# Объектно-ориентированное программирование в системах управления. Задания по лабораторным работам

Язык реализации: C++.

# Лабораторная работа №1. Динамические массивы

*Консольное приложение. Запросить у пользователя размер массива и его элементы. На печать вывести исходные и преобразованные массивы. По окончании работы с массивом освободить занимаемую им память.*

*Продемонстрировать отладочный режим работы среды Visual Studio на примере выполнения цикла вывода массива с наблюдением состояния счётчика цикла и элементов массива.*

*В шапку программы поместить как комментарий задание по варианту.*

1. Ввести массив из *n* целых чисел. Поменять в массиве наибольшее и наименьшее число местами.
2. Дана матрица А. Получить квадрат этой матрицы А2.
3. Ввести 5 цифр, считая, что эти числа являются диагональными элементами матрицы. Вывести на экран общий вид матрицы обозначив остальные элементы символом "Х".
4. В массиве из *n* вводимых чисел поставить максимальное в середине последовательности.
5. Сформировать матрицу размером *m*×*n*, состоящую из букв, упорядоченных в строках по алфавиту (произвольная матрица вводится с клавиатуры).
6. В массиве из *n* вводимых числах найти минимальное и поставить его предварительно сдвинув остальные элементы на одну позицию последним в этой последовательности.
7. Упорядочить по возрастанию массив, состоящий из *n* произвольных букв латинского алфавита.
8. Заданы два массива А(*m*) и В(*n*). Первым на печать вывести массив, сумма значений которого окажется наименьшей.
9. Заданы два массива А(*m*) и В(*n*). Первым на печать вывести массив, произведение значений которого окажется наименьшим.
10. Заданы два массива А(*m*) и В(*n*). В каждом из массивов найти наименьшее значение и прибавить его ко всем элементам массивов. На печать вывести исходные и преобразованные массивы.
11. Написать программу, организующую массив из *n* компонент, каждая из которых содержит информацию о типе компьютера, тактовой частоте и стоимости.
12. Заданы два массива А(*m*) и В(*n*). В каждом из массивов найти наибольшее значение и вычесть его из всех элементов массивов. На печать вывести исходные и преобразованные массивы.
13. Заданы два массива А(*m*) и В(*n*). В каждом из массивов найти среднее арифметическое всех элементов массивов. На печать вывести исходные массивы и найденные значения.
14. Заданы два массива А(*m*) и В(*n*). Первым на печать вывести массив, содержащий наименьшее значение. Напечатать также это значение и его порядковый номер.
15. Заданы два массива А(*m*) и В(*n*). Первым на печать вывести массив, содержащий наибольшее значение. Напечатать также это значение и его порядковый номер.
16. Заданы два массива А(*m*) и В(*n*). В каждом из массивов найти наименьшее значение и умножить на него все элементы массивов. На печать вывести исходные и преобразованные массивы.
17. Заданы два массива А(*m*) и В(*n*). Подсчитать в них количество отрицательных элементов и первым на печать вывести массив, имеющий наименьшее их количество.
18. Упорядочить массив *n* целых чисел в порядке убывания.
19. Организовать циклическую перестановку элементов *n*-мерного массива в группах из трех элементов.
20. Заданы два массива А(*m*) и В(*n*). Подсчитать в них количество положительных элементов и первым на печать вывести массив, имеющий наименьшее их количество.
21. Заданы два массива А(*m*) и В(*n*). Подсчитать в них количество отрицательных элементов и первым на печать вывести массив, имеющий наибольшее их количество.
22. Определить среднее значение из *n* элементов массива и записать его в первую позицию.
23. Записать массив из *n* чисел в обратном порядке (новый массив не вводить).
24. Заданы два массива А(*m*) и В(*n*). Подсчитать в них количество элементов, больших значения *t* и первым на печать вывести массив, имеющий наименьшее их количество.
25. Заданы два массива А(*m*) и В(*n*). Подсчитать в них количество элементов, меньших значения *t* и первым на печать вывести массив, имеющий наименьшее их количество.
26. Заданы два массива А(*m*) и В(*n*). Подсчитать в них количество элементов, больших значения *t* и первым на печать вывести массив, имеющий наибольшее их количество.
27. Заданы два массива А(*m*) и В(*n*). Подсчитать в них количество положительных элементов и первым на печать вывести массив, имеющий наибольшее их количество.
28. Сформировать таблицу *sin*(*x*) на интервале [-π, π] и записать её в обратном порядке.
29. В массиве, сформированном по закону *y*(*i*)=*sin*(*x*(*i*)), где *a*<=*x*<=*b* (границы а и в а также шаг изменения аргумента вводится с клавиатуры), поменять знаки компонент. На печать вывести исходный и преобразованные массивы.
30. Заданы два массива А(*m*) и В(*n*). В каждом из массивов найти наибольшее значение и умножить на него все элементы массивов. На печать вывести исходные и преобразованные массивы.
31. Отсортировать *n*-мерный массив типа *long* в порядке возрастания. На печать вывести исходные и преобразованные массивы.
32. Заданы два массива А(*m*) и В(*n*). В каждом из массивов найти наименьшее значение и разделить на него все элементы массивов. На печать вывести исходные и преобразованные массивы.

# Лабораторная работа №2. Классы. Объекты. Конструктор. Деструктор. Функция-друг

*Консольное приложение. Запросить у пользователя количество объектов и его поля (данные хранятся в составе класса). Организовать проект, в котором объявление класса находится в заголовочном файле \*.h, а реализация (тело метода) – в исполняемом файле, с функцией main(). Создание объектов – с динамическим выделением памяти операцией new. По окончании работы с массивом объектов освободить занимаемую им память.*

*Организовать простейшее меню (1 просмотр текущего состояния объектов, 2 создание объектов, 3 выполнение метода индивидуального задания, 4 выход).*

*В шапку программы поместить как комментарий задание по варианту.*

1. Создать программу с классом Product, порождающий объекты – данные о товарах, имеющихся на складе. Класс включает в себя следующие поля:

- наименование товара;

- количество единиц товара;

- цену единицы товара;

- дату поступления товара на склад;

и следующие методы:

- конструктор с параметрами;

- деструктор;

- метод просмотра текущего состояния объектов;

- функция-друг подсчёта общей стоимости указанного товара.

1. Создать программу с классом Student, порождающий объекты – данные о студентах желающих получить места в общежитии. Класс включает в себя следующие элементы:

- Ф.И.О. студента;

- номер группы (буква и четыре цифры);

- средний балл;

- доход на одного члена семьи;

и следующие функции-члены класса:

- конструктор по умолчанию;

- деструктор;

- метод просмотра текущего состояния объектов;

- функция-друг, выводящая всех студентов с уровнем дохода на одного члена семьи ниже заданного.

Общежитие в ПЕРВУЮ очередь предоставляется тем студентам у кого доход на члена семьи меньше двух минимальных зарплат

Функция друг должна ВЫВЕСТИ ФИО студентов которым будут вне очереди предоставлены места в общежитии

Функция друг должна ВЫВЕСТИ ФИО студентов со средним баллом выше указанного

1. Создать программу с классом Bus порождающий объекты – данные об автобусных рейсах. Класс включает в себя следующие поля:

- номер рейса;

- тип автобуса (маршрутка, городской автобус, международный автобус);

- пункт назначения;

- время отправления;

- время прибытия на конечный пункт;

и следующие методы:

- конструктор с параметрами;

- деструктор;

- метод просмотра текущего состояния объектов;

- функция-друг, выводящая информацию о рейсах, которыми можно воспользоваться для прибытия в пункт назначения раньше заданного времени.

1. Создать программу с классом Phone, порождающий объекты – данные о телефонных разговорах на междугородней АТС. Класс включает в себя следующие поля:

- дату разговора;

- код и название города;

- время разговора;

- тариф;

- исходящий телефонный номер в этом городе;

- входящий телефонный номер абонента;

и следующие методы:

- конструктор по умолчанию

- деструктор;

- метод просмотра текущего состояния объектов;

- функция-друг, выводящая по заданному городу общее время разговора с ним.

1. Создать программу с классом Firm, порождающий объекты – данные о конкретном сотруднике фирмы. Класс включает в себя следующие поля:

- Ф И О сотрудника;

- табельный номер;

- количество проработанных часов за месяц;

- почасовой тариф;

и следующие методы:

- конструктор с параметрами;

- деструктор;

- метод просмотра текущего состояния объектов.

- функция-друг, выводящая данные о сотрудниках с отработанным временем свыше 144 часов. Оно считается сверхурочным и оплачивается в двойном размере.

1. Создать программу с классом Sport, порождающий объекты – данные об участниках спортивных соревнований. Класс включает в себя следующие поля:

- Ф И О игрока;

- вид спорта;

- пол;

- возраст;

- разряд (кандидат в мастера спорта (КМС), I разряд, II разряд, III разряд, юношеские разряды);

- рост;

и следующие методы:

- конструктор по умолчанию

- деструктор;

- метод просмотра текущего состояния объектов;

- функция-друг, выводящая данные о количестве спортсменов заданного пола по категориям: детские, юношеские, взрослые.

1. Создать программу с классом Book, порождающий объекты – данные о книгах, хранящихся в библиотеке. Класс включает в себя следующие поля:

- регистрационный номер книги;

- автор;

- название;

- год издания;

- издательство;

- количество страниц,

и следующие методы:

- конструктор с параметрами;

- деструктор;

- метод инициализации текущего состояния объектов;

- метод просмотра текущего состояния объектов;

- функция-друг, выводящая данные о книгах из указанного диапазона года издания.

функция друг которая выводит список книг с фамилиями авторов, изданных после заданного года

функция друг которая выводит количество книг указанного издательства

1. Создать программу с классом Detail порождающий объекты – данные о партиях деталей выпускаемых различными цехами завода. Класс включает в себя следующие элементы:

- наименование;

- количество;

- номер цеха;

и следующие методы:

- конструктор по умолчанию;

- деструктор;

- метод просмотра текущего состояния объектов;

- функция-друг, выводящая данные о всех деталях для заданного цеха.

1. Создать программу с классом Sotr, порождающий объекты – c информацией о сотрудниках предприятия. Класс включает в себя следующие поля:

- Ф И О;

- пол;

- номер отдела;

- должность;

- дату начала работы;

и следующие методы:

- конструктор с параметрами;

- деструктор;

- метод просмотра текущего состояния объектов.

- функция-друг, выводящая список сотрудников, чей возраст через два года станет пенсионным (он разный для мужчин и для женщин).

1. Создать программу с классом Abitur, порождающий объекты – данные об абитуриентах сдавших вступительные экзамены в БГУИР. Класс включает в себя следующие поля:

- Ф И О;

- адрес;

- оценки по трём предметам;

и следующие методы:

- конструктор по умолчанию;

- деструктор;

- метод просмотра текущего состояния объектов;

- функция-друг, подсчитывающая средний балл для всех абитуриентов по каждому предмету.

1. Создать программу с классом Tovar порождающий объекты – данные о товарах, имеющихся на складе. Класс включает в себя следующие поля:

- наименование товара;

- количество единиц товара;

- цену единицы товара;

- дату поступления товара на склад.

и следующие функции-члены класса:

- конструктор с параметрами;

- деструктор;

- метод инициализации текущего состояния объектов;

- метод просмотра текущего состояния объектов;

- функция-друг, выводящая данные о товарах в заданном диапазоне цен.

метод подсчета общей стоимости указанного товара

функция-друг, которая позволяет ВЫВЕСТИ ТОВАРЫ СТОИМОСТЬ КОТОРЫХ ПРЕВЫШАЕТ 1000 рублей.

1. Создать программу с классом Chip, порождающим объекты – данные о микросхемах памяти, хранящиеся на складе. Класс включает в себя следующие поля:

- тип (оперативная – RAM, постоянная – ROM);

- информационная ёмкость (бит);

- наименование;

- месяц/год даты производства;

и следующие функции-члены класса:

- конструктор по умолчанию;

- деструктор;

- метод инициализации текущего состояния объектов;

- функция-друг, выводящая список микросхем, указанной информационной ёмкости.

1. Создать программу с классом RealEstate, порождающим объекты – данные об объектах недвижимости. Класс включает в себя следующие поля:

- название объекта (по названию определяется жилое/нежилое, т.е. жилой многоэтажный дом, коттедж, дача, детский сад, школа, кинотеатр, заводской корпус, библиотека, магазин,…);

- адрес;

- Ф.И.О. владельца;

- общая площадь;

- жилая площадь (для жилых, – с указанием количества комнат);

и следующие методы:

- конструктор с параметрами;

- деструктор;

- метод просмотра текущего состояния объектов;

- функция-друг, вычисляющий и выводящий нежилую площадь и количество комнат.

1. Создать программу с классом Magazine, порождающим объекты – данные о журналах, хранящихся в библиотеке. Класс включает в себя следующие поля:

- регистрационный номер журнала;

- название;

- год издания;

- номер;

- количество страниц;

и следующие методы:

- конструктор с параметром для инициализации названия;

- деструктор;

- метод просмотра текущего состояния объектов;

- функция-друг, выводящая все журналы за указанный год и подсчитывающая общее количество их страниц.

1. Создать программу с классом Technics, порождающим объекты – данные о машинах подъёмно-транспортного оборудования машиностроительного предприятия. Класс включает в себя следующие поля:

- название машины;

- производимые работы;

- состояние (в ремонте/в работе);

- фирма-застройщик;

и следующие методы:

- коэффициент загрузки;

- конструктор по умолчанию;

- деструктор;

- функция-друг, просмотра техники с заданным коэффициентом загрузки.

1. Создать программу с классом Abitur, порождающий объекты – данные об абитуриентах сдавших вступительные экзамены в БГУИР. Класс включает в себя следующие поля:

- Ф И О;

- адрес;

- оценки по трём предметам;

и следующие методы:

- параметризованный конструктор для инициализации объектов;

- деструктор;

- метод просмотра текущего состояния объектов;

- функция-друг, выводящая список студентов, проживающих в заданном городе и подсчитывающая их количество.

1. Создать программу с классом Bus порождающий объекты – данные об автобусных рейсах. Класс включает в себя следующие поля:

- номер рейса;

- тип автобуса (маршрутка, городской автобус, международный автобус);

- пункт назначения;

- время отправления;

- время прибытия на конечный пункт;

и следующие методы:

- конструктор с ограниченным числом параметров (номер рейса);

- деструктор;

- метод просмотра текущего состояния объектов;

- функция-друг, выводящая информацию о рейсах, позволяющих добраться до указанного пункта.

1. Создать программу с классом Detail порождающий объекты – данные о партиях деталей выпускаемых различными цехами завода. Класс включает в себя следующие элементы:

- наименование;

- количество;

- номер цеха;

и следующие методы:

- конструктор по умолчанию;

- деструктор;

- метод просмотра текущего состояния объектов;

- функция-друг, позволяющая для заданного цеха вывести количество выпущенных изделий указанного наименования.

- функция друг позволяющая вывести номера цехов для выпущенных изделий указанного наименования.

1. Создать программу с классом Sport, порождающий объекты – данные об участниках спортивных соревнований. Класс включает в себя следующие поля:

- Ф И О игрока;

- вид спорта;

- пол;

- возраст;

- разряд (кандидат в мастера спорта (КМС), I разряд, II разряд, III разряд, юношеские разряды);

- рост;

и следующие методы:

- конструктор по умолчанию

- деструктор;

- метод просмотра текущего состояния объектов;

- функция-друг, выводящая данные о самом молодом спортсмене в команде;

- функция-друг, выводящая данные о самом лёгком спортсмене в команде.

1. Создать программу с классом Phone, порождающий объекты – данные о телефонных разговорах на междугородней АТС. Класс включает в себя следующие поля:

- дату разговора;

- код и название города;

- время разговора;

- тариф;

- исходящий телефонный номер в этом городе;

- входящий телефонный номер абонента;

и следующие методы:

- конструктор с параметрами;

- деструктор;

- метод просмотра текущего состояния объектов;

- функция-друг, выводящая по номеру телефона название города.

1. Создать программу с классом Sotr, порождающий объекты – данные о сотрудниках предприятия. Класс включает в себя следующие поля:

- Ф И О сотрудника;

- номер отдела;

- должность;

- дату начала работы;

и следующие методы:

- конструктор с параметрами;

- деструктор;

- метод просмотра текущего состояния объектов.

- функция-друг, выводящая список сотрудников по отделами.

1. Создать программу с классом Clothes, порождающим объекты – данные о товарах магазина одежды. Класс включает в себя следующие поля:

- название;

- фирма-производитель;

- размер;

- дата изготовления;

- цена;

и следующие методы:

- конструктор по умолчанию;

- деструктор;

- метод инициализации текущего состояния объектов;

- функция-друг, выводящая список товаров указанной фирмы-производителя.

1. Создать программу с классом Room, порождающим объекты – данные об аудиториях учебного заведения. Класс включает в себя следующие поля:

- номер аудитории в формате №\_корпуса-№\_аудитории;

- учебные дисциплины;

- тип (лекционная, лекционная-поточная, лаборатория, для практических занятий, спортзал, читальный зал);

- размер (количество мест);

- дисциплина;

и следующие методы:

- параметризованный конструктор;

- деструктор;

- метод инициализации текущего состояния объектов;

- функция-друг, выводящая список аудиторий с их данными, в которых возможно проведение занятий по заданной дисциплине.

1. Создать программу с классом Museum, порождающим объекты – данные об экспонатах музея. Класс включает в себя следующие поля и методы:

- название экспоната;

- автор;

- дата создания;

- эпоха;

и следующие методы:

- конструктор по умолчанию;

- деструктор;

- метод инициализации текущего состояния объектов;

- функция-друг, выводящая список объектов экспозиции указанной эпохи;

- функция-друг, выводящая список объектов экспозиции указанного диапазона даты создания.

1. Создать программу с классом Air порождающий объекты – данные об авиарейсах. Класс включает в себя следующие поля:

- номер рейса;

- пункт назначения;

- время вылета;

- время прибытия на конечный пункт;

- время обратного вылета;

- количество и тип пассажиров (взрослые старше 12 лет/дети от 2 до 12 лет/младенцы до 2 лет без места);

- класс (эконом/комфорт/бизнес/первый класс);

и следующие методы:

- конструктор с параметрами;

- деструктор;

- метод просмотра текущего состояния объектов;

- функция-друг, выводящая информацию о рейсах, которыми можно воспользоваться для прибытия в пункт назначения раньше заданного времени для заданного состава пассажиров.

# Лабораторная работа №3. Механизм наследования. Виртуальные функции

*Консольное приложение. Запросить у пользователя количество объектов и его поля. Создать объекты через выделение памяти операцией new. Объявления класса и его методов – в отдельном заголовочном файле \*.h. По окончании работы с массивом освободить занимаемую массивом память с помощью деструктора.*

*Реализация методов – вне класса.*

*Организовать простейшее меню (1 просмотр текущего состояния объектов, 2 создание объектов, 3 выход).*

*В шапку программы поместить в качестве комментария задание по варианту.*

1. Создать программу с классом Student1, порождающим объекты – анкетные данные о студентах.

Класс включает в себя следующие поля и методы:

- ФИО;

- год поступления;

- курс;

- номер группы;

- размер стипендии;

- оценки по *n* предметам (*n* для каждого создаваемого студента запрашивается, а массив оценок создаётся в динамической памяти операцией new);

- массив средних баллов, полученных за *n* семестров.

- конструктор с параметрами;

- деструктор;

- метод просмотра текущего состояния объектов print();

- функция-друг, которая вычисляет средний балл и выводит Ф.И.О. студента, у которого он максимальный;

- виртуальная функция просмотра текущего объекта print().

Производный класс Student2 содержит следующие элементы:

- название общественной работы;

- процент надбавки к стипендии;

- переопределённую функцию вывода данных об общественниках.

2. Создать программу с классом Firm, порождающим объекты – данные о сотрудниках фирмы.

Класс включает в себя следующие поля и методы:

- Ф. И. О. сотрудника;

- табельный номер;

- дату начала работы;

- количество проработанных часов за месяц;

- занимаемая должность;

- стаж работы;

- зарплата;

- почасовой тариф.

- конструктор с параметрами;

- деструктор;

- метод просмотра текущего состояния объектов print();

- функция друг, которая позволяет вывести ФИО сотрудника фирмы и число сверхурочных часов;

- виртуальный метод просмотра текущего объекта print().

Производный класс Firm1 содержит следующие элементы:

- количество командировочных дней в течение месяца;

- процент надбавки к зарплате за каждый день командировки;

- переопределённый метод просмотра состояния объектов – сотрудников, которые в текущем месяце побывали в командировке.

3. Создать программу с классом Deviсe, порождающим объекты – данные об устройствах, входящих в данное оборудование.

Класс включает в себя следующие поля и методы:

- завод поставщик;

- стоимость;

- дата поставки.

- конструктор с параметрами;

- деструктор;

- функция просмотра текущего состояния объектов print();

- функция-друг, позволяющая определить общую стоимость устройств, входящих в оборудование;

- виртуальная функция просмотра текущего объекта print().

Производный класс Module содержит следующие элементы:

- время наработки на отказ;

- потребляемая мощность;

- переопределённую функцию просмотра состояния объектов.

4. Создать программу с классом Student, порождающим объекты – данные о студентах, желающих получить места в общежитии. Общежитие в первую очередь предоставляется тем студентам, у кого доход на члена семьи меньше двух минимальных зарплат.

Класс включает в себя следующие поля:

- Ф.И.О. студента;

- номер группы (буква и четыре цифры);

- средний балл;

- доход на одного члена семьи.

В состав класса входят следующие методы:

- конструктор с параметрами;

- деструктор;

- метод просмотра текущего состояния объектов;

- функция-друг, выводящая Ф.И.О. студентов со средним баллом выше указанного пользователем;

- функция-друг должна вывести ФИО студентов, которым будут вне очереди предоставлены места в общежитии.

- виртуальный метод просмотра текущего объекта print().

Производный класс Stud1 содержит следующие данные на внеочередников:

- вид общественной нагрузки;

- количество членов семьи (для многодетных семей);

- неполная семья (отсутствие одного из родителей);

- переопределённую функцию просмотра состояния объектов – студентов внеочередников.

5. Создать программу с классом Bus, порождающим объекты – данные о конкретных автобусных рейсах.

Класс включает в себя следующие поля и методы:

- номер рейса;

- тип автобуса;

- пункт назначения;

- цена билета;

- время отправления;

- время прибытия на конечный пункт.

- конструктор с ограниченным числом параметров (номер рейса);

- деструктор;

- функция просмотра текущего состояния объектов print();

- функция установки текущего состояния объектов для остальных элементов;

- функция-друг должна вывести информацию о рейсах, позволяющих добраться до указанного пользователем пункта;

- функция друг, выводящая информацию о рейсах, которыми можно воспользоваться для прибытия в пункт назначения раньше заданного времени

- виртуальная функция просмотра текущего объекта print().

Производный класс Express содержит следующие данные об автобусах –экспрессах:

- дни недели работы;

- процентная надбавка на цену билета;

- выйгрыш во времени;

- переопределённую функцию просмотра состояния объектов.

6. Создать программу с классом Sport, порождающим объекты – данные об участниках спортивных соревнований.

Класс включает в себя следующие поля и методы:

- Ф. И. О. игрока;

- игровой номер;

- возраст;

- рост;

- вес.

- конструктор с параметрами;

- деструктор;

- метод просмотра текущего состояния объектов print();

- функция-друг, которая выводит информацию о самом лёгком спортсмене в команде;

- функция-друг, выводящая информацию о самом молодом игроке в команде

- виртуальный метод просмотра текущего объекта print().

Производный класс Dop содержит следующие элементы:

- разряд спортсмена;

- категория (кандидат, мастер и т.д.);

- призёр (Республика, Европа и т.д.);

- переопределённый метод просмотра состояния объектов.

7. Создать программу с классом Phone, порождающим объекты – данные о телефонных разговорах на междугородней АТС

Класс включает в себя следующие поля и методы:

- дату разговора;

- код и название исходящего города;

- код и название входящего города;

- продолжительность разговора;

- тариф;

- номер телефона входящего абонента;

- номер телефона исходящего абонента.

- конструктор с параметрами;

- деструктор;

- метод установки текущего состояния объектов;

- метод, который позволяет вывести по номеру телефона название города;

- функцию-друг, подсчитывающую суммарное время разговора с указанным городом;

- функцию-друг, вычисляющую суммарную стоимость всех звонков на указанный город;

- виртуальный метод просмотра текущего состояния объектов print().

Производный класс Phone1 содержит следующие элементы:

- вид льгот,

- процент для льготного тарифа;

- надбавка к тарифу за срочность вызова;

- переопределённый метод просмотра состояния объектов производного класса print().

8. Создать программу с классом Product, порождающим объекты – данные о товарах, имеющихся на складе.

Класс включает в себя следующие поля и методы:

- страну происхождения;

- фирму-изготовителя;

- наименование товара;

- дату поступления товара на склад.

- текущее количество единиц товара на складе;

- конструктор с параметром для инициализации страны-изготовителя;

- деструктор;

- метод подсчёта общего количества товара указанной фирмы;

- функция-друг, которая позволяет вывести данные о товарах указанной страны происхождения;

- виртуальный метод просмотра текущего объекта print().

Производный класс Order содержит следующие элементы:

- страна заказчик;

- дата заказа;

- количество заказанного товара;

- виртуальный метод просмотра состояния объектов.

9. Создать программу с классом Вооk, порождающим объекты – данные о книгах, хранящихся в библиотеке.

Класс включает в себя следующие поля и методы:

- регистрационный номер книги,

- автор;

- название;

- год издания;

- издательство;

- количество страниц.

- конструктор с параметром для инициализации автора;

- деструктор;

- метод инициализации текущего состояния объектов;

- метод просмотра текущего состояния объектов print();

- функция-друг, выводящая все книги указанного автора и подсчитывающая общее количество их страниц;

- функция-друг, выводящая список книг с фамилиями авторов, изданных после заданного года;

- виртуальный метод просмотра текущего объекта print().

Производный класс Type имеет следующие элементы

- тематика книг;

- тип литературы (отечественная или зарубежная);

- переопределённую функцию просмотра состояния объектов.

10 Создать программу с классом Detail, порождающим объекты – данные о партиях деталей выпускаемых различными цехами завода.

Класс включает в себя следующие поля и методы:

- наименование детали;

- количество деталей в изделии;

- номер цеха, где они изготовлены.

- конструктор с параметрами;

- деструктор;

- метод инициализации текущего состояния объектов;

- метод просмотра текущего состояния объектов print();

- функция-друг, позволяющая подсчитать общее количество изделий указанного наименования;

- функция-друг, выводящая для заданного цеха количество выпущенных изделий указанного наименования;

- виртуальный метод просмотра текущего объекта print().

Производный класс Detail1 имеет следующие элементы:

- дополнительное антикоррозийное покрытие;

- увеличение надёжности в процентах для разных типов деталей;

- переопределённую функцию просмотра текущего состояния деталей.

11. Создать программу с классом Abitur порождающий объекты – данные об абитуриентах сдавших вступительные экзамены в университет.

Класс включает в себя следующие поля и методы:

- Ф И О;

- адрес;

- оценки по *n* сданным предметам (*n* для каждого создаваемого абитуриента запрашивается, а массив оценок создается в динамической памяти операцией new).

- конструктор с параметрами;

- деструктор;

- метод просмотра текущего состояния объектов;

- функция-друг, определяющая количество абитуриентов, проживающих в г.Минске.

Производный класс Abitur1 содержит следующие элементы:

- каким видом спорта занимается абитуриент;

- тип медали (золотая или серебряная);

- призовое место на Республиканской олимпиаде;

- а также переопределенную функцию просмотра состояния объектов-абитуриентов, имеющих льготы при поступлении.

12. Создать программу с классом Product, порождающий объекты – данные о товарах, имеющихся на складе.

Класс включает в себя следующие поля и методы:

- наименование товара;

- категория (продукты/промышленные товары/косметика)

- количество единиц товара;

- цену единицы товара;

- дату поступления товара на склад.

- конструктор по умолчанию;

- деструктор;

- метод установки текущего состояния объектов;

- метод просмотра текущего состояния объектов;

- метод подсчёта общей стоимости указанного товара (тело метода вне класса).

Производный класс Order содержит следующие элементы:

- название магазина-заказчика;

- дата заказа;

- количество заказанного товара;

- виртуальный метод просмотра состояния объектов;

- функция-друг, которая позволяет вывести товары для заданного магазина, стоимость которых превышает 1000 рублей.

13. Создать программу с классом Вооk, порождающим объекты – данные о книгах, хранящихся в библиотеке.

Класс включает в себя следующие поля и методы:

- автор;

- название;

- год издания;

- издательство;

- количество страниц;

- конструктор по умолчанию;

- деструктор;

- метод инициализации текущего состояния объектов;

- метод просмотра текущего состояния объектов print();

- функция-друг, выводящая список книг, указанного издательства;

- функция-друг, выводящая все книги указанного автора и подсчитывающая общее количество их страниц;

- виртуальный метод просмотра текущего объекта print().

Производный класс Type имеет следующие элементы:

- жанр (детская литература, фантастика, справочник, художественная литература, учебник, стихи, сборник статей, …)

- статус книги (находится в хранилище/на руках у читателя);

- виртуальный метод просмотра состояния объектов.

14. Создать программу с классом Firm, порождающим объекты – данные о сотрудниках фирмы.

Класс включает в себя следующие поля и методы:

- Ф. И. О. сотрудника;

- дату начала работы;

- количество проработанных часов за месяц;

- занимаемая должность;

- название отдела;

- зарплата;

- почасовой тариф.

- конструктор с параметрами;

- деструктор;

- метод просмотра текущего состояния объектов print();

- функция-друг, которая позволяет вывести размер заработанной платы каждого сотрудника фирмы за вычетом подоходного налога, который составляет 12% от суммы заработка. Необходимо учесть, что рабочее время свыше 144 часов считается сверхурочным и оплачивается в двойном размере;

- функция друг позволяющая вывести список сотрудников по отделам;

- виртуальный метод просмотра текущего объекта print().

Производный класс Firm1 содержит следующие элементы:

- количество дней, отработанных в ночную смену в течение месяца;

- процент надбавки к зарплате за каждую ночную смену;

- переопределённый метод просмотра состояния объектов – сотрудников, которые в текущем месяце работали в ночную смену.

15. Создать программу с классом Chip, порождающим объекты – данные о микросхемах памяти, хранящиеся на складе.

Класс включает в себя следующие поля и методы:

- тип (оперативная, постоянная);

- информационная ёмкость;

- наименование;

- интерфейс (последовательная/параллельная память);

- год/месяц даты производства;

- фирма/завод-производитель;

- конструктор по умолчанию;

- деструктор;

- метод инициализации текущего состояния объектов;

- функция-друг, выводящая список микросхем, указанной информационной ёмкости;

- виртуальный метод просмотра текущего объекта print().

Производный класс Type имеет следующие элементы:

- вид (для оперативной памяти – статическая/динамическая; для постоянной – с электрическим стиранием/с ультрафиолетовым);

- статус (новая/использованная);

- виртуальный метод просмотра состояния объектов.

16. Создать программу с классом RealEstate, порождающим объекты – данные об объектах недвижимости.

Класс включает в себя следующие поля и методы:

- название объекта;

- адрес;

- Ф.И.О. владельца;

- дата постройки;

- общая площадь;

- материал стен;

- фирма-застройщик;

- конструктор по умолчанию;

- деструктор;

- метод инициализации текущего состояния объектов;

- функция-друг, выводящая список объектов недвижимости указанного района;

- виртуальный метод просмотра текущего объекта print().

Производный класс Flat имеет следующие элементы:

- жилая площадь квартиры;

- количество комнат;

- виртуальный метод просмотра состояния объектов, вычисляющий и выводящий нежилую площадь и количество комнат.

17. Создать программу с классом Technics, порождающим объекты – данные о машинах подъёмно-транспортного оборудования машиностроительного предприятия.

Класс включает в себя следующие поля и методы:

- название машины;

- производимые работы;

- Ф.И.О. владельца;

- дата создания;

- состояние (в ремонте/в работе);

- фирма-застройщик;

- коэффициент загрузки;

- конструктор по умолчанию;

- деструктор;

- метод инициализации текущего состояния объектов;

- функция-друг, выводящая список машин указанной грузоподъёмности;

- виртуальный метод просмотра текущего объекта print().

Производный класс HavyTechnics имеет следующие элементы:

- техника высокой грузоподъемности;

- часы простоя единиц оборудования во внеплановом ремонте;

- виртуальный метод просмотра техники с заданным коэффициентом загрузки print().

18. Создать программу с классом Clothes, порождающим объекты – данные о товарах магазина одежды.

Класс включает в себя следующие поля и методы:

- название;

- фирма-производитель;

- размер;

- дата изготовления;

- цена;

- конструктор по умолчанию;

- деструктор;

- метод инициализации текущего состояния объектов;

- функция-друг, выводящая список товаров указанной фирмы-производителя;

- виртуальный метод просмотра текущего объекта print().

Производный от Clothes класс Man имеет следующие элементы:

- мужские пиджаки и жакеты;

- цена со скидкой;

- виртуальный метод просмотра состояния объектов, вычисляющий и выводящий размер возможной экономии с учётом скидки.

Производный от Clothes класс Woman имеет следующие элементы:

- платья и сарафаны;

- тип (домашнее/офисное/вечернее);

- виртуальный метод просмотра состояния объектов выводящий платья заданного типа больших размеров.

19. Создать программу с классом Magazine, порождающим объекты – данные о журналах, хранящихся в библиотеке.

Класс включает в себя следующие поля и методы:

- регистрационный номер журнала;

- название;

- год издания;

- номер;

- количество страниц.

- конструктор с параметром для инициализации названия;

- деструктор;

- метод инициализации текущего состояния объектов;

- виртуальный метод просмотра текущего состояния объектов print();

- функция-друг, выводящая все журналы за указанный год и подсчитывающая общее количество их страниц;

Производный от Magazine класс Type имеет следующие элементы:

- тематика журнала (технический, научный, здоровье, кулинария, авто, детский, спорт, компьютерный, …);

- тип литературы (отечественная на русском/белорусском, или зарубежная);

- переопределённую функцию просмотра состояния объектов print().

20. Создать класс квадрат, члены класса – длина стороны. Предусмотреть в классе методы вычисления и вывода сведений о фигуре – диагональ, периметр, площадь. Создать производный класс – правильная квадратная призма с высотой H, добавить в класс метод определения объема фигуры, перегрузить методы расчета площади и вывода сведений о фигуре. Написать программу, демонстрирующую работу с этими классами: дано N квадратов и M призм, найти квадрат с максимальной площадью и призму с максимальной диагональю.

19. Создать класс треугольник, члены класса – длины трех сторон. Предусмотреть в классе методы проверки существования треугольника, вычисления и вывода сведений о фигуре – длины сторон, углы, периметр, площадь. Создать производный класс – равносторонний треугольник, перегрузить в классе проверку, является ли треугольник равносторонним, и метод вывода сведений о фигуре. Написать программу, демонстрирующую работу с классом: дано K треугольников и L равносторонних треугольников, найти среднюю площадь для K треугольников и наибольший равносторонний треугольник.

20. Создать класс окружность, член класса – радиус R. Предусмотреть в классе методы вычисления и вывода сведений о фигуре – площади, длины окружности. Создать производный класс – круглый прямой цилиндр с высотой h, добавить в класс метод определения объема фигуры, перегрузить методы расчета площади и вывода сведений о фигуре. Написать программу, демонстрирующую работу с классом: дано N окружностей и M цилиндров, найти окружность максимальной площади и средний объем цилиндров.

21. Создать класс квадрат, члены класса – длина стороны. Предусмотреть в классе методы вычисления и вывода сведений о фигуре – диагональ, периметр, площадь. Создать производный класс – правильная пирамида с апофемой h, добавить в класс метод определения объема фигуры, перегрузить методы расчета площади и вывода сведений о фигуре. Написать программу, демонстрирующую работу с классом: дано N квадратов и M пирамид, найти квадрат с минимальной площадью и количество пирамид с высотой более числа a (a вводить).

22. Создать класс четырехугольник, члены класса – координаты четырех точек. Предусмотреть в классе методы проверки существования четырехугольника вычисления и вывода сведений о фигуре – длины сторон, диагоналей, периметр, площадь. Создать производный класс – параллелограмм, предусмотреть в классе проверку, является ли фигура параллелограммом. Написать программу, демонстрирующую работу с классом: дано N четырехугольников и M параллелограммов, найти среднюю площадь N четырехугольников и параллелограммы наименьшей и наибольшей площади.

23. Создать программу с классом Exhibition, порождающим объекты – данные о выставочных экспонатах продукции предприятий.

Класс включает в себя следующие поля и методы:

- категория продукции;

- название продукта;

- дата производства;

- предприятие;

- выставочное место;

- выставочная площадь;

- конструктор по умолчанию;

- деструктор;

- метод инициализации текущего состояния объектов;

- функция-друг, выводящая список продуктов, указанного предприятия;

- виртуальный метод просмотра текущего объекта print().

Производный класс Type имеет следующие элементы:

- статус продукта (выпускается в настоящее время/выпущен как опытный образец для изучения спроса);

- срок хранения продукта;

- вес продукта;

- виртуальный метод просмотра состояния объектов по указанной категории и указанного веса;

- функция-друг, выводящая сводку по предприятиям.

24. Создать класс треугольник, члены класса – координаты трех точек. Предусмотреть в классе методы проверки существования треугольника, вычисления и вывода сведений о фигуре – длины сторон, углы, периметр, площадь. Создать производный класс – равносторонний треугольник, предусмотреть в классе проверку, является ли треугольник равносторонним. Написать программу, демонстрирующую работу с классом: дано N треугольников и M равносторонних треугольников, вывести номера одинаковых треугольников и равносторонний треугольник с наименьшей медианой.

25. Создать программу с классом Museum, порождающим объекты – данные об экспонатах музея.

Класс включает в себя следующие поля и методы:

- название экспоната;

- автор;

- дата создания;

- эпоха;

- местонахождение (в зале экспозиции, в мастерской на восстановлении, в хранилище, в составе выездной выставки);

- ценность экспоната;

- деструктор;

- метод инициализации текущего состояния объектов;

- функция-друг, выводящая список объектов экспозиции указанной эпохи;

- виртуальный метод просмотра текущего объекта print().

Производный класс Sculpture имеет следующие элементы:

- материал из которого сделан экспонат (дерево, мрамор, бронза, …);

- в каком месте был найден;

- дата обнаружения;

- виртуальный метод просмотра состояния объектов.

26. Создать класс прямоугольник, члены класса – длины сторон a и b. Предусмотреть в классе методы вычисления и вывода сведений о фигуре – длины сторон, диагоналей, периметр, площадь. Создать производный класс – параллелепипед с высотой с, добавить в класс метод определения объема фигуры, перегрузить методы расчета площади и вывода сведений о фигуре. Написать программу, демонстрирующую работу с классом: дано N прямоугольников и M параллелепипедов, найти количество прямоугольников, у которых площадь больше средней площади прямоугольников и количество кубов (все ребра равны).

27. Создать класс окружность, член класса – радиус R. Предусмотреть в классе методы вычисления и вывода сведений о фигуре – площади, длины окружности. Создать производный класс – конус с высотой h, добавить в класс метод определения объема фигуры, перегрузить методы расчета площади и вывода сведений о фигуре. Написать программу, демонстрирующую работу с классом: дано N окружностей и M конусов, найти количество окружностей, у которых площадь меньше средней площади всех окружностей, и наибольший по объему конус.

28. Создать класс четырехугольник, члены класса – координаты четырех точек. Предусмотреть в классе методы вычисления и вывода сведений о фигуре – 14 длины сторон, диагоналей, периметр, площадь. Создать производный класс – равнобочная трапеция, предусмотреть в классе проверку, является ли фигура равнобочной трапецией. Написать программу, демонстрирующую работу с классом: дано N четырехугольников и M трапеций, найти максимальную площадь четырехугольников и количество четырехугольников, имеющих максимальную площадь, и трапецию с наименьшей диагональю.

29. Создать класс равносторонний треугольник, член класса – длина стороны. Предусмотреть в классе методы вычисления и вывода сведений о фигуре – периметр, площадь. Создать производный класс – правильная треугольная призма с высотой H, добавить в класс метод определения объема фигуры, перегрузить методы расчета площади и вывода сведений о фигуре. Написать программу, демонстрирующую работу с классом: дано N треугольников и M призм. Найти количество треугольников, у которых площадь меньше средней площади треугольников, и призму с наибольшим объемом.

30. Создать класс треугольник, члены класса – длины трех сторон. Предусмотреть в классе методы проверки существования треугольника, вычисления и вывода сведений о фигуре – длины сторон, углы, периметр, площадь. Создать производный класс – прямоугольный треугольник, предусмотреть в классе проверку, является ли треугольник прямоугольным. Написать программу, демонстрирующую работу с классом: дано N треугольников и M прямоугольных треугольников, найти треугольник с максимальной площадью и прямоугольный треугольник с наименьшей гипотенузой.

31. Создать класс четырехугольник, члены класса – координаты четырех точек. Предусмотреть в классе методы вычисления и вывода сведений о фигуре – длины сторон, диагоналей, периметр, площадь. Создать производный класс – квадрат, предусмотреть в классе проверку, является ли фигура квадратом. Написать программу, демонстрирующую работу с классом: дано N четырехугольников и M квадратов; найти четырехугольники с минимальной и максимальной площадью и номера одинаковых квадратов.

32. Создать класс треугольник, члены класса – длины трех сторон. Предусмотреть в классе методы проверки существования треугольника, вычисления и вывода сведений о фигуре – длины сторон, углы, периметр, площадь. Создать производный класс – равнобедренный треугольник, предусмотреть в классе проверку, является ли треугольник равнобедренным. Написать программу, демонстрирующую работу с классом: дано N треугольников и M равнобедренных треугольников, найти среднюю площадь для N треугольников и равнобедренный треугольник с наименьшей площадью.

33. Создать класс квадрат, член класса – длина стороны. Предусмотреть в классе методы вычисления и вывода сведений о фигуре – периметр, площадь, диагональ. Создать производный класс – куб, добавить в класс метод определения объема фигуры, перегрузить методы расчета площади и вывода сведений о фигуре. Написать программу, демонстрирующую работу с классом: дано N1 квадратов и N2 кубов. Найти среднюю площадь квадратов и количество кубов с наибольшей площадью.

Методические указания

1 Для определения иерархии классов связать отношением наследования классы заданного варианта. Из перечисленных классов выбрать один, который будет стоять во главе иерархии. Это абстрактный класс. 2 Определить в классах все необходимые конструкторы и деструктор. 3 Поля класса специфицировать как protected. 4 Пример определения статических компонентов: static person\* begin; // указатель на начало списка static void print(); // просмотр списка 5 Статическое поле инициализировать вне определения класса, в глобальной области. 6 Для добавления объекта в список предусмотреть метод класса, т. е. объект сам добавляет себя в список. Например, a.Add() – объект a добавляет себя в список. Включение объекта в список можно выполнять при создании объекта, т. е. поместить операторы включения в конструктор. В случае иерархии классов, включение объекта в список должен выполнять только конструктор базового класса. Вы должны продемонстрировать оба этих способа. 7 Список просматривать путем вызова виртуального метода Show каждого объекта. 8 Статический метод просмотра списка вызывать не через объект, а через класс.

# Лабораторная работа №4. Шаблоны классов

*Консольное приложение. Запросить у пользователя количество объектов и его поля. Создать объекты через выделение памяти операцией new. Объявления класса и его методов – в отдельном заголовочном файле \*.h. По окончании работы с массивом освободить занимаемую массивом память с помощью деструктора. Организовать хранение данных последовательности N с помощью связного списка на базе класса Vector или CArray (библиотека Windows).*

*Реализация методов – вне класса.*

*Организовать простейшее меню (1 просмотр текущего состояния объектов, 2 создание объектов с запросом номера, 3 удаление заданного объекта, 4 выход).*

*В шапку программы поместить в качестве комментария задание по варианту.*

*Варианты заданий использовать из Лабораторной работы №3.*

Дано: число *N* и последовательность *a*1, *a*2, … *aN*

Создать шаблон класса, порождающего динамические одномерные массивы с элементами различных типов (вещественные, целочисленные, символьные и т.д.). Тип данных и результат являются параметрами по отношению к классу. Программа должна иметь методы инициализации, конструктор, деструктор, просмотра значений созданного массива, а также для решения задач формирования нового массива по следующим алгоритмам:

1. *a*1, (*a*1+*a*2), … ,(*a*1+*a*2+…+*aN*)

2. (a1\*a1), (a1\*a2), …, (a1\*aN)

3. |a1|, |a1+a2|, …, |a1+a2+…aN|

4. a1, -a1\*a2, +a1\*a2\*a3, … ,(-1)N\*a1\*a2\*…aN

5. -a1, +a2, -a3, … , (-1)N\*aN

6. (a1+1), (a2+2) , (a3+3), …, (aN+N)

7. a1\*1, a2\*2, a3\*3, , aN\*N

8. a1\*a2, a2\*a3, … , aN-1\*aN

9. a1/1, a2/2, a3/3, …,aN/N

10. (a1+a2), (a2+a3),… ,(aN-1+aN)

11. (a1+a2+a3), (a2+a3+a4), (a3+a4+a5), … (aN-2+aN-1+aN)

12. (N+a1), ( N-1+a2), ,(1+aN)

13. (N\*a1), ( (N-1)\*a2), ,(1\*aN)

14. A1/N, a2/N, … ,aN/1

15. (N+a1), ( N-1+a2), ,(1+aN)

Перечень лабораторных работ по курсу ООП

1. Динамические массивы (операции new, delete)

2. Классы: элементы данные x, y, z, t тип double; Методы:

* Set – установка начальных значений (исходных данных)
* Print – просмотр текущего состояния
* Run – вычисление по формуле

3. Классы: элементы данные x, y, z, t тип double, создаются в динамической области памяти; Методы:

* Конструктор, деструктор
* Set – установка начальных значений (исходных данных)
* Print – просмотр текущего состояния

Run – вычисление по формуле

4.Класс, порождающий динамические массивы.

5. Наследование и механизм виртуальных функций.

6. Класс для создания и работ со строками: элементы – указатель на строку.

Методы:

* Конструктор, деструктор
* Set – установка начальных значений (исходных данных)
* Print – просмотр текущего состояния

Run – решение поставленной задачи.

7. Шаблоны классов

8. Зачетная задача (выдать в апреле)

# Вопросы для зачёта

1. Основы ООП. Понятие инкапсуляции и полиморфизма.
2. Класс в ООП и его основные компоненты.
3. Основы ООП, 4 основных момента (абстракция, ограничение доступа,
4. модульность и иерархическая структура)
5. Уровни доступа.
6. Область видимости объектов, скрытие имен. Операция **"::".**
7. Динамическая память, функции работы с памятью.
8. Динамическая память, операции работы с памятью.
9. Алгоритм создания одномерного динамического массива.
10. Алгоритм создания двухмерного динамического массива.
11. Ссылки в С++. Общее определение. Отличие ссылки от переменной-указателя.
12. Способы передачи аргументов в функции пользователя. Преимущество использования ссылок.
13. Передача аргументов в функцию по умолчанию.
14. Перегрузка функций пользователя. Раннее связывание.
15. Понятие класса, общая структура.
16. Характеристика элементов-данных класса.
17. Характеристика методов класса. Использование операции привязки "**::".**
18. Указатель **this**. Пример явного использования.
19. Функции-друзья класса.
20. Дружба классов.
21. Функции-конструкторы. Явный и косвенный вызов конструктора.
22. Перегрузка конструкторов.
23. Конструктор со списком инициализаторов.
24. Функции-деструкторы.
25. Динамическое создание объектов класса.
26. Методы класса с атрибутом const.
27. Статические методы и данные (атрибут static).
28. Указатели на компоненты класса Операции **.\* и ->\***
29. Наследование. Базовый и производный классы.
30. Инициализация объектов при множественном наследовании.
31. Указатели на производный и базовый классы.
32. Виртуальный базовый класс.
33. Конструктор во множественном наследовании.
34. Виртуальные функции. Позднее связывание.
35. Виртуальные функции и указатели.
36. Понятие абстрактного класса. "Чистая виртуальная функция".
37. Перезагрузка операций.
38. Особенности перезагрузки операций при помощи методов класса и функций-друзей.
39. Перезагрузка операций методами класса.
40. Перезагрузка операций фукциями-друзьями класса.
41. Использование ссылок при перезагрузке унарных операций.
42. Стандартная библиотека. Общая характеристика.
43. Потоковые классы. Понятие потока. Стандартные потоки. Операции извлечения и включения.
44. Иерархия классов для поддержки потоков в С++.
45. Форматирование данных при вводе/выводе потоками.
46. Файловые потоки.
47. Строковая стандартная библиотека.
48. Контейнерные классы.
49. Итераторы.
50. Алгоритмы.

## *Лабораторная работа №2. Классы. Программирование линейных алгоритмов с использованием функций инициализации set() и вывода результатов print()*

Пользовательский класс должен содержать необходимые элементы-данные,

Метод установки их начальных значений: void set(double X, …);

Метод печати: void print(void);

Метод, решающий поставленную задачу: Void Run(void);

Код методов – вне пространства определения класса. Программа иллюстрирует прямой и косвенный способы обращения к методам. Создать метод Run для вычисления выражения:

1. 

При x=14.26, y=-1.22, z=3.5ґ **t=0.564849.**

1. 

При x=-4.5, y=0.75ґ,z=0.845 **u=-55.6848.**

1. 

При x=3.74, y=-0.825, z=0.16, **v=1.0553.**

1. 

При x=0.4 ,y=-0.875, z=-0.475 **w=1.9873.**

1. 

При x=-15.246, y=4.642, z=20.001 **=-182.036.**

1. 

При x=16.55, y=-2.75, z=0.15 **=-40.630.**

1. 

При x=0.1722, y=6.33, z=3.25 **=-205.305.**

1. 

При x=-2.235, y=2.23, z=15.221 **=39.374.**

1. 

При x=1.825, y=18.225, z=-3.298 **=1.2131.**

1. 

При x=6.251, y=0.827, z=25.001 **b=0.7121.**

1. 

При x=3.251, y=0.325, z=0.466 **c=4.25.**

1. .

При x=17.421, y=10.365, z=0.828 **f=0.33056.**

1. .

При x=12.3, y=15.4, z=0.252 **g=82.8257.**

1. .

При x=2.444, y=0.869, z=-0.13 **-0.49871.**

## *Лабораторная работа №3. Классы. Программирование линейных алгоритмов с использованием конструктора, деструктора, функций-друзей инициализации set() и вывода результатов print()*

Пользовательский класс Х должен содержать необходимые элементы-данные, которые создаются в динамической области памяти.

Конструктор для их создания (операция new) и установки их начальных значений: **Х ();**

Деструктор: ~ **Х ();**

Функция-друг печати: friend **void print(void);**

Функция-друг , решающий поставленную задачу: **friend Void Run(void);**

Код методов и функций– вне пространства определения класса

Программа иллюстрирует прямой и косвенный способы обращения к методам

Составить метод Run для вычисления выражения:

1. 

При x=14.26, y=-1.22, z=3.5ґ **t=0.564849.**

1. 

При x=-4.5, y=0.75ґ,z=0.845 **u=-55.6848.**

1. 

При x=3.74, y=-0.825, z=0.16, **v=1.0553.**

1. 

При x=0.4 ,y=-0.875, z=-0.475 **w=1.9873.**

1. 

При x=-15.246, y=4.642, z=20.001 **=-182.036.**

1. 

При x=16.55, y=-2.75, z=0.15 **=-40.630.**

1. 

При x=0.1722, y=6.33, z=3.25 **=-205.305.**

1. 

При x=-2.235, y=2.23, z=15.221 **=39.374.**

1. 

При x=1.825, y=18.225, z=-3.298 **=1.2131.**

1. 

При x=6.251, y=0.827, z=25.001 **b=0.7121.**

1. 

При x=3.251, y=0.325, z=0.466 **c=4.25.**

1. .

При x=17.421, y=10.365, z=0.828 **f=0.33056.**

1. .

При x=12.3, y=15.4, z=0.252 **g=82.8257.**

1. .

При x=2.444, y=0.869, z=-0.13 **-0.49871.**

## *Лабораторная работа №4. Классы. Программирование класса Array с использованием конструктора, деструктора, функций вывода результатов print(), доступа к элементам массива get()*

1. Дан массив А(n,n).Найти число элементов массива a(i,j)>t и просуммировать все эти элементы.

2. Дан одномерный массив А(n).Сформировать массив B(k),состоящий из a(i)>t. На печать вывести исходный массив, сформированный массив и его размерность.

3. Дан массив A(N,N). Вычислить сумму всех неотрицательных элементов, а также их количество.

4. Дан массив A(n,n). Вычислить сумму всех отрицательных его элементов и их количество.

5. Дан массив A(n,n). Сформировать вектор В(к) из a(i,j)<0. На печать вывести исходный массив, полученный вектор и его размерность.

6. Дан массив A(n,n). Написать программу его поворота на 90 градусов относительно его центра. На печать вывести исходный и повернутый массивы.

7. Дан массив A(n,n). Написать программу его поворота на 180 градусов относительно его центра. На печать вывести исходный и повернутый массивы.

8. Дан массив A(n,n). Написать программу его поворота на 270 градусов относительно его центра. На печать вывести исходный и повернутый массивы.

9. Дан массив A(n,n). Найти сумму всех его элементов, расположенных выше главной диагонали.

10. Дан массив A(n,n). Найти сумму всех его элементов, расположенных ниже главной диагонали.

11. Дан массив A(n,n). Найти сумму всех его элементов, расположенных выше диагонали, противоположной главной.

12. Дан массив A(n,n). Найти сумму всех его элементов, расположенных ниже диагонали, противоположной главной.

13. Задана матрица А(n,n). Найти суммы и произведения элементов, стоящих на главной и противоположной (побочной) диагоналях.

14. Задана матрица А(n,n), состоящая из нулей и единиц. Подсчитать количество нулей и единиц в этой матрице.

15. Задана матрица А(n,n). Переставить местами к-ю и i-ю строки, а эатем l-й и j-й столбцы.

16. Задан массив действительных чисел А(n). Необходимо каждый элемент массива разделить на среднее арифметическое этих элементов. На печать вывести исх. и преобразов. массивы.

17. Задан массив А(n). Получить массив В(к), состоящий из элементов массива А, которые делятся на 3. Подсчитать количество элементов массива В.

18. Задана матрица А(n,n). Получить матрицу В=А^2.

19. Вычислить первую норму матрицы А(n,n), определяемую как максимальная из всех сумм ее элементов по столбцам.

20. Вычислить вторую норму матрицы А(n,n), определяемую как максимальная из всех сумм ее элементов по строкам.

## Лабораторная работа №3. Классы. Использование конструктора, деструктора, функций-друзей инициализации set() и вывода результатов print()

Пользовательский класс Х должен содержать необходимые элементы-данные, которые создаются в динамической области памяти.

Конструктор для их создания (операция new) и установки их начальных значений: **Х ();**

Деструктор: ~ **Х ();**

Функция-друг печати: friend **void print(void);**

Функция-друг , решающий поставленную задачу: **friend Void Run(void);**

Код методов и функций– вне пространства определения класса

Программа иллюстрирует прямой и косвенный способы обращения к методам

1. Напишите программу, удаляющую из введенного фрагмента текста заданное слово, где бы оно не встречалось.
2. Во введенном тексте замените фрагмент, границы которого определяются заданными словосочетаниями, на содержимое некоторой строки.
3. Подсчитайте количество одинаковых букв, слов и словосочетаний в заданном тексте.
4. Удалите из текста повторяющиеся предложения.
5. Выведите на экран наиболее длинное предложение, содержащееся в тексте.
6. Замените в тексте все глаголы прошедшего времени троеточиями.
7. Выровняйте текст по ширине экрана путем равномерного добавления пробелов.
8. Измените порядок следования предложений в тексте на противоположный.
9. Определите, содержат ли два текста одинаковые предложения.
10. Выведите на экран встречающиеся в тексте собственные имена.
11. В заданном тексте определите среднее количество букв в слове, слов в предложении.
12. Выберите из текста все слова, в которых количество гласных букв равно количеству согласных.
13. Удалите из заданного фрагмента текста все слова, содержащие по две соседние гласные буквы.

5 Наследование классов, механизм виртуальных функций

Программа должна содержать:

* Базовый класс **Х**, включающий два элемента х1, х2 типа **int,** конструктор с параметрами для создания объектов в динамической области памяти, деструктор, виртуальные методы просмотра текущего и переустановки состояния объектов базового класса.
* Производный класс **У**, включающий один элемент у типа **int ,** конструктор с параметрами и списком инициализаторов, передающий данные конструктору базового класса, переопределенные методы просмотра и переустановки текущего состояния объектов и метод **Run,** определяющий:

1. Сумму компонент классов
2. Произведение компонент классов
3. Сумму квадратов компонент классов
4. Значение х1+х2 – у
5. Значение (х1+х2)/у
6. Значение (х1+х2)\*у
7. Значение х1\*у+х2
8. Значение х1+х2\*у
9. Произведение квадратов компонент класса
10. Значение х1\*х2+у
11. Значение х1\*х2/у
12. Значение х1\*х2-у

Программа должна продемонстрировать работу конструкторов базового и производного классов, начальное значение задается конструкторами, а переустановка их в новое состояние производится через косвенную адресацию.

## №6. Классы. Программирование алгоритмов со строками с использованием конструктора, деструктора, функций-друзей инициализации set() и вывода результатов print()

Пользовательский класс Х должен содержать необходимые элементы-данные, которые создаются в динамической области памяти.

Конструктор для их создания строк и установки их начальных значений: **Х ();**

Деструктор: ~ **Х ();**

Функция-друг печати: friend **void print(void);**

Функция-друг , решающий поставленную задачу: **friend Void Run(void);**

Код методов и функций– вне пространства определения класса

Программа иллюстрирует прямой и косвенный способы обращения к методам

Ввести с клавиатуры строку символов. Признак окончания ввода

строки - нажатие клавиши "Ввод". Программа должна определить длину

введенной строки L и если:

1. длина L нечетная, то удаляется символ, стоящий посередине строки;

2. длина L четная, то удаляются 2 первых и 2 последних символа;

3. длина L кратна 2-м, то удаляются все числа, которые делятся на 2;

4. длина L кратна 3-м, то удаляются все числа, делящиеся на 3;

5. длина L >10, то удаляются все цифры;

6. длина L >15, то удаляются все a..z;

7. длина L=10, то удаляются все A..Z;

8. длина L кратна 4-м, то первая часть строки меняется местами со

второй;

9. длина L кратна 5-и, то подсчитывается количество скобок всех

видов;

10. длина L >5-и, то выделяется подстрока до первого пробела;

11. длина L >6-и, то выделяется подстрока { } скобках;

12. длина L >10-и, то удаляется подстрока в [] скобках;

13. длина L >12-и, то удаляется подстрока до первой ( скобки;

14. длина L кратна 4-м, то выделяется подстрока после последнего

пробела;

15. длина L >5, то удаляются все точки.

16. длина L четная, то выделяется подстрока до первого пробела

17. длина L четная, то удаляется подстрока до первого пробела

18. длина L четная, то выделяется подстрока со второго пробела

19. длина L нечетная, то выделяется подстрока после первого пробела

20. длина L нечетная, то удаляется подстрока со второго пробела

21. длина L кратна 3, то удаляется каждый 3-й символ

22. длина L четная, то удаляется каждый 2-й символ

23. длина L нечетная, то

24. длина L четная, то выделяется подстрока до последнего пробела

25. длина L нечетная, то выделяется подстрока от последней цифры

На печать вывести исходную и преобразованную строки.

#include <iostream.h> // пример записи/считывания массива с объектами

#include <stdio.h> // поржденными классом в файл (используем наследование)

#include <conio.h>

#include <string.h>

struct B

{

char name[50]; // название

int y; // надбавка в процентах

}s[5];

void main(void)

{

int n; // объявили переменную для реального размера списка

char fname[10]; // строка для ввода имени файла

FILE \*f; // объявляем указатель на файл

printf("Формирование файла!!! Введи имя файла\n");

gets(fname);

f=fopen(fname,"w");

if(f==NULL) // блок контроля за открытием файла

{

puts("нельзя открыть файл");

getch();

return;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

//Формирование списка

printf("введите размер списка: ");

scanf("%d",&n);

printf("\n");

for (int i=0;i<n;i++)

{

printf("введите имя:");

fflush(stdin);

gets(s[i].name);

printf(" надбавка: ");

scanf("%d",&s[i].y);

fwrite(&s[i],sizeof(s[i]),1,f);

};

printf("ФАЙЛ СФОРМИРОВАН записано %d компонент. Press any key...",n);

fclose(f);

getch();

//-----------------------------------------------------

//блок просмотра существующего списка

printf("\n\n Просмотр!!! Введи имя файла:");

scanf("%s",fname);

f=fopen(fname,"r");

puts("\n СОСТАВ: \n\n");

i=0;

fread(&s[i],sizeof(s[i]),1,f);

while (!feof(f))

{

printf("Имя:%s \n Надбавка: %d\n\n",

s[i].name, s[i].y);

i++; fread(&s[i],sizeof(s[i]),1,f);

}

printf("\n прочитано %d компонент",i);

getch(); }

Написать программу, в которой описана иерархия классов: геометрические фигуры (круг, прямоугольник, треугольник). Реализовать методы вычисления площади и периметра фигуры. Продемонстрировать работу всех методов классов, предоставив пользователю выбор типа фигуры для демонстрации. 8. Написать программу, в которой описана иерархия классов: геометрические фигуры (эллипс, квадрат, трапеция). Реализовать методы вычисления площади и периметра фигуры. Продемонстрировать работу всех методов классов, предоставив пользователю выбор типа фигуры для демонстрации.

9. Написать программу, в которой описана иерархия классов: геометрические фигуры (ромб, параллелепипед, эллипс). Реализовать методы вычисления площади и периметра фигуры. Продемонстрировать работу всех методов классов, предоставив пользователю выбор типа фигуры для демонстрации.

10. Написать программу, в которой описана иерархия классов: геометрические фигуры (куб, цилиндр, тетраэдр). Реализовать методы вычисления объема и площади поверхности фигуры. Продемонстрировать работу всех методов классов, предоставив пользователю выбор типа фигуры для демонстрации.

11. Написать программу, в которой описана иерархия классов: геометрические фигуры (конус, шар, пирамида). Реализовать методы вычисления объема и площади поверхности фигуры. Продемонстрировать работу всех методов классов, предоставив пользователю выбор типа фигуры для демонстрации.

12. Написать программу, в которой описана иерархия классов: числа (целое, вещественное, комплексное). Реализовать методы сложения, вычитания, произведения, деления. Продемонстрировать работу всех методов классов, предоставив пользователю выбор типа основного и вспомогательного числа для демонстрации.

13. Написать программу, в которой описана иерархия классов: треугольник (равнобедренный, равносторонний, прямоугольный). Базовый класс должен иметь поля для хранения длины двух сторон и угла между ними. Реализовать методы вычисления площади и периметра фигуры. Продемонстрировать работу всех методов классов, предоставив пользователю выбор типа треугольника для демонстрации.

14. Написать программу, в которой описана иерархия классов: прогрессия (арифметическая, геометрическая). Классы наследники должны иметь поля для хранения параметров прогрессии. Реализовать функции вычисления i-го элемента, суммы элементов до n-го элемента. Продемонстрировать работу всех методов классов, предоставив пользователю выбор типа треугольника для демонстрации.

15. Написать программу, в которой описана иерархия классов: геометрические фигуры (круг, параллелепипед, трапеция). Реализовать методы вычисления площади и периметра фигуры. Продемонстрировать работу всех методов классов, предоставив пользователю выбор типа фигуры для демонстрации.

16. Написать программу, в которой описана иерархия классов: геометрические фигуры (эллипс, квадрат, треугольник). Реализовать методы вычисления площади и периметра фигуры. Продемонстрировать работу всех методов классов, предоставив пользователю выбор типа фигуры для демонстрации.

17. Написать программу, в которой описана иерархия классов: геометрические фигуры (шар, цилиндр, пирамида). Реализовать методы вычисления объема и площади поверхности фигуры. Продемонстрировать работу всех методов классов, предоставив пользователю выбор типа фигуры для демонстрации.

18. Написать программу, в которой описана иерархия классов: геометрические фигуры (куб, конус, тетраэдр). Реализовать методы вычисления объема и площади поверхности фигуры. Продемонстрировать работу всех методов классов, предоставив пользователю выбор типа фигуры для демонстрации.

19. Написать программу, в которой описана иерархия классов: геометрические фигуры (ромб, прямоугольник, эллипс). Реализовать методы вычисления площади и периметра фигуры. Продемонстрировать работу всех методов классов, предоставив пользователю выбор типа фигуры для демонстрации.