

**Федеральное агентство по образованию РФ  
Омский государственный университет им. Ф.М. Достоевского  
Факультет компьютерных наук  
Кафедра вычислительных систем**

**Задания по курсу «Программирование  
на языке высокого уровня»  
(2 семестр)**

**Омск - 2006**

**Белим С.Ю.** Задания по курсу «Программирование на языке высокого уровня»(2 семестр): Методические указания. – Омск: Омск. гос. ун-т, 2006. –36 с.

Методические указания составлены для практических занятий по курсу «Программирование на языке высокого уровня». Задания разбиты на семь тем. По каждой теме указаны требования к выполнению заданий.

Издание может быть использовано для подготовки специалистов по специальности – «Вычислительные машины, комплексы, системы и сети».

Рекомендовано к изданию учебно методической комиссией факультета компьютерных наук ОмГУ им. Ф.М. Достоевского.

## Тема 1. Деревья

*Программа у пользователя запрашивает узлы дерева в порядке прямого обхода (с использованием терминов «корень», «число сыновей», «i-ый сын j- того узла»).*

- 1.1. Напишите программу преобразования данного списка узлов дерева в список, составленный при обходе в обратном порядке, используя стек узлов, хранящий путь от корня до родителя текущего узла.
- 1.2. Напишите программу преобразования данного списка узлов дерева в список, составленный при обходе в обратном порядке, используя рекурсию.
- 1.3. Напишите программу преобразования данного списка узлов дерева в список, составленный при обходе в симметричном порядке, используя стек узлов, хранящий путь от корня до родителя текущего узла.
- 1.4. Напишите программу преобразования данного списка узлов дерева в список, составленный при обходе в симметричном порядке, используя рекурсию.
- 1.5. Напишите программу преобразования данного списка узлов дерева в список, составленный при обходе в порядке уровней, используя стек узлов, хранящий путь от корня до родителя текущего узла. При прохождении дерева в порядке уровней в список узлов сначала заносится корень дерева, затем все узлы глубины 1 (слева направо), далее все узлы глубины 2 и т. д.
- 1.6. Напишите программу вычисления высоты дерева, используя стек узлов, хранящий путь от корня до родителя текущего узла.
- 1.7. Напишите программу вычисления арифметических выражений при обходе дерева в прямом порядке, используя стек узлов, хранящий путь от корня до родителя текущего узла.
- 1.8. Напишите программу вычисления арифметических выражений при обходе дерева в обратном порядке, используя стек узлов, хранящий путь от корня до родителя текущего узла.
- 1.9. Напишите программу, перечисляющую родителей всех узлов данного дерева.
- 1.10. Напишите программу, перечисляющую правых братьев всех узлов данного дерева.
- 1.11. Напишите программу, перечисляющую левых сыновей всех узлов данного дерева.

- 1.12. Напишите программу, определяющую является ли узел  $i$  предком узла  $j$ .
- 1.13. Напишите программу, определяющую является ли узел  $i$  потомком узла  $j$ .
- 1.14. Напишите программу, которая ищет узел по ключу и удаляет его из дерева (вместе со всеми его потомками). Вывод результата – в виде списка узлов нового дерева в порядке прямого обхода.
- 1.15. Напишите программу, которая ищет узел по его логическому номеру в порядке прямого обхода и удаляет его из дерева (вместе со всеми его потомками). Вывод результата – в виде списка узлов нового дерева в порядке прямого обхода.
- 1.16. Напишите программу, которая ищет узел по ключу, удаляет его из дерева, а на его место подставляет самого левого сына (в образовавшуюся дырку – снова самого левого сына). Вывод результата – в виде списка узлов нового дерева в порядке прямого обхода.
- 1.17. Напишите программу, которая составляет массив всех листов дерева.
- 1.18. Напишите программу, которая представляет операторы, выполняемые над деревом -  $PARENT(n, T)$ ,  $LEFTMOST\_CHILD(n, T)$ ,  $RIGHT\_SIBLING(n, T)$ . Всё должно быть реализовано с помощью указателей на родителей.
- 1.19. Напишите программу, которая представляет операторы, выполняемые над деревом -  $PARENT(n, T)$ ,  $LEFTMOST\_CHILD(n, T)$ ,  $RIGHT\_SIBLING(n, T)$ . Всё должно быть реализовано с помощью указателей на самого левого сына и на правого брата.
- 1.20. Напишите программу, которая представляет операторы, выполняемые над деревом -  $PARENT(n, T)$ ,  $LEFTMOST\_CHILD(n, T)$ ,  $RIGHT\_SIBLING(n, T)$ . Всё должно быть реализовано с помощью списка сыновей.
- 1.21. Напишите программу, которая представляет операторы, выполняемые над деревом -  $LABEL(n, T)$ ,  $CREATE_i(v, T_1, T_2)$ ,  $ROOT(T)$ . Всё должно быть реализовано с помощью указателей на родителей.
- 1.22. Напишите программу, которая представляет операторы, выполняемые над деревом -  $LABEL(n, T)$ ,  $CREATE_i(v, T_1, T_2)$ ,  $ROOT(T)$ . Всё должно быть реализовано с помощью указателей на самого левого сына и на правого брата.
- 1.23. Напишите программу, которая представляет операторы, выполняемые над деревом -  $LABEL(n, T)$ ,  $CREATE_i(v, T_1, T_2)$ ,  $ROOT(T)$ . Всё должно быть реализовано с помощью списка сыновей.

- 1.24. По заданной таблице вероятностей появления символов найдите код Хаффмана.
- 1.25. По заданной таблице вероятностей появления символов изобразите дерево Хаффмана (сверху вниз).

## **Тема 2. Простые методы сортировки** (включениями (вставками), обменом, выбором)

*Программа должна запрашивать из файла двумерный массив. Результат сортировки по желанию пользователя может сохраняться в файле. Необходимо разработать модуль, включающий в себя:*

- реализацию указанных алгоритмов сортировки;
- профайлер, позволяющий подсчитать количество пересылок и сравнений, произведенных во время сортировки.

*\*Кроме этого, для задач 2.13-2.25 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).*

**2-1.** Дана матрица, элементами которой являются латинские буквы. Отсортировать в лексикографическом порядке элементы каждой строки: А) сортировкой выбором; В) сортировкой обменом.

**2-2.** Дана матрица, элементами которой являются латинские буквы. Отсортировать в лексикографическом порядке элементы каждого столбца: А) сортировкой выбором; В) сортировкой обменом

**2-3.** Дана матрица, элементами которой являются латинские буквы. Отсортировать в лексикографическом порядке элементы каждой строки: А) сортировкой вставками; В) сортировкой обменом

**2-4.** Дана матрица, элементами которой являются латинские буквы. Отсортировать в лексикографическом порядке элементы каждого столбца: А) сортировкой вставками; В) сортировкой обменом

**2-5.** Дана матрица, элементами которой являются латинские буквы. Отсортировать в обратном лексикографическом порядке элементы каждой строки: А) сортировкой выбором; В) сортировкой вставками.

**2-6.** Дана матрица, элементами которой являются латинские буквы. Отсортировать в обратном лексикографическом порядке элементы каждого столбца: А) сортировкой выбором; В) сортировкой вставками.

**2-7.** Дана матрица, элементами которой являются латинские буквы. Отсортировать в обратном лексикографическом порядке элементы каждой строки: А) сортировкой выбором; В) сортировкой обменом

**2-8.** Дана матрица, элементами которой являются латинские буквы. Отсортировать в обратном лексикографическом порядке элементы каждого столбца: А) сортировкой выбором; В) сортировкой обменом.

**2-9.** Дана матрица, элементами которой являются латинские буквы. Отсортировать в лексикографическом порядке элементы каждой строки: А) сортировкой вставками; В) сортировкой обменом.

**2-10.** Дана матрица, элементами которой являются латинские буквы. Отсортировать в лексикографическом порядке элементы каждого столбца: А) сортировкой вставками; В) сортировкой обменом.

**2-11.** Отсортировать элементы диагоналей вещественной матрицы, параллельных главной диагонали, по не возрастанию: А) сортировкой вставками; В) сортировкой обменом.

**2-12.** Отсортировать элементы диагоналей вещественной матрицы, параллельных побочной диагонали, по не убыванию: А) сортировкой вставками; В) сортировкой обменом.

**2-13.** Отсортировать элементы строк вещественной матрицы по не возрастанию: А) сортировкой вставками; В) сортировкой обменом.

**2-14.** Отсортировать элементы столбцов вещественной матрицы по не возрастанию: А) сортировкой вставками; В) сортировкой обменом.

**2-15.** Отсортировать элементы строк вещественной матрицы по не возрастанию: А) сортировкой выбором; В) сортировкой обменом.

**2-16.** Отсортировать элементы столбцов вещественной матрицы по не возрастанию: А) сортировкой выбором; В) сортировкой обменом.

**2-17.** Отсортировать элементы строк вещественной матрицы по не убыванию: А) сортировкой выбором; В) сортировкой обменом.

**2-18.** Отсортировать элементы столбцов вещественной матрицы по не убыванию: А) сортировкой выбором; В) сортировкой обменом.

**2-19.** Отсортировать элементы строк вещественной матрицы по не убыванию: А) сортировкой выбором; В) сортировкой вставками.

**2-20.** Отсортировать элементы столбцов вещественной матрицы по не убыванию: А) сортировкой выбором; В) сортировкой вставками.

**2-21.** Отсортировать элементы диагоналей вещественной матрицы, параллельных главной диагонали, по не возрастанию:

А) сортировкой выбором; В) сортировкой вставками.

**2-22.** Отсортировать элементы нечетных строк вещественной матрицы по не убыванию, а элементы четных строк по не возрастанию:

А) сортировкой выбором; В) сортировкой вставками.

**2-23.** Отсортировать элементы нечетных строк вещественной матрицы по не убыванию, а элементы четных строк по не возрастанию:

А) сортировкой вставками; В) сортировкой обменом.

**2-24.** Отсортировать элементы нечетных столбцов вещественной матрицы по не убыванию, а элементы четных столбцов по не возрастанию:

А) сортировкой вставками; В) сортировкой обменом.

**2-25.** Отсортировать элементы нечетных столбцов вещественной матрицы по не убыванию, а элементы четных столбцов по не возрастанию:

А) сортировкой вставками; В) сортировкой обменом.

### **Тема 3. Сложная сортировка**

*(Пирамидальная, Шелла, Хоора)*

*Программа должна запрашивать из файла данные для сортировки. Результат сортировки по желанию пользователя может сохраняться в файле. Необходимо разработать модуль, включающий в себя:*

- *реализацию указанных алгоритмов сортировки;*
- *профайлер, позволяющий подсчитать количество пересылок и сравнений, произведенных во время сортировки.*

*При необходимости разработать модуль, генерирующий массивы вещественных чисел нужного размера.*

**3-1.** Отсортировать в лексикографическом порядке последовательность слов одинаковой длины: А) пирамидальной сортировкой; В) сортировкой Шелла.

**3-2.** Отсортировать в лексикографическом порядке последовательность слов разной длины: А) пирамидальной сортировкой; В) сортировкой Шелла.

**3-3.** Отсортировать в лексикографическом порядке последовательность слов одинаковой длины: А) пирамидальной сортировкой; В) сортировкой Хоора.

**3-4.** Отсортировать в лексикографическом порядке последовательность слов разной длины: А) сортировкой Хоора; В) сортировкой Шелла.

**3-5.** Отсортировать в порядке не возрастания элементы столбцов вещественной матрицы размера  $(n \times m)$  А) пирамидальной сортировкой; В) сортировкой Шелла.

**3-6.** Отсортировать в порядке не возрастания элементы строк вещественной матрицы размера  $(n \times m)$  А) сортировкой Хоора; В) сортировкой Шелла.

**3-7.** В исходном файле содержатся слова с разным числом вхождения какой-то буквы. Отсортировать заданную последовательность слов сначала в лексикографическом порядке, а затем расположить их в порядке убывания количества вхождений выбранной буквы:

А) сортировкой Хоора; В) сортировкой Шелла. Сделать вывод об устойчивости алгоритмов.

**3-8.** В исходном файле содержатся слова с разным числом вхождения какой-то буквы. Отсортировать заданную последовательность слов сначала в лексикографическом порядке, а затем расположить их в порядке убывания количества вхождений выбранной буквы:

А) сортировкой Хоора; В) пирамидальной сортировкой. Сделать вывод об устойчивости алгоритмов.

**3-9.** В исходном файле содержатся фамилии и имена 10 студентов. Отсортировать их в лексикографическом порядке сначала по именам, потом по фамилиям: А) сортировкой Хоора; В) пирамидальной сортировкой. Сделать вывод об устойчивости алгоритмов.

**3-10.** В исходном файле содержатся фамилии и имена 10 студентов. Отсортировать их в лексикографическом порядке сначала по именам, потом по фамилиям: А) сортировкой Хоора; В) сортировкой Шелла. Сделать вывод об устойчивости алгоритмов.

**3-11.** В исходном файле содержатся фамилии и имена 10 студентов. Отсортировать их в лексикографическом порядке сначала по именам, потом по фамилиям: А) пирамидальной сортировкой; В) сортировкой Шелла. Сделать вывод об устойчивости алгоритмов.

**3-12.** Отсортировать массив вещественных чисел ( $size=200$ ) по не возрастанию с помощью алгоритма Шелла при  $d=1, 2, 4, \dots$  и при  $d=1, 4, 13, \dots$ . Сделать вывод об эффективности на основе многократных наблюдений (сравнивать средние значения количества пересылок и сравнений).

**3-13.** Отсортировать массив вещественных чисел ( $size=200$ ) по не убыванию с помощью алгоритма Хоора при

$median = arr\left[\frac{left + right}{2}\right]$  и



при  $median = arr\left[\sqrt{(left + 1) \cdot right}\right]$ . Сделать вывод об эффективности на основе многократных наблюдений (сравнивать средние значения количества пересылок и сравнений).

**3-14.** Отсортировать массив вещественных чисел ( $size=2000$ ) не убыванию с помощью алгоритма Шелла при  $d=1, 2, 4, \dots$  и при  $d=1, 4, 13, \dots$ . Сделать вывод об эффективности на основе многократных наблюдений (сравнивать средние значения количества пересылок и сравнений).

**3-15.** Отсортировать массив вещественных чисел ( $size=100$ ) по не возрастанию с помощью алгоритма Хоора при

$$median = arr\left[\frac{left + right}{2}\right] \text{ и при } median = arr\left[\sqrt{(left + 1) \cdot right}\right].$$

Сделать вывод об эффективности на основе многократных наблюдений над различными массивами одинаковой размерности (сравнивать средние значения количества пересылок и сравнений).

**3-16.** С помощью сортировки Шелла найти  $k$  наименьших элементов в массиве вещественных чисел из  $N$  элементов.

**3-17.** С помощью сортировки Хоора найти  $k$  наименьших элементов в массиве вещественных чисел из  $N$  элементов.

**3-18.** С помощью пирамидальной сортировки найти  $k$  наименьших элементов в массиве вещественных чисел из  $N$  элементов.

**3-19.** С помощью сортировки Шелла найти  $k$  наибольших элементов, являющихся целыми числами, в массиве вещественных чисел из  $N$  элементов.

**3-20.** С помощью сортировки Хоора найти  $k$  наибольших элементов, являющихся целыми числами, в массиве вещественных чисел из  $N$  элементов.

**3-21.** С помощью пирамидальной сортировки найти  $k$  наибольших элементов, являющихся целыми числами, в массиве вещественных чисел из  $N$  элементов.

**3-22.** Дан массив вещественных чисел ( $size=2000$ ). С помощью алгоритма Шелла упорядочить по не возрастанию элементы массива, начиная с заданного элемента, с определенным шагом.

**3-23.** Дан массив вещественных чисел ( $size=2000$ ). С помощью алгоритма Хоора упорядочить по не возрастанию элементы массива, начиная с заданного элемента, с определенным шагом.

**3-24.** Дан массив вещественных чисел ( $size=2000$ ). С помощью пирамидальной сортировки упорядочить по не возрастанию элементы массива, начиная с заданного элемента, с определенным шагом.

**3-25.** Дан массив вещественных чисел ( $size=2000$ ). С помощью алгоритма Хоора упорядочить по не возрастанию элементы массива, и

определить элемент, встречающийся чаще других. Если таких элементов несколько, вывести все.

## Тема 4. Классы. Наследование

**4-1.** 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.

**4-2.** 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».

**4-3.** 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».

**4-4.** 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.

**4-5.** 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, ку-

пившего 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.

**4-6.** 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».

**4-7.** 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;

**4-8.** 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.

**4-9.** 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.

**4-10.** 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.

**4-11.** 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.

**4-12.** 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».

**4-13.** 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.

**4-14.** 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.

**4-15.** 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.

**4-16.** 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.

**4-17.** 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.

**4-18.** 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.

**4-19.** 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.

**4-20.** 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.

**4-21.** 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 просмат-

ривать все расписание на неделю, на отдельный день и редактировать поле «название курса».

**4-22.** 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.

**4-23.** 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.

**4-24.** 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.

**4-25.** 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.

## Тема 5. Перегрузка операций

*В каждом варианте необходимо, во-первых, реализовать определенную структуру данных, с которой можно работать только через перегруженные операции. Во-вторых, с помощью механизма наследования реализовать другую структуру данных на основе первой также с перегруженными операциями. Причем описание перегружаемых в потомке операций должно происходить через операции, перегруженные в родительском классе.*

**5-1.** Создайте класс, реализующий однонаправленный список на основе одномерного массива. В интерфейс класса (public) должны входить только перегруженные операции:

- 1) переместиться в начало списка (--);
  - 2) передвинуть указатель на один элемент (++);
  - 3) добавить элемент в текущее положение указателя (+);
  - 4) удалить элемент по текущему положению указателя (-).
- все остальные методы и поля должны иметь тип доступа private.

С помощью механизма наследования реализуйте на основе линейного однонаправленного списка очередь. В интерфейс класса «очередь» (public) должны входить только перегруженные операции:

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

Для демонстрации работы программы необходимо реализовать меню, позволяющее вызывать операции, как для списка, так и для очереди. На экране должна отображаться вся последовательность введенных элементов, над которыми производятся действия.

**5-2.** Создайте класс, реализующий однонаправленный список на основе одномерного массива. В интерфейс класса (public) должны входить только перегруженные операции:

- 1) переместиться в начало списка (--);
- 2) передвинуть указатель на один элемент (++);
- 3) добавить элемент в текущее положение указателя (+);
- 4) удалить элемент по текущему положению указателя (-).

С помощью механизма наследования реализуйте на основе линейного однонаправленного списка стек. В интерфейс класса «стек» (public) должны входить только перегруженные операции:



- 1) добавить элемент в стек (<<);
- 2) извлечь элемент из стека (>>).

Для демонстрации работы программы необходимо реализовать меню, позволяющее вызывать операции, как для списка, так и для стека. На экране должна отображаться вся последовательность введенных элементов, над которыми производятся действия.

**5-3.** Создайте класс, реализующий однонаправленный список на основе одномерного массива. В интерфейс класса (public) должны входить только перегруженные операции:

- 1) переместиться в начало списка (--);
- 2) передвинуть указатель на один элемент (++);
- 3) добавить элемент в текущее положение указателя (+);
- 4) удалить элемент по текущему положению указателя (-).

С помощью механизма наследования реализуйте на основе линейного однонаправленного списка двунаправленный список. В интерфейс класса (public) должны входить только 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 (-).

Для демонстрации работы программы необходимо реализовать меню, позволяющее вызывать операции, как для списка, так и для двунаправленного списка. На экране должна отображаться вся последовательность введенных элементов, над которыми производятся действия.

**5-4.** Создайте класс, реализующий динамический однонаправленный список. В интерфейс класса (public) должны входить только перегруженные операции:

- 1) переместиться в начало списка (--);
- 2) передвинуть указатель на один элемент (++);
- 3) добавить элемент в текущее положение указателя (+);
- 4) удалить элемент по текущему положению указателя (-).

С помощью механизма наследования реализуйте на основе линейного однонаправленного списка очередь. В интерфейс класса (public) должны входить только перегруженные операции:

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

Для демонстрации работы программы необходимо реализовать меню, позволяющее вызывать операции, как для списка, так и для очереди. На экране должна отображаться вся последовательность введенных элементов, над которыми производятся действия.

**5-5.** Создайте класс, реализующий динамический однонаправленный список. В интерфейс класса (public) должны входить только перегруженные операции:

- 1) переместиться в начало списка (--);
- 2) передвинуть указатель на один элемент (++);
- 3) добавить элемент в текущее положение указателя (+);
- 4) удалить элемент по текущему положению указателя (-).

С помощью механизма наследования реализуйте на основе линейного однонаправленного списка стек. В интерфейс класса (public) должны входить только перегруженные операции:

- 1) добавить элемент в стек (<<);
- 2) извлечь элемент из стека (>>).

Для демонстрации работы программы необходимо реализовать меню, позволяющее вызывать операции, как для списка, так и для стека. На экране должна отображаться вся последовательность введенных элементов, над которыми производятся действия.

**5-6.** Создайте класс, реализующий динамический однонаправленный список. В интерфейс класса (public) должны входить только перегруженные операции:

- 1) переместиться в начало списка (--);
- 2) передвинуть указатель на один элемент (++);
- 3) добавить элемент в текущее положение указателя (+);
- 4) удалить элемент по текущему положению указателя (-).

С помощью механизма наследования реализуйте на основе линейного однонаправленного списка двунаправленный список. В интерфейс класса (public) должны входить только 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 (+).

Для демонстрации работы программы необходимо реализовать меню, позволяющее вызывать операции, как для однонаправленного списка, так и для двунаправленного списка. На экране должна отображаться вся последовательность введенных элементов, над которыми производятся действия.

**5-7.** Создайте класс, реализующий динамический однонаправленный список. В интерфейс класса (public) должны входить только перегруженные операции:

- 1) переместиться в начало списка (--);
- 2) передвинуть указатель на один элемент (++);
- 3) добавить элемент в текущее положение указателя (+);
- 4) удалить элемент по текущему положению указателя (-).

С помощью механизма наследования реализуйте на основе линейного однонаправленного списка одномерный массив. В интерфейс класса (public) должна входить только перегруженная операция индексирования.

Для демонстрации работы программы необходимо реализовать меню, позволяющее вызывать операции, как для списка, так и для массива. На экране должна отображаться вся последовательность введенных элементов, над которыми производятся действия.

**5-8.** Создайте класс, реализующий динамический однонаправленный список. В интерфейс класса (public) должны входить только перегруженные операции:

- 1) переместиться в начало списка (--);
- 2) передвинуть указатель на один элемент (++);
- 3) добавить элемент в текущее положение указателя (+);
- 4) удалить элемент по текущему положению указателя (-).

С помощью механизма наследования реализуйте на основе линейного однонаправленного списка двумерный массив. В интерфейс класса (public) должна входить только перегруженная операция индексирования.

Для демонстрации работы программы необходимо реализовать меню, позволяющее вызывать операции, как для списка, так и для массива. На экране должна отображаться вся последовательность введенных элементов, над которыми производятся действия.

**5-9.** Создайте класс, позволяющий реализовать векторную алгебру в двумерном пространстве. В интерфейс класса (public) должны входить только перегруженные операции:

- 1) сложение векторов (+);
- 2) вычитание векторов (-);
- 3) скалярное произведение векторов (\*);
- 4) присвоение векторов (=), причем можно присваивать значение одного вектора другому или строке целых чисел;
- 5) нахождение длины вектора (len(x)).

С помощью механизма наследования реализуйте класс комплексных чисел. В интерфейс класса (public) должны входить только перегруженные операции:

- 1) сложение комплексных чисел (+);
- 2) вычитание комплексных чисел (-);
- 3) умножение комплексных чисел (\*);
- 4) деление комплексных чисел (/);
- 5) нахождение модуля комплексного числа (mod(z)).

Для демонстрации работы программы необходимо реализовать меню, позволяющее вызывать операции, как для векторов, так и для

комплексных чисел. На экране должны отображаться все введенные элементы, над которыми производятся действия.

**5-10.** Создайте класс, позволяющий реализовать векторную алгебру в трехмерном пространстве. В интерфейс класса (public) должны входить только перегруженные операции:

- 1) сложение векторов (+);
- 2) вычитание векторов (-);
- 3) скалярное произведение векторов (\*);
- 4) присвоение векторов (=), причем можно присваивать значение одного вектора другому или строке целых чисел;
- 5) нахождение длины вектора ( $\text{len}(x)$ ).

С помощью механизма наследования реализуйте класс, позволяющий работать с квадратными матрицами размером  $3 \times 3$ . В интерфейс класса (public) должны входить только перегруженные операции:

- 1) сложение матриц (+);
- 2) вычитание матриц (-);
- 3) умножение матриц (\*);
- 4) нахождение определителя матрицы ( $\text{det}(z)$ ).

Для демонстрации работы программы необходимо реализовать меню, позволяющее вызывать операции, как для векторов, так и для матриц. На экране должны отображаться все введенные элементы, над которыми производятся действия.

**5-11.** Создайте класс, позволяющий реализовать работу с комплексными числами. В интерфейс класса (public) должны входить только перегруженные операции:

- 1) сложение комплексных чисел (+);
- 2) вычитание комплексных чисел (-);
- 3) умножение комплексных чисел (\*);
- 4) деление комплексных чисел (/);
- 5) нахождение модуля комплексного числа ( $\text{mod}(z)$ ).

С помощью механизма наследования реализуйте класс двумерных векторов. В интерфейс класса (public) должны входить только перегруженные операции:

- 1) сложение векторов (+);
- 2) вычитание векторов (-);
- 3) скалярное произведение векторов (\*);
- 4) присвоение векторов (=), причем можно присваивать значение одного вектора другому или строке целых чисел;
- 5) нахождение длины вектора ( $\text{len}(x)$ ).

Для демонстрации работы программы необходимо реализовать меню, позволяющее вызывать операции, как для комплексных чисел, так и для векторов. На экране должны отображаться все введенные элементы, над которыми производятся действия.

**5-12.** Создайте класс, реализующий однонаправленную очередь на основе одномерного массива. В интерфейс класса (public) должны входить только перегруженные операции:

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

С помощью механизма наследования реализуйте на основе однонаправленной очереди стек. В интерфейс класса (public) должны входить только перегруженные операции:

- 1) добавить элемент в стек (<<);
- 2) извлечь элемент из стека (>>).

Для демонстрации работы программы необходимо реализовать меню, позволяющее вызывать операции, как для очереди, так и для стека. На экране должна отображаться вся последовательность введенных элементов, над которыми производятся действия.

**5-13.** Создайте класс, реализующий однонаправленную очередь на основе одномерного массива. В интерфейс класса (public) должны входить только перегруженные операции:

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

С помощью механизма наследования реализуйте на основе однонаправленной очереди двунаправленную очередь. В интерфейс класса (public) должны входить только 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 (<<).

Для демонстрации работы программы необходимо реализовать меню, позволяющее вызывать операции, как для однонаправленной очереди, так и для двунаправленной очереди. На экране должна отображаться вся последовательность введенных элементов, над которыми производятся действия.

**5-14.** Создайте класс, реализующий однонаправленную очередь на основе одномерного массива. В интерфейс класса (public) должны входить только перегруженные операции:

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

С помощью механизма наследования реализуйте на основе однонаправленной очереди однонаправленный список. В интерфейс класса (public) должны входить только перегруженные операции:

- 1) переместиться в начало списка (--);
- 2) передвинуть указатель на один элемент (++);
- 3) добавить элемент в текущее положение указателя (+);
- 4) удалить элемент по текущему положению указателя (-).

Для демонстрации работы программы необходимо реализовать меню, позволяющее вызывать операции, как для очереди, так и для списка. На экране должна отображаться вся последовательность введенных элементов, над которыми производятся действия.

**5-15.** Создайте класс, реализующий однонаправленную очередь на основе одномерного массива. В интерфейс класса (public) должны входить только перегруженные операции:

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

С помощью механизма наследования реализуйте на основе однонаправленной очереди двунаправленный список. В интерфейс класса (public) должны входить только 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 (-).

Для демонстрации работы программы необходимо реализовать меню, позволяющее вызывать операции, как для очереди, так и для списка. На экране должна отображаться вся последовательность введенных элементов, над которыми производятся действия.

**5-16.** Создайте класс, реализующий двунаправленную очередь на основе одномерного массива. В интерфейс класса (public) должны входить только 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 (<<).

С помощью механизма наследования реализуйте на основе двунаправленной очереди стек. В интерфейс класса (public) должны входить только перегруженные операции:

- 1) добавить элемент в стек (<<);
- 2) извлечь элемент в стек (>>).

Для демонстрации работы программы необходимо реализовать меню, позволяющее вызывать операции, как для очереди, так и для стека. На экране должна отображаться вся последовательность введенных элементов, над которыми производятся действия.

**5-17.** Создайте класс, реализующий двунаправленную очередь на основе одномерного массива. В интерфейс класса (public) должны входить только 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 (<<).

С помощью механизма наследования реализуйте на основе двунаправленной очереди однонаправленный список. В интерфейс класса (public) должны входить только перегруженные операции:

- 1) переместиться в начало списка (--);
- 2) передвинуть указатель на один элемент (++);
- 3) добавить элемент в текущее положение указателя (+);
- 4) удалить элемент по текущему положению указателя (-).

Для демонстрации работы программы необходимо реализовать меню, позволяющее вызывать операции, как для списка, так и для очереди. На экране должна отображаться вся последовательность введенных элементов, над которыми производятся действия.

**5-18.** Создайте класс, реализующий двунаправленную очередь на основе одномерного массива. В интерфейс класса (public) должны входить только 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 (<<).

С помощью механизма наследования реализуйте на основе двунаправленной очереди двунаправленный список. В интерфейс класса (public) должны входить только 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 (-).

Для демонстрации работы программы необходимо реализовать меню, позволяющее вызывать операции, как для списка, так и для очереди. На экране должна отображаться вся последовательность введенных элементов, над которыми производятся действия.

**5-19.** Создайте класс, реализующий динамическую однонаправленную очередь. В интерфейс класса (public) должны входить только перегруженные операции:

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

С помощью механизма наследования реализуйте на основе однонаправленной очереди стек. В интерфейс класса (public) должны входить только перегруженные операции:

- 1) добавить элемент в стек (<<);
- 2) извлечь элемент из стека (>>).

Для демонстрации работы программы необходимо реализовать меню, позволяющее вызывать операции, как для очереди, так и для стека. На экране должна отображаться вся последовательность введенных элементов, над которыми производятся действия.

**5-20.** Создайте класс, реализующий динамическую однонаправленную очередь. В интерфейс класса (public) должны входить только перегруженные операции:

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

С помощью механизма наследования реализуйте на основе однонаправленной очереди двунаправленную очередь. В интерфейс класса (public) должны входить только 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 (<<).

Для демонстрации работы программы необходимо реализовать меню, позволяющее вызывать операции, как для однонаправленной очереди, так и для двунаправленной очереди. На экране должна отображаться вся последовательность введенных элементов, над которыми производятся действия.

**5-21.** Создайте класс, реализующий динамическую однонаправленную очередь. В интерфейс класса (public) должны входить только перегруженные операции:

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

С помощью механизма наследования реализуйте на основе однонаправленной очереди однонаправленный список. В интерфейс класса (public) должны входить только перегруженные операции:

- 1) переместиться в начало списка (--);
- 2) передвинуть указатель на один элемент (++);
- 3) добавить элемент в текущее положение указателя (+);
- 4) удалить элемент по текущему положению указателя (-).

Для демонстрации работы программы необходимо реализовать меню, позволяющее вызывать операции, как для списка, так и для очереди. На экране должна отображаться вся последовательность введенных элементов, над которыми производятся действия.

**5-22.** Создайте класс, реализующий динамическую однонаправленную очередь. В интерфейс класса (public) должны входить только перегруженные операции:

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



С помощью механизма наследования реализуйте на основе однонаправленной очереди двунаправленный список. В интерфейс класса (public) должны входить только 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 (-).

Для демонстрации работы программы необходимо реализовать меню, позволяющее вызывать операции, как для списка, так и для очереди. На экране должна отображаться вся последовательность введенных элементов, над которыми производятся действия.

**5-23.** Создайте класс, реализующий динамическую двунаправленную очередь. В интерфейс класса (public) должны входить только 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 (<<).

С помощью механизма наследования реализуйте на основе двунаправленной очереди стек. В интерфейс класса (public) должны входить только перегруженные операции:

- 1) добавить элемент в стек (<<);
- 2) извлечь элемент в стек (>>).

Для демонстрации работы программы необходимо реализовать меню, позволяющее вызывать операции, как для стека, так и для очереди. На экране должна отображаться вся последовательность введенных элементов, над которыми производятся действия.

**5-24.** Создайте класс, реализующий динамическую двунаправленную очередь. В интерфейс класса (public) должны входить только 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 (<<).

С помощью механизма наследования реализуйте на основе двунаправленной очереди однонаправленный список. В интерфейс класса (public) должны входить только перегруженные операции:

- 1) переместиться в начало списка (--);
- 2) передвинуть указатель на один элемент (++);
- 3) добавить элемент по текущему положению указателя (+);
- 4) удалить элемент по текущему положению указателя (-).

Для демонстрации работы программы необходимо реализовать меню, позволяющее вызывать операции, как для списка, так и для очереди. На экране должна отображаться вся последовательность введенных элементов, над которыми производятся действия.

**5-25.** Создайте класс, реализующий динамический стек. В интерфейс класса (public) должны входить только перегруженные операции:

- 1) добавить элемент в стек (<<);
- 2) извлечь элемент в стек (>>).

С помощью механизма наследования реализуйте на основе стека двунаправленную очередь. Доступными извне должны быть только 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 (<<).

Для демонстрации работы программы необходимо реализовать меню, позволяющее вызывать операции, как для стека, так и для очереди. На экране должна отображаться вся последовательность введенных элементов, над которыми производятся действия.

## Тема 6. Шаблоны классов

*Во всех вариантах необходимо первоначально 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, создающий динамический массив заданного типа, с нулевым количеством занятых элементов;*
- *конструктор копирования;*
- *очистку массива;*
- *метод «обработки массива»;*
- *деструктор.*

*Далее на основе данного шаблона создать класс для работы со строкой символов, специализировав метод «обработки массива» для вашей конкретной задачи.*

**6-1.** Дан текст из строчных русских букв, за которым следует точка. Напечатать этот текст заглавными буквами.

**6-2.** Дан непустой текст из заглавных русских букв, за которым следует точка. Определить - упорядочены ли эти буквы по алфавиту. В случае неупорядоченности указать позицию первого символа, нарушающего алфавитный порядок.

**6-3.** Напечатать в алфавитном порядке все различные строчные русские буквы, входящие в заданный текст.

**6-4.** Дана строка латинских символов. Напечатать эту строку, предварительно заменив все вхождения 'abc' на 'def'.

**6-5.** Дана строка латинских символов. Напечатать эту строку, предварительно удалив первое вхождение 'w', если такое есть (образовавшуюся «дыру» заполнить последующими буквами, а в конец добавить пробел).

**6-6.** Дана строка латинских символов. Напечатать эту строку, предварительно заменив на 'ks' первое вхождение 'x', если оно есть.

**6-7.** Дана последовательность, содержащая от 1 до 30 слов, в каждом из которых от 1 до 5 строчных латинских букв; между соседними слова-

ми – запятая, за последним словом точка. Напечатать эту же последовательность слов, но в обратном порядке.

**6-8.** Дана последовательность, содержащая от 1 до 30 слов, в каждом из которых от 1 до 5 строчных латинских букв; между соседними словами – запятая, за последним словом точка. Напечатать эту же последовательность слов, удалив из нее повторно входящие слова.

**6-9.** Дана последовательность, содержащая от 1 до 30 слов, в каждом из которых от 1 до 5 строчных латинских букв; между соседними словами – запятая, за последним словом точка. Напечатать все слова, которые встречаются в последовательности по одному разу.

**6-10.** Дана последовательность, содержащая от 1 до 30 слов, в каждом из которых от 1 до 5 строчных латинских букв; между соседними словами – запятая, за последним словом точка. Напечатать все различные слова, указав для каждого из них число его вхождений в последовательность.

**6-11.** Дана последовательность, содержащая от 1 до 30 слов, в каждом из которых от 1 до 5 строчных латинских букв; между соседними словами – запятая, за последним словом точка. Напечатать все слова в алфавитном порядке, разделяя их пробелами без запятых.

**6-12.** Дана последовательность, содержащая от 1 до 30 слов, в каждом из которых от 1 до 5 строчных латинских букв; между соседними словами – не менее одного пробела, за последним словом точка. Напечатать слова, которые отличны от последнего слова и являются симметричными.

**6-13.** Дана последовательность, содержащая от 1 до 30 слов, в каждом из которых от 1 до 5 строчных латинских букв; между соседними словами – не менее одного пробела, за последним словом точка. Напечатать слова, первая буква которых входит в них еще ровно один раз, при этом выбранные слова должны отличаться от последнего слова.

**6-14.** Дана последовательность, содержащая от 1 до 30 слов, в каждом из которых от 1 до 5 строчных латинских букв; между соседними словами – не менее одного пробела, за последним словом точка. Выберите слова, отличающиеся от последнего слова. Напечатать те из них, у которых буквы упорядочены по алфавиту.

**6-15.** Дана последовательность, содержащая от 1 до 30 слов, в каждом из которых от 1 до 5 строчных латинских букв; между соседними словами – не менее одного пробела, за последним словом точка. Выберите слова, отличающиеся от последнего слова. Напечатать те из них, в которых нет повторяющихся букв.

**6-16.** Дана последовательность, содержащая от 1 до 30 слов, в каждом из которых от 1 до 5 строчных латинских букв; между соседними словами – не менее одного пробела, за последним словом точка. Выберите

слова, отличающиеся от последнего слова. В выбранных словах перенесите первую букву в конец слова. Напечатайте полученные слова.

**6-17.** Дана последовательность, содержащая от 1 до 30 слов, в каждом из которых от 1 до 5 строчных латинских букв; между соседними словами – не менее одного пробела, за последним словом точка. Выберите слова, отличающиеся от последнего слова. В выбранных словах перенесите последнюю букву в начало слова. Напечатайте полученные слова.

**6-18.** Дана последовательность, содержащая от 1 до 30 слов, в каждом из которых от 1 до 5 строчных латинских букв; между соседними словами – не менее одного пробела, за последним словом точка. Выберите слова, отличающиеся от последнего слова. В выбранных словах удалите первую букву. Напечатайте полученные слова.

**6-19.** Дана последовательность, содержащая от 1 до 30 слов, в каждом из которых от 1 до 5 строчных латинских букв; между соседними словами – не менее одного пробела, за последним словом точка. Выберите слова, отличающиеся от последнего слова. В выбранных словах удалите последнюю букву. Напечатайте полученные слова.

**6-20.** Дана последовательность, содержащая от 1 до 30 слов, в каждом из которых от 1 до 5 строчных латинских букв; между соседними словами – не менее одного пробела, за последним словом точка. Выберите слова, отличающиеся от последнего слова. В выбранных словах удалите все буквы, совпадающие с первой, кроме первой буквы. Напечатайте полученные слова.

**6-21.** Дана последовательность, содержащая от 1 до 30 слов, в каждом из которых от 1 до 5 строчных латинских букв; между соседними словами – не менее одного пробела, за последним словом точка. Выберите слова, отличающиеся от последнего слова. В выбранных словах удалите все буквы, совпадающие с последней, кроме самой последней буквы. Напечатайте полученные слова.

**6-22.** Дана последовательность, содержащая от 1 до 30 слов, в каждом из которых от 1 до 5 строчных латинских букв; между соседними словами – не менее одного пробела, за последним словом точка. Выберите слова, отличающиеся от последнего слова. В выбранных словах удалите все повторные вхождения букв. Напечатайте полученные слова.

**6-23.** Дана последовательность, содержащая от 1 до 30 слов, в каждом из которых от 1 до 5 строчных латинских букв; между соседними словами – не менее одного пробела, за последним словом точка. Выберите слова, отличающиеся от последнего слова. В выбранных словах удалите среднюю букву, если слово нечетной длины. Напечатайте полученные слова.

**6-24.** Дан текст из заглавных латинских букв, за которым следует пробел. Определить, является ли этот текст правильной записью римски-

ми цифрами целого числа от 1 до 999, и, если является, распечатать это число арабскими цифрами.

**6-25.** Напечатать таблицу умножения в шестнадцатеричной системе счисления.

## Тема 7. Потокосные классы

**7-1.** Реализуйте модуль, подключение которого перегружает операции помещения в поток вывода и извлечение из потока ввода для класса матриц размером  $3 \times 3$ . Данный класс также описывается в указанном модуле и содержит перегруженные операции сложения и умножения матриц.

**7-2.** 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.

**7-3.** 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.

**7-4.** 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.

**7-5.** 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.

**7-6.** 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.

**7-7.** 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.

**7-8.** 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.

**7-9.** 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.

**7-10.** 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.

**7-11.** 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.

**7-12.** Реализуйте модуль, подключение которого перегружает операции помещения в поток вывода и извлечение из потока ввода для класса, содержащего методы разложения функции  $\ln(1+ax)$  в ряд Тейлора. На экран должно выводиться  $n$  первых слагаемых разложения. Значения  $a$  и  $n$  определяет пользователь.

**7-13.** Реализуйте модуль, подключение которого перегружает операции помещения в поток вывода и извлечение из потока ввода для класса, содержащего методы разложения функции  $\sin(x)+\cos(x)$  в ряд Тейлора. На экран должно выводиться  $n$  первых слагаемых разложения. Значение  $n$  определяет пользователь.

**7-14.** Реализуйте модуль, подключение которого перегружает операции помещения в поток вывода и извлечение из потока ввода для класса, содержащего методы разложения функции  $\sin(ax)+\cos(x)$  в ряд Тейлора. На экран должно выводиться  $n$  первых слагаемых разложения. Значения  $a$  и  $n$  определяет пользователь.

**7-15.** Реализуйте модуль, подключение которого перегружает операции помещения в поток вывода и извлечение из потока ввода для класса, содержащего методы разложения функции  $\sin(x)+\cos(ax)$  в ряд Тейлора. На экран должно выводиться  $n$  первых слагаемых разложения. Значения  $a$  и  $n$  определяет пользователь.

**7-16.** Реализуйте модуль, подключение которого перегружает операции помещения в поток вывода и извлечение из потока ввода для класса,

содержащего методы разложения функции  $\exp(x) + \exp(-x)$  в ряд Тейлора. На экран должно выводиться  $n$  первых слагаемых разложения. Значение  $n$  определяет пользователь.

**7-17.** Реализуйте модуль, подключение которого перегружает операции помещения в поток вывода и извлечение из потока ввода для класса, содержащего методы разложения функции  $\exp(x) + \exp(ax)$  в ряд Тейлора. На экран должно выводиться  $n$  первых слагаемых разложения. Значения  $a$  и  $n$  определяет пользователь.

**7-18.** Реализуйте модуль, подключение которого перегружает операции помещения в поток вывода и извлечение из потока ввода для класса, содержащего методы разложения функции  $\exp(x) - \exp(-x)$  в ряд Тейлора. На экран должно выводиться  $n$  первых слагаемых разложения. Значение  $n$  определяет пользователь.

**7-19.** Реализуйте модуль, подключение которого перегружает операции помещения в поток вывода и извлечение из потока ввода для класса, содержащего методы разложения функции  $\exp(x) - \exp(ax)$  в ряд Тейлора. На экран должно выводиться  $n$  первых слагаемых разложения. Значения  $a$  и  $n$  определяет пользователь.

**7-20.** Реализуйте модуль, подключение которого перегружает операции помещения в поток вывода и извлечение из потока ввода для класса, содержащего методы разложения функции  $(1+x)^m$  в ряд Тейлора. На экран должно выводиться  $n$  первых слагаемых разложения. Значение  $n$  и  $m$  определяет пользователь.

**7-21.** Реализуйте модуль, подключение которого перегружает операции помещения в поток вывода и извлечение из потока ввода для класса, содержащего методы разложения функции  $(1+ax)^m$  в ряд Тейлора. На экран должно выводиться  $n$  первых слагаемых разложения. Значения  $a$ ,  $m$  и  $n$  определяет пользователь.

**7-22.** Реализуйте модуль, подключение которого перегружает операции помещения в поток вывода и извлечение из потока ввода для класса, содержащего методы разложения функции  $\exp(x) + \ln(1+x)$  в ряд Тейлора. На экран должно выводиться  $n$  первых слагаемых разложения. Значение  $n$  определяет пользователь.

**7-23.** Реализуйте модуль, подключение которого перегружает операции помещения в поток вывода и извлечение из потока ввода для класса, содержащего методы разложения функции  $\exp(x) - \ln(1+x)$  в ряд Тейлора. На экран должно выводиться  $n$  первых слагаемых разложения. Значение  $n$  определяет пользователь.

**7-24.** Реализуйте модуль, подключение которого перегружает операции помещения в поток вывода и извлечение из потока ввода для класса, содержащего методы разложения функции двух переменных  $(a+x)^m$  в



ряд Тейлора. На экран должно выводиться  $n$  первых слагаемых разложения. Значения  $a$ ,  $m$  и  $n$  определяет пользователь.

**7-25.** Реализуйте модуль, подключение которого перегружает операции помещения в поток вывода и извлечение из потока ввода для класса, содержащего методы разложения функции двух переменных  $(b+ax)^m$  в ряд Тейлора. На экран должно выводиться  $n$  первых слагаемых разложения. Значения  $a$ ,  $b$ ,  $m$  и  $n$  определяет пользователь.

## Литература

1. Т.А. Павловская С/С++. Программирование на языке высокого уровня. СПб.:Питер, 2002.
2. Е.Л.Романов Практикум по программированию на С++. СПб.:БХВ-Петербург, 2004.
3. В.Г. Давыдов Прграммирование и основы алгоритмизации М.:Высшая школа, 2003.
4. А.В. Ахо, Дж.Э. Хопкрофт, Д.Д. Ульман Структуры данных и алгоритмы М.: Издательский дом «Вильямс», 2003.

---

## Содержание

1. Деревья	3
2. Простые методы сортировки	5
3. Сложная сортировка	7
4. Классы. Наследование	10
5. Перегрузка операций	16
6. Шаблоны классов	27
7. Потокосые классы	30
8. Литература	34

Светлана Юрьевна Белим

ЗАДАНИЯ ПО КУРСУ  
«ПРОГРАММИРОВАНИЕ НА ЯЗЫКЕ ВЫСОКОГО УРОВНЯ»

Методические указания  
по курсу «Программирование на языке высокого уровня».  
Специальность – «Вычислительные машины, комплексы,  
системы и сети»

Авторское редактирование

Подписано в печать 24.01.2006. Формат 60x84 1/16.  
Печ.л. 1.5 Уч.-изд.л 2.1 Тираж 70 экз.

Полиграфический центр КАН  
644050, г. Омск, пр. Мира 32, ком.11, тел. (381-2) 65-47-31  
Лицензия ПЛД № 58-47 от 21.04.97 г.