугольник

- координаты X, Y левого верхнего угла
- размеры ширина и высота
- цвет (строка)

2. Эллипс

- координаты X, Y левого верхнего угла описывающего прямоугольника
- размеры ширина и высота
- цвет (строка)

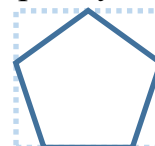
3. Звезда

- координаты X, Y левого верхнего угла описывающего прямоугольника
- размеры ширина и высота
- цвет (строка)
- количество вершин



4. Многоугольник

- координаты X, Y левого верхнего угла описывающего прямоугольника
- размеры ширина и высота
- цвет (строка)
- количество вершин



5. Текст

- координаты позиции с которой выводится текст X, Y
- размер шрифта в пт (float)
- текст (строка)
- цвет (строка)

Вариант 11

Вас пригласили поучаствовать в разработке многопользовательской онлайн-ролевой игры (MMORPG). Перед вами стоит задача разработать иерархию наследования классов для хранения в игре информации о наборе заданий на доске объявлений. Для минимальной рабочей версии игры вам предстоит обеспечить представление в игре следующих категорий заданий и их свойств:

1. Задание на отстрел монстров определённого вида в заданной локации (например, «крысы расплодились в подвале»)
 - художественно оформленный текст задания (строка)
 - количество золотых монет, получаемых за выполнение задания
 - идентификатор локации, где обитают монстры
 - идентификатор типа монстров
 - количество монстров указанного типа, которое необходимо уничтожить
2. Задание на отстрел конкретного монстра
 - художественно оформленный текст задания (строка)
 - количество золотых монет, получаемых за выполнение задания
 - идентификатор локации, где обитает монстр
 - идентификатор типа монстра
 - титул, присваиваемый победителю этого монстра (строка)
3. Заказ на доставку («принеси мне то, не знаю что...»)
 - количество золотых монет, получаемых за выполнение задания
 - идентификатор локации расположения предмета
 - идентификатор локации расположения заказчика
 - идентификатор заказываемого предмета
 - отметка времени, до которой должна быть завершена доставка
4. Задание на охрану каравана
 - количество золотых монет, получаемых за выполнение задания
 - идентификатор локаций – начала маршрута
 - идентификатор локаций – конец маршрута
 - отметка времени, до которой караван должен прибыть в место назначения
 - расчётное время движения каравана по маршруту в секундах

Примечания:

Идентификатор локации - целое число от 1 до 10^6 .

Идентификатор типа монстра - целое число от 1 до 200.

Идентификатор предмета - целое число от 1 до 10^4 .

Время измеряется в реальных секундах.

Моменты времени хранятся в виде количества секунд с начала Unix эпохи (полночь первого января 1970 года).

Вариант 12

Для учёта имеющейся компьютерной техники на факультете РФиКТ необходимо разработать программу инвентаризации. В рамках этого проекта вам необходимо разработать иерархию наследования классов для представления параметров каждого компьютера. Классы разрабатываемой иерархии должны представлять только существенные для учёта параметры компьютера вместе с их свойствами:

1. HDD

- Ёмкость
- Год выпуска
- Производитель (строка)

2. SSD

- Ёмкость
- TBW, байт
- Фактическое количество записанных на накопитель байт по показаниям S.M.A.R.T. во время последней инвентаризации
- Год выпуска
- Производитель (строка)

3. Монитор

- Диагональ в дюймах
- Рабочее разрешение (ширина и высота)
- Частота обновления экрана
- Год выпуска

4. Процессор

- Год выпуска
- Модель (строка)
- Количество ядер
- Базовая частота, ГГц

Примечания:

TBW (Total Byte Written) - заявленный компанией-производителем параметр – это количество байт данных, которое можно гарантированно записать на диск прежде, чем ресурс чипов памяти будет исчерпан.

10.2. ПОЛИМОРФИЗМ. ВИРТУАЛЬНЫЕ МЕТОДЫ

Реализовать класс «одномерный динамический массив из потомков-объектов, описанных в 10.1».

Реализовать следующие методы:

- *создание массива;*
- *занесение объекта в массив;*
- *удаление объекта из массива;*
- *вывод на экран элементов массива с соответствующими полями для каждого объекта;*
- *удаление массива.*

Определить: конструктор, конструктор копирования, оператор копирования, деструктор, виртуальные методы. Выполнить тестирование написанных методов.

Для сред разработки, поддерживающих стандарт C++11 и выше, определить конструктор перемещения и оператор перемещения.

10.3. «ФАБРИКА КЛАССА»

Реализовать персистентность объекта из задания 10.2, а именно, сохранение массива в файл при выходе из программы и загрузку из файла при запуске.

10.4. ПОЛИМОРФИЗМ. ВИРТУАЛЬНЫЕ МЕТОДЫ

Создать шаблон класса «односвязный список из потомков-объектов, описанных в 10.1». Воспользоваться решением задания 8.4. Для данного списка реализовать следующие методы:

- *инициализация списка;*
- *добавление объекта в список;*
- *удаление объекта из списка;*
- *просмотр элементов списка и вывод на экран элементов списка с соответствующими полями для каждого объекта;*
- *удаление списка.*

Определить: конструктор, конструктор копирования, оператор копирования, деструктор, виртуальные методы. Выполнить тестирование написанных методов.

Для сред разработки, поддерживающих стандарт C++11 и выше, определить конструктор перемещения и оператор перемещения.