

Немного полезной информации

1. Как «вычистить» из буфера ввода/вывода ENTER?

```
#include <limits>
void clearBuffer() { // Очистка буфера ввода/вывода
    cin.clear();
    cin.ignore(numeric_limits<streamsize>::max(), '\n');
}
int main(void) {
    ...
    clearBuffer();
    cout << "Для продолжения нажмите любую клавишу...";
    cin.get();
    ...
}
```

2. Остановка до нажатия любой клавиши: `system("Pause")`;

3. Русскоязычная консоль: `setlocale(LC_ALL, "Russian")`;

4. Если забыли, как работать с файлами:

```
FILE *f = fopen("text.txt", "w");
fprintf(f, "%i\n", 1);
fclose(f);
f = fopen("text.txt", "r");
fscanf(f, "%i\n", i);
fclose(f);
```

5. Если забыли, как работать с потоками ввода/вывода:

```
#include <iostream>
using namespace std;
int main() {
    char s[255];
    cout << "Введите строку: ";
    cin >> s;
    cout << s << endl;
    return 0;
}
```

Пузырьковая сортировка (сортировка обменом)

```
for(int a = 1; a < count; ++a)
    for(int b = count-1; b >= a; --b) {
        if(items[b-1] > items[b]) {
            t = items[b-1];
            items[b-1] = items[b];
            items[b] = t;
        }
    }
}
```

Сортировка выбором

```
for(a = 0; a < count-1; ++a) {
    exchange = 0;
    c = a;
    t = items[a];
    for(b = a+1; b < count; ++b) {
        if(items[b] < t) {
            c = b;
            t = items[b];
            exchange = 1;
        }
    }
    if(exchange) {
        items[c] = items[a];
        items[a] = t;
    }
}
```

```
for(a=1; a < count; ++a) {
    t = items[a];
    for(b=a-1; (b >= 0) && (t < items[b]); b--)
        items[b+1] = items[b];
    items[b+1] = t;
}
```

Задание 1.

1. На примере исходных данных объяснить различия указанных в вашем задании алгоритмов сортировки (в письменной форме).
2. Программа должна запрашивать из файла двумерный массив. Результат сортировки по желанию пользователя может сохраняться в файле. Имена файлов вводятся с клавиатуры.
3. Используемые массивы должны быть динамическими.
4. Каждый логически законченный фрагмент должен быть оформлен в виде отдельной функции (в том числе выделение и освобождение памяти).
5. Все необходимые данные для функции должны передаваться в качестве параметров (глобальные переменные запрещены).
6. Разработать модуль (отдельный сpp-файл, имена заголовочного и соответствующего сpp-файла должны различаться только расширением), включающий в себя:
 - реализацию указанных алгоритмов сортировки;
 - профайлер, позволяющий подсчитать количество «пересылок» и сравнений, произведенных во время сортировки.
7. Составить тесты (не менее 10 файлов) для отладки работы алгоритмов.
8. Составить сравнительную таблицу, позволяющую оценить эффективность работы каждого из алгоритмов на
 - упорядоченной,
 - случайной,
 - упорядоченной в обратном порядке

последовательности чисел. Исходная последовательность получается путем считывания подряд строк (столбцов) упорядоченного двумерного массива (она будет считаться случайной последовательностью; после её упорядочения одним из методов получаем упорядоченную в прямом порядке, инвертировав её – в обратном порядке).

- 1.1.** Дана матрица, элементами которой являются латинские буквы. Отсортировать в лексикографическом порядке элементы каждой строки: А) сортировкой выбором; В) сортировкой обменом.
- 1.2.** Дана матрица, элементами которой являются латинские буквы. Отсортировать в лексикографическом порядке элементы каждого столбца: А) сортировкой выбором; В) сортировкой обменом
- 1.3.** Дана матрица, элементами которой являются латинские буквы. Отсортировать в лексикографическом порядке элементы каждой строки: А) сортировкой вставками; В) сортировкой обменом
- 1.4.** Дана матрица, элементами которой являются латинские буквы. Отсортировать в лексикографическом порядке элементы каждого столбца: А) сортировкой вставками; В) сортировкой обменом
- 1.5.** Дана матрица, элементами которой являются латинские буквы. Отсортировать в обратном лексикографическом порядке элементы каждой строки: А) сортировкой выбором; В) сортировкой вставками.
- 1.6.** Дана матрица, элементами которой являются латинские буквы. Отсортировать в обратном лексикографическом порядке элементы каждого столбца: А) сортировкой выбором; В) сортировкой вставками.
- 1.7.** Дана матрица, элементами которой являются латинские буквы. Отсортировать в обратном лексикографическом порядке элементы каждой строки: А) сортировкой выбором; В) сортировкой обменом
- 1.8.** Дана матрица, элементами которой являются латинские буквы. Отсортировать в обратном лексикографическом порядке элементы каждого столбца: А) сортировкой выбором; В) сортировкой обменом.
- 1.9.** Дана матрица, элементами которой являются латинские буквы. Отсортировать в лексикографическом порядке элементы каждой строки: А) сортировкой вставками; В) сортировкой обменом.
- 1.10.** Дана матрица, элементами которой являются латинские буквы. Отсортировать в лексикографическом порядке элементы каждого столбца: А) сортировкой вставками; В) сортировкой обменом.
- 1.11.** Отсортировать элементы диагоналей вещественной матрицы, параллельных главной диагонали, по не возрастанию: А) сортировкой вставками; В) сортировкой обменом.
- 1.12.** Отсортировать элементы диагоналей вещественной матрицы, параллельных побочной диагонали, по не убыванию: А) сортировкой вставками; В) сортировкой обменом.
- 1.13.** Отсортировать элементы строк вещественной матрицы по не возрастанию: А) сортировкой вставками; В) сортировкой обменом.
- 1.14.** Отсортировать элементы столбцов вещественной матрицы по не возрастанию: А) сортировкой вставками; В) сортировкой обменом.
- 1.15.** Отсортировать элементы строк вещественной матрицы по не возрастанию: А) сортировкой выбором; В) сортировкой обменом.
- 1.16.** Отсортировать элементы столбцов вещественной матрицы по не возрастанию: А) сортировкой выбором; В) сортировкой обменом.

- 1.17. Отсортировать элементы строк вещественной матрицы по не убыванию: А) сортировкой выбором; В) сортировкой обменом.
- 1.18. Отсортировать элементы столбцов вещественной матрицы по не убыванию: А) сортировкой выбором; В) сортировкой обменом.
- 1.19. Отсортировать элементы строк вещественной матрицы по не убыванию: А) сортировкой выбором; В) сортировкой вставками.
- 1.20. Отсортировать элементы столбцов вещественной матрицы по не убыванию: А) сортировкой выбором; В) сортировкой вставками.
- 1.21. Отсортировать элементы диагоналей вещественной матрицы, параллельных главной диагонали, по не возрастанию: А) сортировкой выбором; В) сортировкой вставками.
- 1.22. Отсортировать элементы нечетных строк вещественной матрицы по не убыванию, а элементы четных строк по не возрастанию: А) сортировкой выбором; В) сортировкой вставками.
- 1.23. Отсортировать элементы нечетных строк вещественной матрицы по не убыванию, а элементы четных строк по не возрастанию: А) сортировкой вставками; В) сортировкой обменом.
- 1.24. Отсортировать элементы нечетных столбцов вещественной матрицы по не убыванию, а элементы четных столбцов по не возрастанию: А) сортировкой вставками; В) сортировкой обменом.
- 1.25. Отсортировать элементы нечетных столбцов вещественной матрицы по не убыванию, а элементы четных столбцов по не возрастанию: А) сортировкой вставками; В) сортировкой обменом.

Лабораторная работа №2 «Простые классы»

Задание 2.

- Разработать программу, реализующую простой класс на основе заданной структуры данных.
- Разработать набор **функций-аксессоров** вида **get**, **set** и **show**.
- Функция **main** должна иллюстрировать использование разработанного класса.
- Интерфейс класса необходимо разместить в заголовочном файле, а реализацию класса и главную функцию – в двух отдельных файлах.

Указания к решению (вариант 30)

Буддийские монастыри Японии периода Нара			
Название	Школа	Количество монахов	Площадь земли (га)
Тодайдзи	Т	220	368.8
Якусидзи	С	50	54.7
Дайандзи	Д	10	12.2
Примечание: Т - Тэндай; С - Сингон; Д - Дзедзицу			

1. Описание полей класса

Начнем разработку класса с выбора типа данных для его полей. В соответствии с заданием наш класс должен иметь следующие поля: *Название*, *Школа*, *Количество монахов*, *Площадь земли*.

Название. Тип этого поля необходимо определить как **char ***, так как это позволит размещать строки различной длины, не обращая внимания на возможность выхода за границы массива.

Школа. Данное поле будет хранить всего лишь один символ, поэтому его тип мы определим как **char**.

Количество монахов. Для хранения данных данного поля достаточно типа **int** и так как эти данные не могут принимать отрицательные значения, то для исключения ошибок тип определим как **unsigned int**.

Площадь земли. Тип данного поля определим как **float**.

2. Функции-аксессоры

В соответствии с условиями задания наш класс должен иметь три функции-аксессора: **set()**, **get()**, **show()**.

Начнем с функции **set()**. Задача этой функции - считать данные, вводимые пользователем с клавиатуры, и записать их в поля класса. Т.е для работы с функцией необходимо определить ряд переменных (параметров функции), типы которых соответствуют типам полей класса. При присваивании строковых полей необходимо воспользоваться функцией **strcpy()**, для чего не забыть в список используемых файлов библиотек подключить файл **<string.h>**.

Далее приступим к функции **get()**. Она отвечает за считывание значений из полей класса. Работа функции **get()** полностью аналогична работе функции **set()** за исключением того, что в роли приемника значений выступают переменные, а в роли передатчика - поля класса.

Необходимо обратить внимание на то, что для корректной работы со строковым полем требуется освободить память, занимаемую переменной, а затем выделить память размером с длину новой строки:

```
delete[] a; a = new char [strlen(name)+1];
```

Третья функция-аксессор **show()** производит вывод на экран значений полей класса и ее реализация не составляет особого труда (используйте потоки ввода-вывода).

2.1.

Некоторые виды антилоп			
Название	Группа	Место обитания	Численность популяции
Джейран	А	Азия	30000
Гну	В	Африка	560000
Бейза	Н	Африка	2500
Группы: А - настоящие антилопы, В - коровьи антилопы, Н - лошадиные антилопы			

2.2.

Фирмы - производители СУБД			
Фирма	Количество. продуктов	Годовой объем продажи (\$)	Часть рынка (%)
Oracle	1	2488000000	31.1
IBM	3	2392000000	29.9
Microsoft	2	1048000000	13.1

2.3.

Отдел кадров			
Фамилия	Инициалы	Год рожд	Оклад
Иванов	И.И.	1975	517.50
Петренко	П.П.	1956	219.10
Паниковский	М.С.	1967	300.00

2.4.

Ведомость деталей			
Наименование	Тип	Количество	Вес 1 детали (г)
Фланец	З	3	450
Переходник	П	8	74
Станина	О	1	117050
Примечание: принято такое кодирование типов: О - оригинальная, П - покупная, З - заимствованная			

2.5.

Характеристики ПЭВМ			
Процессор	Частота (Mgz)	RAM (Mb)	Тип
Pentium-III	233		С
AMD-K6	166		С
PowerPC-620	2000		R
Тип: С - CISC-процессор, R - RISC-процессор			

2.6.

Каталог библиотеки			
Автор книги	Название	Год выпуска	Группа
Сенкевич	Потоп	1978	Х
Ландау	Механика	1989	У
Дойль	Сумчатые	1990	С
Примечание: Х - художественная литература; У - учебная литература; С - справочная литература			

2.7.

Ведомость комплектующих			
Обозначение	Тип	Номинал	Количество
RT-11-24	R	100000	12
RT-11-24	R	50000	10
CGU-12K	C	17.5	3
Примечание: R - резистор; C - конденсатор			

2.8.

Проекты поиска внеземных сигналов			
Год	Научный руководитель	Диаметр антенны (м)	Рабочая частота (МГц)
1960	Дрейк	26	1420
1970	Троицкий	14	1875
1978	Хоровиц	300	1665

2.9.

Офисные пакеты			
Наименование	Производитель	Количество сост. частей	Цена (\$)
Office	Microsoft	4	870
SmartSute	Lotus	5	1020
StarOffice	Sun	4	9

2.10.

Сельскохозяйственные культуры			
Наименование	Тип	Посевная площадь (га)	Урожайность (ц/га)
Соя	Б	13000	45
Чумиза	З	8000	17
Рис	З	25650	24
Примечание: З - зерновые, Б - бобовые			

2.11.

Ведомость спортивных состязаний			
Фамилия участника	Код команды	Количество баллов	Место в итоге
Баландин	С	123.7	2
Шишков	Ш	79.98	3
Кравченко	Д	134.8	1
Примечание: Д – «Динамо», С – «Спартак», Ш – «Шахтер»			

2.12.

Ведомость общественного транспорта			
Вид транспорта	№ маршрута	Протяженность маршрута (км)	Время в дороге (мин)
Тр	12	27.55	75
Т-с	17	13.6	57
А	12а	57.3	117
Примечание: Тр - трамвай, Тс - троллейбус, А - автобус			

2.13.

Приблизительное количество звезд разных спектральных классов в Галактике			
Спектральный класс	Приблизительная масса (отн.Солнца)	Часть %	Численность
О	32	0.00002	55000
F	1.25	2.9	12000000000
M	0.2	73.2	293000000000

2.14.

Время выполнения некоторых программ, которые применяют параллельные алгоритмы			
Название программы	Количество строк кода	Время исп. на SGI Challenge (сек)	Время исп. на SGI Indy (сек)
OCEAN	1917	8.70	18.2
DYFESM	3386	6.95	22
TRFD	417	1.05	2.98

2.15.

Конфигурация программных средств информационных систем				
Операционная система	СУБД	Мин.объем внешней памяти (МВ)	Мин.объем оперативной памяти (МВ)	Приблизительная цена (\$)
OS/2	DB2	130	22	3343
Windows/NT	SQLServer	230	24	2685
SCO/Unix	Oracle	110	48	3745

2.16.

Сельскохозяйственные предприятия			
Название	Вид собственности	Площадь земли (га)	Кол. работников
Заря	Д	300	120
Росинка	К	174	27
Петренко	Ч	56	6
Вид собственности: Д - государственная, Ч - частная, К - кооперативная			

2.17.

Прайс-лист			
Наименование товара	Тип товара	Цена за 1 шт (руб)	Минимальное количество в партии
Папка	К	4.75	4
Бумага	К	13.90	10
Калькулятор	О	411.00	1
Примечание: К - канцтовары, О - оргтехника			

2.18.

Вязкость металлов в жидком состоянии			
Вещество	Атомный номер	Температура (град.С)	Вязкость (кг/м*сек)
Алюминий	13	700	2.90
Висмут	83	304	1.65
Свинец	82	441	2.11

2.19.

Коэффициенты теплопроводности материалов			
Вещество	Тип	Влажность (%)	Коэффициент
Алюминий	М	0-100	209.3
Стекловата	Т	0-100	0.035
Глина	Д	15-20	0.73

Примечание: М - металлы, Т - термоизоляционные материалы, Д - другие материалы

2.20.

Скорость звука в жидкостях			
Вещество	Тип	Температура (град.С)	Скорость (м/сек)
Анилин	Ч	20	1656
Ртуть	Ч	20	1451
Кедровое	М	29	1406

Тип жидкости: Ч - чистое вещество, М - масло

2.21.

Температура перехода веществ в сверхпроводниковое состояние		
Вещество	Тип	Температура
Zn	М	0.8-0.8
Pb-Au	П	2.0-7.3
NbC	С	10.1-10.5

Тип вещества: М - металл, П - сплав, С - соединение

2.22.

Сплавы с высоким сопротивлением			
Сплав	Сопротивление	Темп.коэфф.сопр	Макс.температура
Константан	0.44	0.00001	500
Никелин	0.39	0.39	150
Фехраль	1.1	0.0001	900

2.23.

Свойства полупроводников			
Вещество	Ширина запрещенной зоны	Движение электронов	Движение дырок
Бор	1.1	10	10
Телур	0.36	1700	1200
CdTe	1.45	450	100

2.24.

Элементарные частицы			
Частица	Группа	Заряд	Масса покоя
Нейтрон	Н	0	940
Ка-плюс	М	+1	494
Электрон	Л	-1	0.511

Группы частиц: Г - гипероны, Н - нуклоны, М - мезоны, Л - лептоны

2.25.

Искусственно полученные элементы			
Атомный номер	Название	Массовое число	Период полураспада (лет)
93	Нептуний	237	220000
97	Берклий	247	700
98	Калифорний	251	660

2.26.

Японские острова			
Остров	Площадь (кв.км)	Кол. малых островов	Протяженность береговой линии (км)
Хонсю	230500	192	11875
Хоккайдо	78500	75	3072
Сикоку	18800	13	2946

2.27.

Прайс-лист			
Код товара	Категория	Цена (руб)	Срок поставки (дн)
01-0018	К	11282	7
01-0033	С	170	1
02-0170	М	1735	14

Примечание: К - компьютеры, С - комплектующие, М - множительная техника

2.28.

Журнал регистрации аварий:			
Название предприятия	Категория	Утраты от аварии (тыс.руб)	Срок простоя (дн)
Птицефабрика	1	12.7	4
Теплоцентраль	1	0.711	1
Автосервис	2	1770.15	38
Примечание: указываются только прямые затраты, связанные с ликвидацией последствий аварии			

2.29.

Ведомости о присвоении квалификации			
Фамилия	Год рождения	Оценка результатов экзаменов	Присвоен уровень квалификации
Шевченко	1970	4.33	6
Розанов	1975	5.1	5
Панасюк	1966	3.07	3
Примечание: показана итоговая оценка по результатам экзаменов по теоретической и практической подготовке			

2.30.

Буддийские монастыри Японии периода Нара			
Название	Школа	Количество монахов	Площадь земли (га)
Тодайдзи	Т	220	368.8
Якусидзи	С	50	54.7
Дайандзи	Д	10	12.2
Примечание: Т - Тэндай; С - Сингон; Д - Дзедзицу			

Лабораторная работа № «Простые классы»**Задание 3.**

3.1. Реализуйте класс, содержащий следующие методы работы с текстовым файлом:

- 1) Открыть файл с расширением .txt
- 2) Создать новый файл с тем же именем и расширением .dat
- 3) Записать в новый файл слова из старого файла в алфавитном порядке.
- 4) Закрыть оба файла.

3.2. Реализуйте класс, содержащий следующие методы работы с текстовым файлом:

- 1) Открыть файл с расширением .txt
- 2) Создать новый файл с тем же именем и расширением .html
- 3) Переписать в новый файл содержимое старого, поставив в конце каждой строки символы
, а в начале каждого абзаца символы <p> (абзац определяется как новая строка, начинающаяся с одного или более пробелов)
- 4) Закрыть оба файла.

3.3. Реализуйте класс, содержащий следующие методы работы с текстовым файлом:

- 1) Открыть файл с расширением .txt
- 2) Создать новый файл с тем же именем и расширением .dat
- 3) Записать в новый файл последовательность нулей и единиц, соответствующую ASCII кодам символов исходного файла в двоичной системе счисления. Между кодами соседних символов пробелы.
- 4) Закрыть оба файла.

3.4. Реализуйте класс, содержащий следующие методы работы с текстовым файлом:

- 1) Открыть файл с расширением .txt
- 2) Создать новый файл с тем же именем и расширением .dat
- 3) Рассматривая исходный файл как последовательность битов записать в новый файл только четные биты (должен получиться файл из нулей и единиц).
- 4) Закрыть оба файла.

3.5. Реализуйте класс, содержащий следующие методы работы с текстовым файлом:

- 1) Открыть файл с расширением .txt
- 2) Создать новый файл с тем же именем и расширением .dat
- 3) Записать в новый файл из исходного буквы, ASCII коды которых в двоичной системе счисления содержат четное число единиц.
- 4) Закрыть оба файла.

3.6. Реализуйте класс, содержащий следующие методы работы с текстовым файлом:

- 1) Открыть файл с расширением .txt
- 2) Создать новый файл с тем же именем и расширением .dat
- 3) Записать в новый файл символы из исходного, преобразованные следующим образом: берется ASCII код символа в двоичной системе счисления и проводится инверсия, то есть нули заменяются единицами и наоборот, затем по полученному коду определяется соответствующий символ.
- 4) Закрыть оба файла.

3.7. Реализуйте класс, содержащий следующие методы работы с текстовым файлом:

- 1) Открыть файл с расширением .txt
- 2) Создать новый файл с тем же именем и расширением .dat

3) Записать в новый файл символы исходного файла, проведя перед этим операцию логического побитового сложения с заданным ключом.

4) Закрывать оба файла.

3.8. Реализуйте класс, содержащий следующие методы работы с текстовым файлом:

1) Открыть файл с расширением .txt

2) Создать новый файл с тем же именем и расширением .dat

3) Записать в новый файл символы исходного файла, проведя перед этим операцию логического побитового умножения с заданным ключом.

4) Закрывать оба файла.

3.9. Реализуйте класс, содержащий следующие методы работы с текстовым файлом:

1) Открыть файл с расширением .txt

2) Создать новый файл с тем же именем и расширением .dat

3) Записать в новый файл символы исходного файла, проведя перед этим операцию логического побитового исключающего или с заданным ключом.

4) Закрывать оба файла.

3.10. Реализуйте класс, содержащий следующие методы работы с текстовым файлом:

1) Открыть файл с расширением .txt

2) Создать новый файл с тем же именем и расширением .dat

3) Записать в новый файл из исходного буквы, ASCII коды которых в двоичной системе счисления содержат не меньше трех единиц.

4) Закрывать оба файла.

Лабораторная работа №4 «Перегрузка операций»

Задание 4. В классе, реализованном в задании №2, необходимо перегрузить нижеследующие операции, при этом две функции-операции должны быть реализованы как *методы* класса, а одна – как *дружественная функция*:

- Присваивание «=»;
- Сложение (в случае символьных полей сложение понимается как конкатенация) «+»;
- Соответствие (равенство) «==».

Указания к решению (вариант 30)

Напомним, что класс, описывающий *Буддистские монастыри Японии* имеет следующие поля:

Название. `char *name;`

Школа. `char *school;` Тип этого поля претерпел изменения, вызванные новыми условиями. На первый взгляд, тип этого поля можно оставить прежним `char`. Но на самом деле оказывается, что для выполнения перегрузки оператора суммы и корректного выполнения программы, тип данного поля необходимо определить как `char *`. Это можно объяснить тем, что при перегрузке оператора суммы необходимо произвести конкатенацию символьных полей.

Количество монахов. `unsigned int count;`

Площадь земли. `float square;`

Пусть разрабатываемый класс, назовем его **Church**, должен иметь следующие функции-операции, являющиеся методами:

1. `Church& operator = (Church&);`

Данная функция производит перегрузку оператора присваивания, то есть выполняет присваивание полей одного объекта класса полям другого объекта этого же класса. Как и в задании №2, для корректной работы со строками требуется динамическое выделение памяти и ее освобождение.

2. `int operator == (Church&);`

Функция позволяет установить соответствие между экземплярами одного класса. Ее тело включает проверку на равенство значений числовых полей и сравнение содержимого строк, которое выполняется с помощью функции `strcmp`. В зависимости от того, тождественны ли экземпляры функцией, возвращается 0 или 1.

```
int Church::operator == (Church & c) {
    if ((count != c.count) ||
        (ceil(square) != ceil(c.square)) ||
        (strcmp(name, c.name) != 0) ||
        (strcmp(school, c.school) != 0))
        return 0; //ложь
    else
        return 1; //истина
}
```

и функцию-операцию, являющуюся дружественной функцией:

3. `friend Church operator + (Church&, Church&);`

Оператор + необходимо перегрузить способом, аналогичным своему традиционному арифметическому использованию. Поэтому важно, чтобы ни один из операндов не менялся. Например, когда вы складываете 10+4, результат равен 14, но ни 10, ни 4 не меняются. Таким образом, внутри функции необходим временный объект для хранения результата.

Лабораторная работа №5 «Перегрузка операций»

Задание 5.

- Реализовать заданную динамическую структуру данных, с которой можно работать через перегруженные операции.
- Для демонстрации работы программы необходимо реализовать меню, позволяющее вызывать операции реализованной структуры данных. На экране должна отображаться вся последовательность введенных элементов, над которыми производятся действия.

5.1. Создайте класс, реализующий стек. В интерфейс класса должны входить перегруженные операции:

- 1) положить в стек (<<);
- 2) взять из стека (>>).

5.2. Создайте класс, реализующий стек. В интерфейс класса должны входить перегруженные операции:

- 1) положить в стек (+);
- 2) взять из стека (-).

5.3. Создайте класс, реализующий однонаправленную очередь. В интерфейс класса должны входить перегруженные операции:

- 1) добавить элемент в конец очереди (<<);
- 2) извлечь элемент из начала очереди (>>).

5.4. Создайте класс, реализующий однонаправленную очередь. В интерфейс класса должны входить перегруженные операции:

- 1) добавить элемент в конец очереди (+);
- 2) извлечь элемент из начала очереди (-).

5.5. Создайте класс, реализующий двунаправленную очередь. В интерфейс класса должны входить перегруженные операции:

- 1) добавить элемент в конец очереди (>=);
- 2) добавить элемент в начало очереди (<=);
- 3) извлечь элемент из начала очереди (<<);
- 4) извлечь элемент из конца очереди (>>).

5.6. Создайте класс, реализующий двунаправленную очередь. В интерфейс класса должны входить перегруженные операции:

- 1) добавить элемент в конец очереди (+);
- 2) добавить элемент в начало очереди (<<);
- 3) извлечь элемент из начала очереди (>>);
- 4) извлечь элемент из конца очереди (-).

5.7. Создайте класс, реализующий двунаправленный список. В интерфейс класса должны входить перегруженные операции:

- 1) переместиться в начало списка (--);
- 2) переместиться в конец списка (++);
- 3) передвинуть указатель на один элемент вправо (>);
- 4) передвинуть указатель на один элемент влево (<);
- 5) добавить элемент в текущее положение указателя (+);
- 6) удалить элемент по текущему положению указателя (-).

5.8. Создайте класс, реализующий двунаправленный список. В интерфейс класса должны входить перегруженные операции:

- 1) переместиться в начало списка (++);
- 2) переместиться в конец списка (--);
- 3) передвинуть указатель на один элемент вправо (>=);
- 4) передвинуть указатель на один элемент влево (<=);
- 5) добавить элемент в текущее положение указателя (<<);
- 6) удалить элемент по текущему положению указателя (>>).

5.9. Создайте класс, реализующий дерево поиска. В интерфейс класса должны входить перегруженные операции:

- 5) добавить элемент (+);
- 6) удалить элемент (-).

5.10. Создайте класс, реализующий дерево поиска. В интерфейс класса должны входить перегруженные операции:

- 5) добавить элемент (<<);
- 6) удалить элемент (>>).

Лабораторная работа №6 «Наследование»

Задание 6.

- И базовый, и производный классы должны иметь как минимум конструктор по умолчанию, конструктор копирования, деструктор (если необходимо), перегруженную операцию присваивания.
- Где это возможно, необходим вызов функций (методов, конструкторов) базового класса, а не копирование фрагментов кода.

6.1. Реализуйте однонаправленный список как класс. Используя механизм наследования, реализуйте на базе списка ежедневник на текущий год с полями «Число», «Месяц», «День недели», «Дела на день». Интерфейс должен позволять читать и редактировать запись «Дела на день» по дате.

6.2. Реализуйте однонаправленный список как класс. Используя механизм наследования, реализуйте на базе списка расписание занятий 1 курса факультета компьютерных наук с полями «День недели», «Номер пары», «Название курса». Интерфейс должен позволять просматривать все расписание на неделю, на отдельный день и редактировать поле «название курса».

6.3. Реализуйте абстрактную структуру данных «множество» как класс. Должны быть описаны методы проверки принадлежности элемента множеству, объединения и пересечения множеств. Используя множество английских слов с их переводом на русский язык, хранящихся в файле-словаре и механизм наследования, реализуйте на базе множества подстрочный переводчик с английского языка. Словарь английских слов (не меньше 20) сделайте самостоятельно в отдельном файле. Переводчик должен сопоставлять строке английских слов строку русских слов, если в файле-словаре нет соответствующего английского слова, необходимо выводить русское слово без изменений. Переводчик, как минимум, должен справляться с фразой «Мама мыла раму».

6.4. Реализуйте однонаправленный список как класс. Используя механизм наследования, реализуйте на базе списка программу контроля автобусного парка. Программа должна содержать две базы данных с полями «Номер автобуса», «Водитель», «Номер маршрута». Первая база данных контролирует автобусы, находящиеся в данный момент в гараже, вторая – на маршруте. При выезде из гаража вводится номер автобуса, и соответствующая запись переносится в другую базу данных, аналогичная операция производится при возвращении автобуса с маршрута. Список всех имеющихся автобусов хранится в файле, в начале рабочего дня все автобусы находятся в гараже.

6.5. Реализуйте однонаправленный список как класс. Используя механизм наследования, реализуйте на базе списка систему бронирования авиабилетов с полями «Номер рейса», «Дата вылета», «Номер места», «Пассажир». У непроданных билетов поле пассажир пустое. Если билет продан, то в поле «Пассажир» вносится фамилия пассажира, купившего билет. Интерфейс должен позволять бронировать билеты на ближайшие 3 дня, а также отказываться от ранее забронированных билетов. Для впервые запрашиваемого рейса заводится отдельный файл. Для уже запрашивавшихся рейсов загружаются файлы. Файлы уже отправленных рейсов уничтожаются.

6.6. Реализуйте однонаправленный список как класс. Используя механизм наследования, реализуйте на базе списка программу, которая содержит текущую информацию о книгах в библиотеке. Сведения о книгах содержат фамилию и инициалы автора, название, год издания, количество экземпляров в библиотеке. Программа должна обеспечивать:

- 1) начальное формирование данных обо всех книгах в виде списка;
- 2) уменьшать количество экземпляров книги при выдаче одного экземпляра на руки читателю, и увеличивать при возврате.
- 3) осуществлять поиск по маске «фамилия автора».

6.7. Реализуйте однонаправленный список как класс. Используя механизм наследования, реализуйте на базе списка каталог файлов с полями:

- 1) имя файла;
- 2) дата создания.

Программа должна обеспечивать:

- 1) начальное формирование каталога файлов;
- 2) вывод каталога файлов;
- 3) удаление файлов, «время жизни» которых больше определенного срока.

6.8. Реализуйте однонаправленный список как класс. Используя механизм наследования, реализуйте на базе списка бюро обмена квартир. Сведения о каждой квартире содержат:

- 1) количество комнат;
- 2) этаж;
- 3) площадь;
- 4) адрес.

Программа должна обеспечивать:

- 1) начальное формирование картотеки и её пополнение;
- 2) поиск в картотеке подходящего варианта по количеству комнат.
- 3) вывод всего списка.

6.9. Реализуйте однонаправленный список как класс. Используя механизм наследования, реализуйте на базе списка картотеку абонентов междугородней телефонной станции, содержащей сведения о телефонах и их владельцах. Программа должна обеспечивать:

- 1) начальное формирование картотеки;
- 2) вывод всей картотеки;
- 3) ввод номера телефона и времени разговора;
- 4) вывод извещения об оплате, содержащего фамилию и сумму.

6.10. Реализуйте однонаправленный список как класс. Используя механизм наследования, реализуйте на базе списка автоматизированную информационную систему железнодорожного вокзала, содержащую сведения об отправлении поездов дальнего следования. Для каждого поезда указывается:

- 1) номер поезда;
- 2) станция назначения;
- 3) время отправления.

Программа должна обеспечивать:

- 1) первоначальный ввод данных и формирование списка по времени отправления;
- 2) вывод всего списка;
- 3) вывод по номеру поезда всех данных о нем;
- 4) вывод информации обо всех поездах до данной станции назначения.

6.11. Реализуйте класс, позволяющий работать с абстрактной структурой данных «множество». Должны быть описаны методы проверки принадлежности элемента множеству, объединения и пересечения множеств. Используя механизм наследования, реализуйте на базе множества ежедневник на текущий год с полями «Число», «Месяц», «День недели», «Дела на день». Интерфейс должен позволять читать и редактировать запись «Дела на день» по дате.

6.12. Реализуйте класс, позволяющий работать с абстрактной структурой данных «множество». Должны быть описаны методы проверки принадлежности элемента множеству, объединения и пересечения множеств. Используя механизм наследования, реализуйте на базе множества расписание занятий 1 курса факультета компьютерных наук с полями «День недели», «Номер пары», «Название курса». Интерфейс должен позволять просматривать все расписание на неделю, на отдельный день и редактировать поле «название курса».

6.13. Реализуйте класс, позволяющий работать с абстрактной структурой данных «множество». Должны быть описаны методы проверки принадлежности элемента множеству, объединения и пересечения множеств. Используя механизм наследования, реализуйте на базе множества программу контроля автобусного парка. Программа должна содержать две базы данных с полями «Номер автобуса», «Водитель», «Номер маршрута». Первая база данных контролирует автобусы, находящиеся в данный момент в гараже, вторая – на маршруте. При выезде из гаража вводится номер автобуса, и соответствующая запись переносится в другую базу данных, аналогичная операция производится при возвращении автобуса с маршрута. Список всех имеющихся автобусов хранится в файле, в начале рабочего дня все автобусы находятся в гараже.

6.14. Реализуйте класс, позволяющий работать с абстрактной структурой данных «множество». Должны быть описаны методы проверки принадлежности элемента множеству, объединения и пересечения множеств. Используя механизм наследования, реализуйте на базе множества систему бронирования авиабилетов с полями «Номер рейса», «Дата вылета», «Номер места», «Пассажир». У не проданных билетов поле пассажир пустое. Если билет продан, то в поле «Пассажир» вносится фамилия пассажира, купившего билет. Интерфейс должен позволять бронировать билеты на ближайшие 3 дня, а также отказываться от ранее забронированных билетов. Для впервые запрашиваемого рейса заводится отдельный файл. Для уже запрашивавшихся рейсов загружаются файлы. Файлы уже отправленных рейсов уничтожаются.

6.15. Реализуйте класс, позволяющий работать с абстрактной структурой данных «множество». Должны быть описаны методы проверки принадлежности элемента множеству, объединения и пересечения множеств. Используя механизм наследования, реализуйте на базе множества программу, которая содержит текущую информацию о книгах в библиотеке. Сведения о книгах содержат фамилию и инициалы автора, название, год издания, количество экземпляров в библиотеке. Программа должна обеспечивать:

- 1) начальное формирование данных обо всех книгах в виде списка;
- 2) уменьшать количество экземпляров книги при выдаче одного экземпляра на руки читателю, и увеличивать при возврате;
- 3) осуществлять поиск по маске фамилии автора.

6.16. Реализуйте класс, позволяющий работать с абстрактной структурой данных «множество». Должны быть описаны методы проверки принадлежности элемента множеству, объединения и пересечения множеств. Используя механизм наследования, реализуйте на базе множества каталог файлов с полями:

- 1) имя файла;
- 2) дата создания;

Программа должна обеспечивать:

- 1) начальное формирование каталога файлов;
- 2) вывод каталога файлов;
- 3) удаление файлов, «время жизни» которых больше определенного срока.

6.17. Реализуйте класс, позволяющий работать с абстрактной структурой данных «множество». Должны быть описаны методы проверки принадлежности элемента множеству, объединения и пересечения множеств. Используя механизм наследования, реализуйте на базе множества бюро обмена квартир. Сведения о каждой квартире содержат:

- 1) количество комнат;
- 2) этаж;
- 3) площадь;
- 4) адрес.

Программа должна обеспечивать:

- 1) начальное формирование картотеки и её пополнение;
- 2) поиск в картотеке подходящего варианта по количеству комнат.
- 3) вывод всего списка.

6.18. Реализуйте класс, позволяющий работать с абстрактной структурой данных «множество». Должны быть описаны методы проверки принадлежности элемента множеству, объединения и пересечения множеств. Используя механизм наследования, реализуйте на базе множества картотеку абонентов междугородней телефонной станции, содержащей сведения о телефонах и их владельцах.

Программа должна обеспечивать:

- 1) начальное формирование картотеки;
- 2) вывод всей картотеки;
- 3) ввод номера телефона и времени разговора;
- 4) вывод извещения об оплате, содержащего фамилию и сумму.

6.19. Реализуйте класс, позволяющий работать с абстрактной структурой данных «множество». Должны быть описаны методы проверки принадлежности элемента множеству, объединения и пересечения множеств. Используя механизм наследования, реализуйте на базе множества автоматизированную информационную систему железнодорожного вокзала, содержащую сведения об отправлении поездов дальнего следования. Для каждого поезда указывается:

- 1) номер поезда;
- 2) станция назначения;
- 3) время отправления.

Программа должна обеспечивать:

- 1) первоначальный ввод данных и формирование списка по времени отправления;
- 2) вывод всего списка;
- 3) вывод по номеру поезда всех данных о нем;
- 4) вывод информации обо всех поездах до данной станции назначения.

6.20. Реализуйте двунаправленную очередь как класс. Используя механизм наследования, реализуйте на базе очереди ежедневник на текущий год с полями «Число», «Месяц», «День недели», «Дела на день». Интерфейс должен позволять читать и редактировать запись «Дела на день» по дате.

6.21. Реализуйте двунаправленную очередь как класс. Используя механизм наследования, реализуйте на базе очереди расписание занятий 1 курса факультета компьютерных наук с полями «День недели», «Номер пары», «Название курса». Интерфейс должен позволять просматривать все расписание на неделю, на отдельный день и редактировать поле «название курса».

6.22. Реализуйте двунаправленную очередь как класс. Используя механизм наследования, реализуйте на базе очереди программу контроля автобусного парка. Программа должна содержать две базы данных с полями «Номер автобуса», «Водитель», «Номер маршрута». Первая база данных контролирует автобусы, находящиеся в данный момент в гараже, вторая – на маршруте. При выезде из гаража вводится номер автобуса, и соответствующая запись переносится в другую базу данных, аналогичная операция производится при возвращении автобуса с маршрута. Список всех имеющихся автобусов хранится в файле, в начале рабочего дня все автобусы находятся в гараже.

6.23. Реализуйте двунаправленную очередь как класс. Используя механизм наследования, реализуйте на базе очереди картотеку абонентов междугородней телефонной станции, содержащей сведения о телефонах и их владельцах.

Программа должна обеспечивать:

- 1) начальное формирование картотеки;
- 2) вывод всей картотеки;
- 3) ввод номера телефона и времени разговора;
- 4) вывод извещения об оплате, содержащего фамилию и сумму.

6.24. Реализуйте двунаправленную очередь как класс. Используя механизм наследования, реализуйте на базе очереди каталог файлов с полями:

- 1) имя файла; 2) дата создания;

Программа должна обеспечивать:

- 1) начальное формирование каталога файлов;
- 2) вывод каталога файлов;
- 3) удаление файлов, «время жизни» которых больше определенного срока.

6.25. Реализуйте двунаправленную очередь как класс. Используя механизм наследования, реализуйте на базе очереди автоматизированную информационную систему железнодорожного вокзала, содержащую сведения об отправлении поездов дальнего следования. Для каждого поезда указывается:

- 1) номер поезда;
- 2) станция назначения;
- 3) время отправления.

Программа должна обеспечивать:

- 1) первоначальный ввод данных и формирование списка по времени отправления;
- 2) вывод всего списка;
- 3) вывод по номеру поезда всех данных о нем;
- 4) вывод информации обо всех поездах до данной станции назначения.

Лабораторная работа №7 «Шаблоны классов»

Задание 7.

1. Создать шаблон класса для работы с массивом произвольного типа данных. Шаблон должен включать:

- указатель, хранящий адрес размещения массива в динамической памяти;
- целочисленную переменную, показывающую количество занятых элементов массива;
- конструктор без параметров с параметрами по умолчанию;
- конструктор копирования;
- метод «обработка массива» (на начальном этапе метод выводит на консоль какое-либо сообщение);
- деструктор

и другие необходимые для работы методы (например, перегрузка операций индексирования и присваивания, методы ввода / вывода и т.д.).

2. Предусмотреть генерацию и обработку исключений для возможных ошибочных ситуаций (отрицательная размерность при задании массива, выход за границы диапазона при обращении по индексу и т.д.).

3. В функции **main()** показать использование созданного класса, включая ситуации, приводящие к генерации исключений; показать **инстанцирование** шаблона для различных типов.

4. На основе данного шаблона создать класс для работы со строкой символов, **специализировав** метод «обработка массива» для вашей конкретной задачи.

7.1. Дана строка латинских символов. Преобразовать строку, заменив все символы на заглавные.

7.2. Дана строка латинских символов. Преобразовать строку, заменив все символы на строчные.

7.3. Дана строка латинских символов. Преобразовать строку, разместив в алфавитном порядке все различные строчные латинские буквы, входящие в исходную строку.

7.4. Дана строка латинских символов. Преобразовать строку, заменив все вхождения «abc» на «def».

7.5. Дана строка латинских символов. Преобразовать строку, удалив первое вхождение «w», если такое есть (образовавшуюся «дыру» заполнить последующими буквами, а в конец добавить пробел).

7.6. Дана строка латинских символов. Преобразовать строку, заменив на «ks» первое вхождение «x», если оно есть.

7.7. Дана строка латинских символов. Преобразовать строку в эту же последовательность символов, но в обратном порядке.

7.8. Дана строка латинских символов. Преобразовать строку, удалив из нее повторно входящие символы.

7.9. Дана строка латинских символов. Преобразовать строку, оставив символы, которые встречаются в последовательности по одному разу.

7.10. Дана строка латинских символов. Преобразовать строку, оставив все различные символы, указав для каждого из них число его вхождений в исходную строку.

7.11. Дана строка латинских символов. Преобразовать строку, разместив символы в алфавитном порядке, разделяя их пробелами.

7.12. Дана строка латинских символов. Преобразовать строку, оставив символы, которые являются строчными.

- 7.13. Дана строка латинских символов. Преобразовать строку, оставив символы, которые являются заглавными.
- 7.14. Дана строка латинских символов. Преобразовать строку, оставив символы, отличающиеся от последнего.
- 7.15. Дана строка латинских символов. Преобразовать строку, удалив из нее символы, встречающиеся ровно два раза.
- 7.16. Дана строка латинских символов. Преобразовать строку, оставив символы, отличающиеся от первого.
- 7.17. Дана строка латинских символов. Преобразовать строку, разместив символы в алфавитном порядке, разделяя их запятыми.
- 7.18. Дана строка латинских символов. Преобразовать строку, заменив на «ks» все вхождения «x», если они есть.
- 7.19. Дана строка латинских символов. Преобразовать строку, заменив на «ww» все вхождения «www», если они есть.
- 7.20. Дана строка латинских символов. Преобразовать строку, заменив на «w» первое вхождение «www», если оно есть.

Лабораторная работа №7 «Шаблоны классов»

Задание 8.

1. Создать шаблон класса, реализованного в задании №5 для работы со структурой произвольного типа данных.
2. Предусмотреть генерацию и обработку исключений для возможных ошибочных ситуаций (удаление из пустой структуры, смещение за пределы структуры и т.п.).
3. В функции `main()` показать использование созданного класса, включая ситуации, приводящие к генерации исключений; показать *инстанцирование* шаблона для различных типов (`int`, `float`, `string`).

Лабораторная работа №9 «Потоковые классы»

Задание 9.

- 9.1. Реализуйте модуль, подключение которого перегружает операции помещения в поток вывода и извлечение из потока ввода для класса матриц размером 3×3 . Данный класс также описывается в указанном модуле и содержит перегруженные операции сложения и умножения матриц.
- 9.2. Реализуйте модуль, подключение которого перегружает операции помещения в поток вывода и извлечение из потока ввода для класса полиномов степени 5. Данный класс также описывается в указанном модуле и содержит перегруженные операции сложения и умножения полиномов.
- 9.3. Реализуйте модуль, подключение которого перегружает операции помещения в поток вывода и извлечение из потока ввода для класса, содержащего методы разложения функции $\sin(x)$ в ряд Тейлора. На экран должно выводиться n первых слагаемых разложения ($x - 0.16667x^3 + \dots$). Значение n определяет пользователь.
- 9.4. Реализуйте модуль, подключение которого перегружает операции помещения в поток вывода и извлечение из потока ввода для класса, содержащего методы разложения функции $\cos(x)$ в ряд Тейлора. На экран должно выводиться n первых слагаемых разложения. Значение n определяет пользователь.
- 9.5. Реализуйте модуль, подключение которого перегружает операции помещения в поток вывода и извлечение из потока ввода для класса, содержащего методы разложения функции $\sin^2(x)$ в ряд Тейлора. На экран должно выводиться n первых слагаемых разложения. Значение n определяет пользователь.
- 9.6. Реализуйте модуль, подключение которого перегружает операции помещения в поток вывода и извлечение из потока ввода для класса, содержащего методы разложения функции $\cos^2(x)$ в ряд Тейлора. На экран должно выводиться n первых слагаемых разложения. Значение n определяет пользователь.
- 9.7. Реализуйте модуль, подключение которого перегружает операции помещения в поток вывода и извлечение из потока ввода для класса, содержащего методы разложения функции $\exp(x)$ в ряд Тейлора. На экран должно выводиться n первых слагаемых разложения. Значение n определяет пользователь.
- 9.8. Реализуйте модуль, подключение которого перегружает операции помещения в поток вывода и извлечение из потока ввода для класса, содержащего методы разложения функции $\sin(ax)$ в ряд Тейлора. На экран должно выводиться n первых слагаемых разложения. Значение n определяет пользователь.
- 9.9. Реализуйте модуль, подключение которого перегружает операции помещения в поток вывода и извлечение из потока ввода для класса, содержащего методы разложения функции $\cos(ax)$ в ряд Тейлора. На экран должно выводиться n первых слагаемых разложения. Значения a и n определяет пользователь.
- 9.10. Реализуйте модуль, подключение которого перегружает операции помещения в поток вывода и извлечение из потока ввода для класса, содержащего методы разложения функции $\exp(ax)$ в ряд Тейлора. На экран должно выводиться n первых слагаемых разложения. Значения a и n определяет пользователь.
- 9.11. Реализуйте модуль, подключение которого перегружает операции помещения в поток вывода и извлечение из потока ввода для класса, содержащего методы разложения функции $\ln(1+x)$ в ряд Тейлора. На экран должно выводиться n первых слагаемых разложения. Значение n определяет пользователь.

- [illegible]