**Задание №6**

**“Методы сортировки и динамические структуры данных”**

**Цель работы.**

1. Изучение возможностей языка С++ для создания классов и объектов.
2. Практика в использовании возможностей языка С++ для действий над динамическими структурами данных и передачи их как параметров в методы класса.
3. Практика в использовании файлового ввода/вывода.
4. Практика в использовании графического интерфейса.

**Задание.**

Решить задачу по обработке данных в соответствии с вариантом.

Решение каждого элемента задания оформить в виде отдельного метода класса.

**Требования к программе.**

1. Каждый метод реализовать в виде вычислительной подпрограммы (без вывода на экран). Вывод на экран должен осуществляться в основной программе.
2. Все структуры данных реализовать в виде классов.
3. Исходные структуры данных вводятся из файла на диске в основной программе и передаются как параметры в подпрограмму для обработки. Имена файлов с исходными структурами данных вводятся.

Cоздать **два проекта**:

а) с интерактивным вводом имен файлов через графический интерфейс;

б) с передачей имен файлов как параметров в командной строке для исполняемой программы при запуске ее в **командной строке ОС**.

1. Результаты расчетов передаются через список параметров соответствующего метода.
2. Выход из программы - по желанию пользователя.

**Вопросы**

1. Смысл понятий «объект, метод, свойство, экземпляр» в ООП.
2. Принцип действия метода сортировки вставками.
3. Описать и изобразить отличие линейного списка от бинарного дерева.
4. Объяснить термин «живучесть» для программы. Является ли ваша программа живучей? Подтвердите ответ тестами.
5. Тестирование по принципам «белого ящика» и «черного ящика». Проведите его для своей программы.
6. Указать в отчете ссылки на источники, использованные для выполнения работы в теоретической и практической части.

**Варианты задания**

**Вариант 1**

Составить программу, которая содержит динамическую информацию о наличии

автобусов в автобусном парке.

Сведения о каждом автобусе содержат:

• номер автобуса;

• фамилию и инициалы водителя;

• номер маршрута.

Программа должна обеспечивать:

* начальное формирование данных о всех автобусах в парке в виде списка;

• при выезде каждого автобуса из парка вводится номер автобуса, и программа

удаляет данные об этом автобусе из списка автобусов, находящихся в парке,

и записывает эти данные в список автобусов, находящихся на маршруте;

• при въезде каждого автобуса в парк вводится номер автобуса, и программа

удаляет данные об этом автобусе из списка автобусов, находящихся на маршруте,

и записывает эти данные в список автобусов, находящихся в парке;

• по запросу выдаются сведения об автобусах, находящихся в парке, или об автобусах,

находящихся на маршруте.

**Вариант 2**

Составить программу, которая содержит текущую информацию о книгах в библиотеке.

Сведения о книгах содержат:

• номер УДК;

• фамилию и инициалы автора;

• название;

• год издания;

• количество экземпляров данной книги в библиотеке.

Программа должна обеспечивать:

• начальное формирование данных о всех книгах в библиотеке в виде двоичного

дерева;

• добавление данных о книгах, вновь поступающих в библиотеку;

• удаление данных о списываемых книгах;

• по запросу выдаются сведения о наличии книг в библиотеке, упорядоченные

по годам издания.

**Вариант 3**

Составить программу, которая содержит текущую информацию о заявках на

авиабилеты.

Каждая заявка содержит:

• пункт назначения;

• номер рейса;

• фамилию и инициалы пассажира;

• желаемую дату вылета.

Программа должна обеспечивать:

• хранение всех заявок в виде списка;

• добавление заявок в список;

• удаление заявок;

• вывод заявок по заданному номеру рейса и дате вылета;

• вывод всех заявок.

**Вариант 4**

Составить программу, которая содержит текущую информацию о заявках на

авиабилеты.

Каждая заявка содержат:

• пункт назначения;

• номер рейса;

• фамилию и инициалы пассажира;

• желаемую дату вылета.

Программа должна обеспечивать:

• хранение всех заявок в виде двоичного дерева;

• добавление и удаление заявок;

• по заданному номеру рейса и дате вылета вывод заявок с их последующим

удалением;

• вывод всех заявок.

**Вариант 5**

Составить программу, которая содержит текущую информацию о книгах в библиотеке.

Сведения о книгах содержат:

• номер УДК;

• фамилию и инициалы автора;

• название;

• год издания;

а количество экземпляров данной книги в библиотеке.

Программа должна обеспечивать:

• начальное формирование данных о всех книгах в библиотеке в виде списка;

• при взятии каждой книги вводится номер УДК, и программа уменьшает значение

количества книг на единицу или выдает сообщение о том, что требуемой

книги в библиотеке нет, или требуемая книга находится на руках;

• при возвращении каждой книги вводится номер УДК, и программа увеличивает

значение количества книг на единицу;

• по запросу выдаются сведения о наличии книг в библиотеке.

**Вариант 6**

Составить программу, которая содержит динамическую информацию о наличии   
автобусов в автобусном парке.

Сведения о каждом автобусе содержат:

• номер автобуса;

• фамилию и инициалы водителя;

• номер маршрута;

• признак того, где находится автобус — на маршруте или в парке.

Программа должна обеспечивать:

• начальное формирование данных о всех автобусах в виде списка;

• при выезде каждого автобуса из парка вводится номер автобуса, и программа

устанавливает значение признака «автобус на маршруте»;

• при въезде каждого автобуса в парк вводится номер автобуса, и программа

устанавливает значение признака «автобус в парке»;

• по запросу выдаются сведения об автобусах, находящихся в парке, или об автобусах,

находящихся на маршруте.

**Вариант7**

Составить программу, отыскивающую проход по лабиринту.

Лабиринт представляется в виде матрицы, состоящей из квадратов. Каждый

квадрат либо открыт, либо закрыт. Вход в закрытый квадрат запрещен. Если

квадрат открыт, то вход в него возможен со стороны, но не с угла. Каждый квадрат

определяется его координатами в матрице.

Программа находит проход через лабиринт, двигаясь от заданного входа. После

отыскания прохода программа выводит найденный путь в виде координат квадратов.

Для хранения пути использовать стек.

**Вариант 8**

Гаражная стоянка имеет одну стояночную полосу, причем въезд и выезд находятся

в одном конце полосы. Если владелец автомашины приходит забрать свой автомобиль,

который не является ближайшим к выходу, то все автомашины, загораживающие

проезд, удаляются, машина данного владельца выводится со стоянки,

а другие машины возвращаются на стоянку в исходном порядке.

Написать программу, которая моделирует процесс прибытия и отъезда машин.

Прибытие или отъезд автомашины задается командной строкой, которая содержит

признак прибытия или отъезда и номер машины. Программа должна выводить

сообщение при прибытии или выезде любой машины. При выезде автомашины

со стоянки сообщение должно содержать число раз, которое машина

удалялась со стоянки для обеспечения выезда других автомобилей.

**Вариант 9**

Составить программу, моделирующую заполнение гибкого магнитного диска.

Общий объем памяти на диске 360 Кбайт. Файлы имеют произвольную длину от

18 байт до 32 Кбайт. В процессе работы файлы либо записываются на диск, либо

удаляются с него.

В начале работы файлы записываются подряд друг за другом. После удаления

файла на диске образуется свободный участок памяти, и вновь записываемый

файл либо размещается на свободном участке, либо, если файл не вмещается в

свободный участок, размещается после последнего записанного файла.

В случае, когда файл превосходит длину самого большого свободного участка,

выдается аварийное сообщение. Требование на запись или удаление файла задается

в командной строке, которая содержит имя файла, его длину в байтах, признак

записи или удаления. Программа должна выдавать по запросу сведения о

занятых и свободных участках памяти на диске.

Указание: следует создать список занятых участков и список свободных участков

памяти на диске.

**Вариант 10**

в файловой системе каталог файлов организован как линейный список.

Для каждого файла в каталоге содержатся следующие сведения:

• имя файла;

• дата создания;

• количество обращений к файлу.

Составить программу, которая обеспечивает:

• начальное формирование каталога файлов;

• вывод каталога файлов;

• удаление файлов, дата создания которых меньше заданной;

• выборку файла с наибольшим количеством обращений.

Программа должна обеспечивать диалог с помощью меню и контроль ошибок

при вводе.

**Вариант 11**

Предметный указатель организован как линейный список.

Каждая компонента указателя содержит слово и номера страниц, на которых это

слово встречается. Количество номеров страниц, относящихся к одному слову, от

одного до десяти.

Составить программу, которая обеспечивает:

• начальное формирование предметного указателя;

• вывод предметного указателя;

• вывод номеров страниц для заданного слова.

Программа должна обеспечивать диалог с помощью меню и контроль ошибок

при вводе.

**Вариант 12**

Текст помощи для некоторой программы организован как линейный список.

Каждая компонента текста помощи содержит термин (слово) и текст, содержащий

пояснения к этому термину. Количество строк текста, относящихся к одному

термину, от одной до пяти.

Составить программу, которая обеспечивает:

• начальное формирование текста помощи;

• вывод текста помощи;

• вывод поясняющего текста для заданного термина.

Программа должна обеспечивать диалог с помощью меню и контроль ошибок

при вводе.

**Вариант 13**

Картотека в бюро обмена квартир организована как линейный список.

Сведения о каждой квартире содержат:

• количество комнат;

• этаж;

• площадь;

• адрес.

Составить программу, которая обеспечивает:

• начальное формирование картотеки;

• ввод заявки на обмен;

• поиск в картотеке подходящего варианта: при равенстве количества комнат

и этажа и различии площадей в пределах 10% выводится соответствующая

карточка и удаляется из списка, в противном случае поступившая заявка

включается в список;

• вывод всего списка.

Программа должна обеспечивать диалог с помощью меню и контроль ошибок

при вводе.

**Вариант 14**

Англо-русский словарь построен как двоичное дерево.

Каждая компонента содержит английское слово, соответствующее ему русское

слово и счетчик количества обращений к данной компоненте.

Первоначально дерево формируется согласно английскому алфавиту. В процессе

эксплуатации словаря при каждом обращении к компоненте в счетчик обращений

добавляется единица.

Составить программу, которая:

• обеспечивает начальный ввод словаря с конкретными значениями счетчиков

обращений;

• формирует новое представление словаря в виде двоичного дерева по следующему

алгоритму: а) в старом словаре ищется компонента с наибольшим значением

счетчика обращений; б) найденная компонента заносится в новый

словарь и удаляется из старого; в) переход к п. а) до исчерпания исходного

словаря;

• производит вывод исходного и нового словарей.

Программа должна обеспечивать диалог с помощью меню и контроль ошибок

при вводе.

**Вариант 15**

Анкета для опроса населения содержит две группы вопросов.

Первая группа содержит сведения о респонденте:

• возраст;

• пол;

• образование (начальное, среднее, высшее).

Вторая группа содержит собственно вопрос анкеты, ответ на который либо ДА,

либо НЕТ.

Составить программу, которая:

• обеспечивает начальный ввод анкет и формирует из них линейный список;

• на основе анализа анкет выдает ответы на следующие вопросы: а) сколько

мужчин старше 40 лет, имеющих высшее образование, ответили ДА на вопрос

анкеты; а) сколько женщин моложе 30 лет, имеющих среднее образование, ответили

НЕТ на вопрос анкеты; а) сколько мужчин моложе 25 лет, имеющих

начальное образование, ответили ДА на вопрос анкеты;

• производит вывод всех анкет и ответов на вопросы.

Программа должна обеспечивать диалог с помощью меню и контроль ошибок

при вводе.

**Вариант 16**

Составить программу, которая содержит текущую информацию о книгах в библиотеке.

Сведения о книгах содержат:

• номер УДК;

• фамилию и инициалы автора;

• название;

• год издания;

• количество экземпляров данной книги в библиотеке.

Программа должна обеспечивать:

• начальное формирование данных о всех книгах в библиотеке в виде списка;

• добавление данных о книгах, вновь поступающих в библиотеку;

• удаление данных о списываемых книгах;

• по запросу выдаются сведения о наличии книг в библиотеке, упорядоченные

по годам издания.

**Вариант 17**

На междугородной телефонной станции картотека абонентов, содержащая сведения

о телефонах и их владельцах, организована как линейный список.

Составить программу, которая:

• обеспечивает начальное формирование картотеки в виде линейного списка;

• производит вывод всей картотеки;

• вводит номер телефона и время разговора;

• выводит извещение на оплату телефонного разговора.

Программа должна обеспечивать диалог с помощью меню и контроль ошибок

при вводе.

**Вариант 18**

На междугородной телефонной станции картотека абонентов, содержащая сведения

о телефонах и их владельцах, организована как двоичное дерево.

Составить программу, которая:

• обеспечивает начальное формирование картотеки в виде двоичного дерева;

• производит вывод всей картотеки;

• вводит номер телефона и время разговора;

• выводит извещение на оплату телефонного разговора.

Программа должна обеспечивать диалог с помощью меню и контроль ошибок

при вводе.

**Вариант 19**

Автоматизированная информационная система на железнодорожном вокзале содержит

сведения об отправлении поездов дальнего следования. Для каждого поезда

указывается:

• номер поезда;

• станция назначения;

• время отправления.

Данные в информационной системе организованы в виде линейного списка.

Составить программу, которая:

• обеспечивает первоначальный ввод данных в информационную систему и

формирование линейного списка;

• производит вывод всего списка;

• вводит номер поезда и выводит все данные об этом поезде;

• вводит название станции назначения и выводит данные обо всех поездах, следующих

до этой станции.

Программа должна обеспечивать диалог с помощью меню и контроль ошибок

при вводе.

**Вариант 20**

Автоматизированная информационная система на железнодорожном вокзале содержит

сведения об отправлении поездов дальнего следования.

Для каждого поезда указывается:

• номер поезда;

• станция назначения;

а время отправления.

Данные в информационной системе организованы в виде двоичного дерева.

Составить программу, которая:

• обеспечивает первоначальный ввод данных в информационную систему и формирование

двоичного дерева;

• производит вывод всего дерева;

• вводит номер поезда и выводит все данные об этом поезде;

• вводит название станции назначения и выводит данные о всех поездах, следующих

до этой станции.

Программа должна обеспечивать диалог с помощью меню и контроль ошибок при вводе.