1. Дайте определение понятию «класс» и опишите его синтаксис. Приведите пример объявления.

Класс является абстрактным типом данных, определяемым программистом, и представляет собой модель реального объекта в виде данных и функций для работы с ними.

class /\*имя класса\*/

{

  private:

  /\* список свойств и методов для использования внутри класса \*/

  public:

  /\* список методов доступных другим функциям и объектам программы \*/

  protected:

  /\*список средств, доступных при наследовании\*/

};

1. Опишите работу с массивом структур. Как осуществляется доступ к полям структуры. Приведите пример.

struct person

{

    int age;

    char name[20];

};

int main()

{

    struct person people[] = {23, "Tom", 32, "Bob", 26, "Alice", 41, "Sam"};

  int n = sizeof(people)/sizeof(people[0]);

    return 0;

}

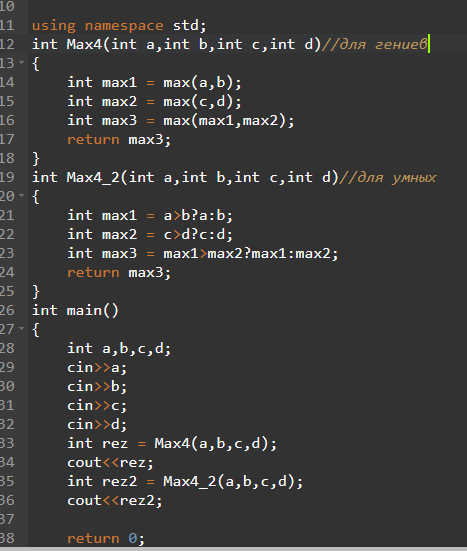
1. Дайте определение понятию «структура» и опишите его синтаксис. Как осуществляется доступ к полям. Приведите пример.

Структура – это составной тип данных, в котором под одним именем объединены данные различных типов. Отдельные данные структуры называются полями.

struct date  
{  
  int day;   
  char \*month;   
  int year;   
};

struct date bd={8,"июня", 1978};

1. Опишите объявление, описание и вызов функции. Написать функцию *Max4(A,B,C,D)* вещественного типа, находящую максимальное из четырех вещественных чисел *A*, *B*, *C* и *D*.



1. Дайте определение понятию «указатель». Приведите пример объявления указателя. Как передать указатель в функцию.

**Указатель** - это переменная или константа, содержащая адрес другой переменной.

double \*py = &y; // взять адрес переменной у и поместить его (адрес переменной y) в указатель ру

для использования указателя в переменной используется \* ????????????????????????????

1. Опишите выполнение операторов выбора, их формы и синтаксис, приведите пример. Напишите таблицу истинности.

if (/\*проверяемое условие\*/)

{

   /\*тело оператора выбора 1\*/;

} else

      {

      }

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **A** | **B** | **Not A** | **A or B** | **A and B** | **A xor B** |
| 0 | 0 | 1 | 0 | 0 | 0 |
| 0 | 1 | 1 | 1 | 0 | 1 |
| 1 | 0 |  | 1 | 0 | 1 |
| 1 | 1 | 0 | 1 | 1 | 0 |

1. Дайте определение понятию «список». Приведите структуру двухсвязного списка. Приведите пример объявления списка.

Связный список – одна из базовых структур данных. Ее часто сравнивают с массивом, так как многие другие структуры можно реализовать с помощью либо массива, либо связного списка. У этих двух типов есть преимущества и недостатки. Связный однонаправленный список состоит из группы узлов, которые вместе образуют последовательность. Каждый узел содержит две вещи: фактические данные, которые в нем хранятся (это могут быть данные любого типа) и указатель (или ссылку) на следующий узел в последовательности. Также существуют двусвязные списки: в них у каждого узла есть указатель и на следующий, и на предыдущий ему элемент в списке. Основные операции в связном списке включают добавление, удаление и поиск элемента в списке.

struct list  
{  
  int field; // поле данных  
  struct list \*next; // указатель на следующий элемент  
  struct list \*prev; // указатель на предыдущий элемент  
};

1. Опишите работу с двумерным массивом. Представление в памяти. Способы сортировки массивов. Приведите пример объявления двухмерного массива вещественного типа размером 5\*7.

\*массив располагается в оперативной памяти, а не на внешнем устройстве, как файлы;

\*элементы массива занимают подряд идущие ячейки, в отличие, например, от списков.

Double mas[5][7];

Выделяются несколько типов сортировок

Пузырьком

Сортировка выбором наименьшего элемента

Сортировка вставками

Хоара, Шелла

1. Дайте определение понятию «стек». Приведите структуру стека. Приведите пример объявления стека.

**Стек** – это базовая структура данных, которая позволяет добавлять или удалять элементы только в её начале. Она похожа на стопку книг: если вы хотите взглянуть на книгу в середине стека, сперва придется убрать лежащие сверху.

Struct stack

{

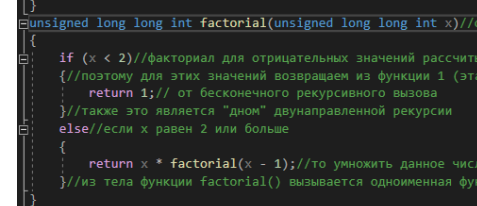
Int \* arr = new int[0];//рамер стека

Int head = 0;//индекс крайнего элемента стека

};

1. Опишите виды и синтаксис рекурсии. Объясните различие косвенной и прямой рекурсией. Объясните различие с циклом. Приведите пример рекурсивной функции расчета факториала.

Рекурсия может быть как прямой , когда программа вызывает саму себя, так и непрямой( косвенной ) , когда программа вызывает другую программу, а та в свою очередь, вызывает первую программу.



1. Дайте определение понятию «переменная», какое имя может быть у переменной. Приведите пример объявления переменной.

Переменная это номер байта

Int x = 666;

1. Опишите синтаксис класса. Объясните для чего используется оператор расширения области видимости “::”. Приведите пример объявления класса.

class List

{

Node\* Head, \* Tail; // first element, last element

int size; // count elements in list

public:

List() :Head(NULL), Tail(NULL), size(0) {} // initialize elements to zero with constructor

~List(); // destructor prototype, created below

void Add(int x); // def prototype to add elements in list

void Show(int size); // def prototype to show (print) elements in list

int Count(); // def prototype to return count of elements in list

};

List::~List() // destructor

{

while (size != 0)

{

Node\* temp = Head->Next;

delete Head;

Head = temp;

size--;

}

}

1. Дайте определение понятию «бинарное дерево», приведите пример структуры бинарного дерева. Приведите пример объявления бинарного дерева.

*Бинарное дерево* – это конечное множество элементов, которое либо пусто, либо содержит элемент (*корень*), связанный с двумя различными бинарными деревьями, называемыми *левым и правым поддеревьями*. Каждый элемент бинарного дерева называется *узлом*. Связи между узлами дерева называются его *ветвями*.

*struct BinareTree // структура узла бинарного дерева*

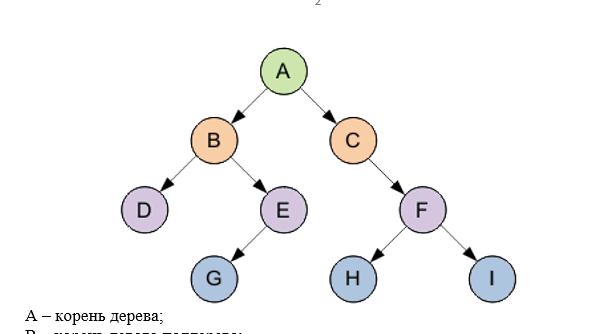
*{*

*DataField data\_field; //поле для данных (информационной части)*

*BinareTree\* left = nullptr; // левый потомок*

*BinareTree\* right = nullptr; // правый потомок*

*};*



1. Перечислите логические операции, составьте таблицу истинности.

Существуют несколько типов логических операций

& и

| или

^ исключающее или

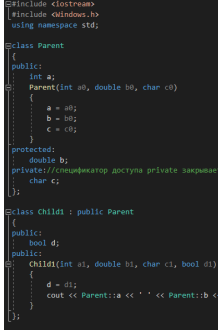
|  |  |  |
| --- | --- | --- |
| & | + | - |
| + | + | - |
| - | - | - |

|  |  |  |
| --- | --- | --- |
| | | + | - |
| + | + | + |
| - | + | - |

|  |  |  |
| --- | --- | --- |
| ^ | + | - |
| + | - | + |
| - | + | - |

1. Дайте определение понятию «наследование». Какие поля и методы наследуются. Перечислите типы наследования. Приведите пример наследования.

Наследование – это один из трех основных принципов ООП, предполагающий создание иерархии классов (объектов классов), в которой объекты-потомки наследуют все члены (свойства и поведение) своих классов-предков. Описывать заново явно в коде унаследованные от родителей члены у потомков не нужно. Потомки имеют унаследованные и свои «личные» свойства и поведения.



1. Опишите выполнение операторов цикла, их формы и синтаксис. Приведите пример вычисления суммы чисел от 1 до 100 включительно.

int sum = 0;

for(int i = 0;i<101;i++)

{

sum+=i;

}

1. Дайте определение понятию «хеширование». Приведите пример хеш-функции.

Процесс преобразования данных называется хешированием.

Int hash(int k)

{

Return (k%10);

}

1. Перечислите типы данных, правила записи идентификаторов. Приведите пример.

Существую такие типы данных как: bool, int , double, char, float

**Идентификатор** может состоять только из букв (нижнего и верхнего регистра) латинского алфавита, цифр и символов подчёркивания. Это означает, что все другие символы и пробелы — запрещены.

1. Дайте определение понятию «функция». Укажите синтаксис объявления и определения функции. Приведите пример объявления и определения функции.

Функция – изолированный именованный блок кода, имеющий определенное назначение.

типВозвращаемогоРезультата имяФункции (списокВходныхПараметров) // заголовок функции

{

операторы; // тело функции (код тела функции)

return возвращаемыйРезультат;

}

Int sum (int I)

{

Return I;

}

1. Дайте определение понятию «класс». Опишите принцип работы с конструктором (с параметрами, без параметров, копирования, по умолчанию).

Класс является абстрактным типом данных, определяемым программистом, и представляет собой модель реального объекта в виде данных и функций для работы с ними.

Class Human

{private:

Int age;

Public:

Human()//конструктор без параметров

{

}

Human(int x)//конструктор с параметрами

{

SetAge(x);

}

};

1. Дайте определение понятию «полиморфизм». Объясните принцип полиморфизма, опишите синтаксис. Приведите пример.

Полиморфизм – (от греч. poli – много, morpheo – форма, поведение) – один из трех основных принципов ООП, заключающийся в возможности определения единого по имени действия (поведения), применимого ко всем объектам иерархии наследования (иерархии классов), причем каждый объект может реализовать это действие (поведение) собственным способом (в зависимости от того, к какому именно классу из данной иерархии этот объект относится).

Виртуальные методы – это методы, объявляемые в родительском классе с ключевым словом virtual и переопределяемые (перегружаемые) в дочерних классах.

1. Дайте определение понятию «тип данных». Перечислите типы данных по типу возвращаемых значений функций.

Каждая переменная имеет определенный тип. Тип данных определяет какие значение может принимать переменная.

Существую такие типы данных как: bool, int , double, char, float

Функция может возвращать любые типы данных

1. Дайте определение понятию «перегрузка функций». Приведите пример перегрузки функции.

Перегрузка операторов (перегрузка операций) – случай полиморфизма в С++, когда общеизвестный допустимый оператор приобретает новый функционал взамен старого при условии, что этот оператор применяется с объектом класса, для которого он перегружен.

class Counter

{

public:

Counter operator ++ (int)//перегрузка оператора # для класса ААА

{

Count++;

Return \*this;

}

};

1. Дайте определение понятию «минимального остовного дерева». Опишите алгоритм Р. Прима. Приведите пример данного алгоритма.

хз

1. Дайте определение понятию «граф» и приведите способы его хранения.

**Граф** – совокупность точек, соединенных линиями. Точки называются **вершинами**, или узлами, а линии – **ребрами**, или дугами.

1. Дайте определение понятию «функция». Объясните различие передачи параметров по значению и ссылке. Приведите пример передачи параметров по значению и ссылке.

Функция – изолированный именованный блок кода, имеющий определенное назначение.

При передаче параметров по ссылке передается ссылка на объект, через которую мы можем манипулировать самим объектов, а не просто его значением.

Передача ссылки происходит при помощи & перед именем переменной

void square(int &a, int &b);

1. Опишите работу комментариев и способы их записей. Для чего нужны комментарии, приведите пример их в коде.

**Комментарий** – это заметка, удобная для чтения программисту, которая вставляется непосредственно в исходный код программы. Комментарии игнорируются компилятором и предназначены только для использования программистом.

Комментарии можно оставлять в конце строки после символа // или комбинацией /\* <текст> \*/для созданий многострочного комментария

1. Опишите алгоритм Дж. Краскала. Дайте определение понятию «минимального остовного дерева». Приведите пример данного алгоритма.
2. Дайте определение понятию «абстрактный класс». Приведите пример абстрактного класса.

Абстрактный класс в объектно-ориентированном программировании — базовый класс, который не предполагает создания экземпляров

1. Опишите синтаксис класса (с методами внутри класса и вне класса). Объясните для чего используется оператор расширения области видимости “::”. Приведите пример.

Используется для наследования классов

1. Дайте определение понятию «очередь». Приведите структуру очереди. Приведите пример объявления очереди.

**Очередь** – эту структуру можно представить, как очередь в продуктовом магазине. Первым обслуживают того, кто пришёл в самом начале – всё как в жизни. Очередь устроена по принципу FIFO (First Input – First Output, «первым пришёл – первым уйдёт»). Это значит, что удалить очередной элемент можно только после того, как были убраны все ранее добавленные перед ним элементы. Очередь позволяет выполнять две основные операции: добавлять элементы в конец очереди и удалять первый элемент из очереди.

struct Spis {

int info;

Spis \*Next;

};

1. Опишите инициализацию и присвоение указателей. Приведите пример инициализации и присвоение указателей

Вроде где-то раньше было

1. Дайте определение понятию «массив». Приведите пример объявления динамического массива.

Массив — это сложный (составной, структурированный) тип данных, который характеризуется следующим:

• элементы массива имеют одинаковый тип в отличие от структур struct, поэтому каждый элемент массива занимает одинаковый объём памяти;

• массив располагается в оперативной памяти, а не на внешнем устройстве, как файлы;

• элементы массива занимают подряд идущие ячейки, в отличие, например, от списков

Int I;

Cin>n;

Int \* mas = new int [n];

1. Перечислите ситуации использования оператора break. Приведите пример ситуации использования оператора break.

Break используется для входа из цикла. Может применяться для выхода при достижении определенного значения в цикле

1. Дайте определение понятию «циклический (кольцевой) список». Приведите структуры списка. Приведите пример объявления кольцевого списка.

**Кольцевой список** (кольцо) – список, в котором последний элемент хранит указатель не на NULL, а на первый элемент этого же списка, в результате чего список становится «закольцованным» (циклическим, круговым) и по нему можно бесконечно ходить, поскольку у него конец переходит в начало.

Struct list

{

Int field;//поле данных

List\* next;//следующий элемент

List\* prev;//предыдущий

};

1. Перечислите типы констант и способы их определения. Приведите пример.

**Константы** бывают числовые, символьные и строковые

Const char text = “Q”;

Const int I = 0;

Const double r = 3.5;

1. Дайте определение понятию «инкапсуляция». Приведите пример инкапсуляция.

**Инкапсуляция** – это один из трех основных принципов ООП, предполагающий сокрытие деталей реализации класса от доступа извне данного класса (и, соответственно, извне данного объекта класса) и недопущение изменения внутреннего состояния объекта класса (значений, хранимых в его полях) без ведома самого данного объекта, а также недопущение изменений в состоянии объекта класса, которые приведут к его некорректной работе и/или уничтожению (не путать с деструктором, который является членом экземпляра класса и для того и существует, чтобы очищать оперативную память от данного объекта класса, уничтожая тем самым этот объект).

1. Опишите работу с одномерным массивом. Представление в памяти. Способы сортировки массивов. Приведите пример.

Хз что тут описывать вроде все всё знают

1. Дайте определение понятию «алгоритм». Какие есть способы записи алгоритма, Приведите пример алгоритма.

Алгоритм – последовательность действия приводящие к решению задач.

Существуют несколько типов записи алгоритмов

Человеческий

Код программы

Графический

Виды алгоритмов

Линейный

Разветвляющий

циклический

1. Опишите принцип работы перегрузки операций, приведите синтаксис. Приведите пример перегрузки операции «+».

Перегрузка операторов (перегрузка операций) – случай полиморфизма в С++, когда общеизвестный допустимый оператор приобретает новый функционал взамен старого при условии, что этот оператор применяется с объектом класса, для которого он перегружен.

Counter operator + (Counter c1, Counter c2)

{

    return Counter(c1.seconds + c2.seconds);

}

1. Дайте определение понятию «хеш-таблица». Опишите структуру хеш-таблицы. Приведите пример объявления хеш-таблицы.

**Хеш**-**таблица** (hash table) — это специальная структура данных для хранения пар ключей и их значений. По сути это ассоциативный массив, в котором ключ представлен в виде **хеш**-функции

Примера нету

1. Опишите способы доступа к элементам массива. Приведите пример объявления двухмерно массива и проинициализируйте элементы массива при помощи датчика случайных чисел в диапазоне (-10, 10).

Доступ к элементам массива в языке С++ осуществляется при помощи индекса или указателем

#include <stdlib>

#include <time.h>

Srand(time(NULL));

Const int m = 10;

Const int n = 10;

Int a = -10;

Int b = 10;

Int mas[m][n];

For(int I = 0;i<m;i++)

{

For(int j = 0;j < n;j++)

{

Mas[i][j] = a+rand()%(b-a+1)

}

}

1. Дайте определение понятию «виртуальная функция». Приведите пример.

Виртуальные методы – это методы, объявляемые в родительском классе с ключевым словом virtual и переопределяемые (перегружаемые) в дочерних классах

virtual void Show()

{

cout << a << endl;

}

1. Объясните разницу между классом и структурой. Приведите пример объявления класса «Студент».

Class Stud

{

private:

Int age;

Int mark;

public:

Stud(int a, int m)

{

SetAge(a);

SetMark(m);

}

GetAge()//геттер

{

return age;

}

SetAge(int a)//сеттер

{

age = a;

}

Getmark()

{

return mark;

}

SetMark(int m)

{

mark = m;

}

};

Если делеать без геттера и сеттера

Public:

Stud(int a, int m)

{

age= a;

Mark = m;

}

1. Дайте определение понятию «кратчайший путь в графе» Опишите алгоритм Флойда.

Кратчайший путь в графе это путь от точки а к точке б с минимальным весом ребер.

1. Опишите работу с текстовыми файлами. Перечислите функции для работы с ними. Приведите пример как вывести все содержимое текстового файла на экран.

Ifstream f(path, ios::in);

Int x ;

f >> x;

while(!f.eof())

{

Cout<<x<<endl;

f>>x;

}

F.close();

1. Дайте определение понятию «идентификаторы». Приведите пример объявлении переменной, функции, класса.

Int Test = 5;

**Идентификатор** — **это** последовательность символов, используемая для обозначения одного из следующих элементов: Имени объекта или переменной Имени класса, структуры или объединения

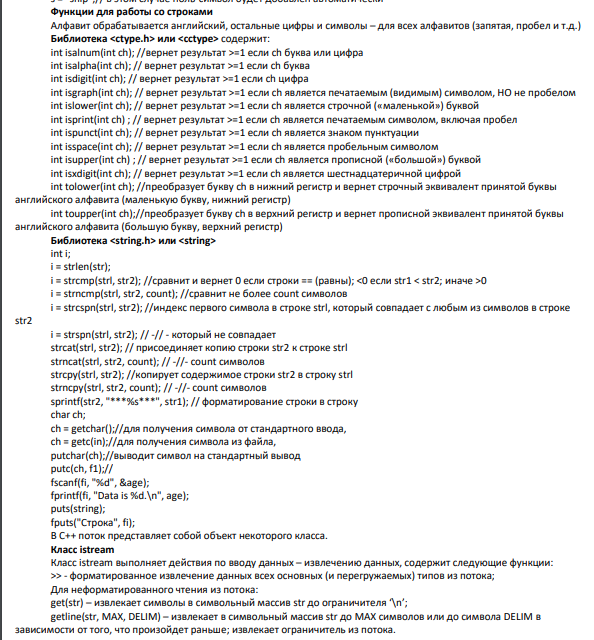
Int Test = 5;

Void Func();

Class Stud;

1. Опишите работу с символьными переменными и строками. Перечислите функции для работы с ними. Приведите пример подсчета цифр в текстовой строке(string).

Ты серьезно, шарпы ван лав



Бери что хочешь

1. Дайте определение понятию «класс» и опишите его синтаксис. Перечислите уровни доступа, объясните различие. Приведите пример объявления класса.

Класс является абстрактным типом данных, определяемым программистом, и представляет собой модель реального объекта в виде данных и функций для работы с ними.

1. Дайте определение понятию «кратчайший путь в графе» Опишите алгоритм Дейкстры.

Кратчайший путь в графе это путь от точки а к точке б с минимальным весом ребер.

1. Дайте определение понятию «выражение». Перечислите операции в порядке их приоритета.

**Выражения** — **это** последовательности операторов и операндов, используемые в следующих целях. Вычисление значения из операндов. Назначение объектов или функций. Создание побочных эффектов.

Скорее всего + - / \*

1. Опишите объявление, описание и вызов функции. Объясните различие формальных и фактических параметров функций. Приведите пример объявления и определения функции.

Вызов функции kek которая принимает 3 значения

X = kek(1,2,3);



1. Дайте определение понятию «константа» и приведите пример объявления константы.

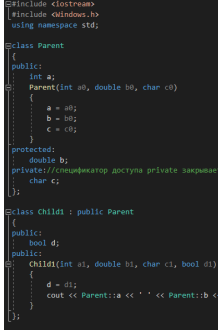
**Константа** — это ограниченная последовательность символов алфавита языка, представляющая собой изображение фиксированного (неизменяемого) объекта.

1. Опишите принцип работы перегрузки функций. Приведите пример перегрузки функции нахождения максимального числа функции от 1, 2 и 3 аргументов.
2. Дайте определение понятию «деструктор». В каких случаях вызывается деструктор.

деструктор – особый метод класса, который уничтожает объекты своего класса, когда они уже, по мнению программиста, не нужны.

1. Объясните принцип наследования, опишите синтаксис. Приведите пример наследования.

Наследование – это один из трех основных принципов ООП, предполагающий создание иерархии классов (объектов классов), в которой объекты-потомки наследуют все члены (свойства и поведение) своих классов-предков.



1. Дайте определение понятию «объект». Как осуществляться доступ к полям и методам класса. Приведите пример объявления объекта класса.
2. Опишите объявление, описание и вызов функции. В чем различие процедуры от функции. Приведите пример функции (аргументы которой три вещественных числа), которая возвращает аргументы по возрастанию.
3. Дайте определение понятию «метод пошаговой детализации».
4. Объясните различие позднего и раннего связывания. Приведите пример позднего и раннего связывания.

Частное учреждение образования

«Колледж бизнеса и права»

**ОСНОВЫ АЛГОРИТМИЗАЦИИ И ПРОГРАММИРОВАНИЕ**

**Задачи** **к экзамену для учащихся 2 курса дневной формы получения образования специальности 2-40 01 01 «Программное обеспечение информационных технологий»**

1. Создать класс «Треугольник» и класс «Точка». Реализовать конструкторы с параметрами и конструктор копирования. Объявить массив из  объектов класса Point. Написать функцию, определяющую, какая из точек лежит внутри, а какая снаружи треугольника. Написать программу, демонстрирующую все разработанные элементы.
2. Определить класс «Стек» хранящее вещественное число. Реализовать конструктор копирования. Объявить объект класса. Добавить в стек несколько элементов извлечь пару элементов и проверить пуст ли стек (т.е. реализовать методы push, pop, is\_empty). Написать программу, демонстрирующую все разработанные элементы. Продемонстрировать все разработанные функции.
3. Определить класс «Очередь» хранящее символ. Реализовать конструктор копирования. Объявить объект класса. Добавить в очередь несколько элементов извлечь пару элементов и проверить пуста ли очередь (т.е. реализовать методы push, pop, is\_empty). Написать программу, демонстрирующую все разработанные элементы. Продемонстрировать все разработанные функции.
4. Определить класс «Дробь» в виде пары  и использующий его класс «Смешанная дробь» в виде целой и дробной части. Реализовать конструкторы с параметрами и конструктор копирования. Перегрузить операции ввода/вывода и сложения/умножения. Объявить массив из *k* дробей, ввести. Вывести наибольшую из дробей. Написать программу, демонстрирующую все разработанные элементы.
5. Создать шаблон класса Matrix. Реализовать конструкторы с параметрами и конструктор копирования. Перегрузить основные операции: сложения, умножения матриц. Массив матриц упорядочить так, чтобы норма возрастала . Написать программу, демонстрирующую все разработанные элементы.
6. Создать абстрактный класс «Книга» и производные от него классы «Справочник» и «Книга по информатике». Реализовать конструкторы с параметрами и конструктор копирования. Используя виртуальную функцию ShowBook( ) реализовать полиморфизм. Все записи с книгами по информатике вывести на экран. Написать программу, демонстрирующую все разработанные элементы.

{

Автор

Название

Год

Количество страниц

Вид издания

}

1. Номиналы рублей могут принимать значения 10, 20, 50, 100, 500, 1000, 10000, 20000, 50000, 100000. Создать класс Money для работы с денежными суммами. Реализовать конструкторы с параметрами и конструктор копирования. Сумма должна быть представлена полями-номиналами, значениями которых должно быть количество купюр данного достоинства. Реализовать через перегрузку операций сложение сумм, вычитание сумм, и операции сравнения (==, !=, <, >). Написать программу, демонстрирующую все разработанные элементы.
2. Создать класс Fraction для работы с десятичными дробными числами. Число должно быть представлено двумя полями: int (целая часть и знак числа), дробная часть – unsigned. Реализовать конструкторы с параметрами и конструктор копирования. Перегрузить арифметические операции: сложение, вычитание, умножение и операции сравнения (==, !=, <, >). Написать программу, демонстрирующую все разработанные элементы.
3. Создать класс Polynom для работы с многочленами. Реализовать конструкторы с параметрами и конструктор копирования. Коэффициенты должны быть представлены динамическим массивом. Младшая степень имеет меньший индекс. Перегрузить арифметические операции (+,\*) и операции сравнения (== , !=). Реализовать методы для вычисления значения полинома для заданного значения x. Написать программу, демонстрирующую все разработанные элементы.
4. Реализовать класс Triangle (треугольник) с полями-координатами вершин. Вычислить площадь и периметр треугольника. Реализовать конструкторы с параметрами и конструктор копирования. Для представления координат вершин используйте класс Point. Проверить, является ли треугольник равносторонним, равнобедренным, прямоугольным. Написать программу, демонстрирующую все разработанные элементы.
5. Создать шаблон класса MatVector размерности . Реализовать конструкторы с параметрами и конструктор копирования. Создать метод для вывода на экран вектора. Перегрузить операции сложения векторов, умножения на число, скалярного умножения векторов. Написать программу, демонстрирующую все разработанные элементы.
6. Создать класс Triad (тройка чисел); определить с помощью перегрузки операции ++ операцию увеличения всех полей одновременно на 1. Реализовать конструкторы с параметрами и конструктор копирования. Определить производный класс Time (время) с полями: часы, минуты и секунды. Перегрузить операции (++, --) увеличения (уменьшения) времени на 1 секунду и операции + и - увеличения (уменьшения) времени на n секунд. Написать программу, демонстрирующую все разработанные элементы.
7. Создать классы Point и Line. Реализовать конструкторы по умолчанию, с параметрами и конструктор копирования. Реализовать методы set, get. Объявить массив из n объектов класса Point и определить какая из точек находится дальше всех от прямой. Нахождение расстояния реализовать в методе. Написать программу, демонстрирующую все разработанные элементы.
8. Определить класс «Матрица» размерности . Реализовать конструкторы с параметрами и конструктор копирования. Реализовать метод для установки новой размерности матрицы и метод для установки коэффициентов матрицы. Объявить массив из  объектов. Написать функции, вычисляющие первую и вторую нормы матрицы , . Определить, какая из матриц имеет наименьшую первую и вторую нормы. Написать программу, демонстрирующую все разработанные элементы.
9. Определить классы «Треугольник». Реализовать конструкторы с параметрами и конструктор копирования. Определить, какой из m введенных «Треугольников» имеет наибольшую и наименьшею площадь. . Написать программу, демонстрирующую все разработанные элементы.
10. Создать класс Complex для работы с мнимыми числами. Реализовать конструкторы с параметрами и конструктор копирования. Используя для представления числа два числа реальная и действительная часть числа. Перегрузить арифметические операции (+, -, \*, /) и операции сравнения (==, !=, <, >). Написать программу, демонстрирующую все разработанные элементы.
11. Создать класс Complex для работы с мнимыми числами. Реализовать конструкторы с параметрами и конструктор копирования. Используя для представления числа два числа реальная и действительная часть числа. Перегрузить арифметические операции (+, -, \*, /) и операцию сравнения (==). Реализовать метод для вывода на экран числа. Написать программу, демонстрирующую все разработанные элементы.
12. Создать класс Pair (пара чисел). Реализовать конструкторы с параметрами и конструктор копирования, определить методы изменения полей и вычисления произведения чисел. Определить производный класс RightAngled (прямоугольный треугольник) с полями-катетами. Определить методы вычисления гипотенузы и площади треугольника. Написать программу, демонстрирующую все разработанные элементы.
13. Создать класс Triad (тройка чисел). Реализовать конструкторы с параметрами и конструктор копирования, определить методы изменения полей и вычисления суммы чисел. Определить производный класс Triangle (треугольник) с полями-сторонами. Определить методы для определения: существует ли такой треугольник, для вычисления периметра и площади. Написать программу, демонстрирующую все разработанные элементы.
14. Создать класс Man (человек) с полями: имя, возраст, пол и вес (реализовать конструкторы с параметрами и конструктор копирования). Определить методы переназначения имени, изменения возраста и изменения веса. Создать производный класс Student (реализовать конструкторы с параметрами и конструктор копирования), имеющий поле для года обучения. Определить методы переназначения и увеличения года обучения. Перегрузить операции ++, --. Написать программу, демонстрирующую все разработанные элементы.
15. Создать класс PairNumber (пара вещественных чисел). Реализовать конструкторы с параметрами и конструктор копирования; определить методы изменения полей и вычисления произведения чисел. Определить производный класс Rectangle (прямоугольник) (реализовать конструкторы с параметрами и конструктор копирования) с полями-сторонами. Определить методы вычисления периметра и площади прямоугольника. Написать программу, демонстрирующую все разработанные элементы.
16. Создать класс Time для работы со временем в формате «часы:минуты:секунды». Реализовать конструкторы с параметрами и конструктор копирования. Класс должен включать в себя конструкторы инициализации: числами, строкой (например, «22:59:59»), секундами и временем. Перегрузить операции: вычисление разницы между двумя моментами времени в секундах, сложение времени и заданного количества секунд, вычитание из времени заданного количества секунд. Написать программу, демонстрирующую все разработанные элементы.
17. Создать класс Triangle для представления треугольника. Реализовать конструкторы с параметрами и конструктор копирования. Поля данных должны включать координаты угловых точек треугольника. Требуется реализовать операции: получения и изменения полей данных, вычисления периметра, а также определения вида треугольника (равносторонний, равнобедренный или прямоугольный). Написать программу, демонстрирующую все разработанные элементы.
18. Создать класс Money для работы с денежными суммами. Реализовать конструкторы с параметрами и конструктор копирования. Число должно быть представлено двумя полями: типа int для рублей и типа char для копеек. Дробная часть (копейки) при выводе на экран должна быть отделена от целой части запятой. Перегрузить операции для сложения, вычитания и операции сравнения (==, !=, <, >). Написать программу, демонстрирующую все разработанные элементы.
19. Определить класс «Стек» хранящее текстовую строку. Реализовать конструктор копирования. Объявить объект класса. Добавить в стек несколько элементов извлечь пару элементов и проверить пуст ли стек (т.е. реализовать методы push, pop, is\_empty). Написать программу, демонстрирующую все разработанные элементы.
20. Определить класс «Очередь» хранящее хранящее текстовую строку. Реализовать конструктор копирования. Объявить объект класса. Добавить в очередь несколько элементов извлечь пару элементов и проверить пуста ли очередь (т.е. реализовать методы push, pop, is\_empty). Написать программу, демонстрирующую все разработанные элементы.
21. Определить класс «Binary» (бинарное дерево) хранящее, текстовую строку. Реализовать конструктор копирования и конструктор создания бинарного дерева. Объявить объект класса. Реализовать методы для добавления данных в дерево, вывод всего дерева на экран, и метод удаления узла бинарного дерева. Написать программу, демонстрирующую все разработанные элементы.
22. Определить класс «Binary» (бинарное дерево), хранящее текстовую строку. Реализовать конструктор копирования и конструктор создания бинарного дерева. Объявить объект класса. Реализовать методы для добавления данных в дерево, вывод всего дерева на экран, и метод удаления узла бинарного дерева. И реализовать метод для получения количества узлов в бинарном дереве. Написать программу, демонстрирующую все разработанные элементы.
23. Определить класс «List\_1» (Циклический однонаправленный список) хранящий пару чисел (целое и вещественное). Реализовать конструктор копирования и конструктор создания списка. Объявить объект класса. Реализовать методы для добавления данных в список, вывод всего списка на экран, и метод удаления элемента из списка. И реализовать метод для получения количества элементов в однонаправленном списке. Написать программу, демонстрирующую все разработанные элементы.
24. Определить класс «List\_2» (двусвязный список) хранящий вещественное число и текстовую строку. Реализовать конструктор копирования и конструктор создания двусвязного списка. Объявить объект класса. Реализовать методы для добавления данных в список, вывод всего списка на экран, и метод удаления элемента из списка. И реализовать метод для получения количества элементов в двусвязном списке. Написать программу, демонстрирующую все разработанные элементы.