Условие:

Определить абстрактный базовый класс "Летательный аппарат" содержащий данные, описывающие название, стартовую массу аппарата. В этом классе должна быть чистая виртуальная функция print для вывода данных. Затем определите производный класс «Ракета», в котором будут новые элементы данных – количество ступеней, масса полезной нагрузки. Определите класс «Вертолёт», производный от «Летательного аппарата», в котором будут новые элементы данных – количество человек в экипаже. Вывод данных должен осуществляться с помощью виртуальной функции print.

Написать программу, создающую список объектов этих классов в динамической

памяти.

Исходный код программы:  
  
#include **<iostream>**#include **<algorithm>**#include **<vector>  
using namespace** std;  
  
*//Класс летательных аппаратов***class** FlyApparatClass {  
**protected**:  
 **int** mass;  
 string name;  
  
**public**:  
 FlyApparatClass(**int** m = 0, string n = **""**)  
 : mass(m),  
 name(n)  
 {  
 }  
 **virtual void** Print() **const** = 0;  
 **virtual** ~FlyApparatClass() {}  
};  
  
*//Класс ракеты***class** RocketClass : **public** FlyApparatClass {  
 **int** netto, amount;  
  
**public**:  
 **void** GetData(**int** m = 0, string \_model = **""**, **int** n = 0, **int** a = 0)  
 {  
 mass = m;  
 netto = n;  
 amount = a;  
 name = \_model;  
 }  
 **void** Print() **const** {  
 cout<<**"\n\*Класс ракеты\*"**;  
 cout << **"\nМасса: "**<< mass << **"\nНазвание: "** << name << **"\nМасса полезной нагрузки: "** << netto << **"\nКол-во ступеней: "**<<amount << **"\n"**;  
 }  
};  
  
**class** HelicopterClass : **public** FlyApparatClass {  
 **int** units;  
  
**public**:  
 HelicopterClass(**int** m, string n, **int** u = 0)  
 : FlyApparatClass(m, n)  
 , units(u)  
 {  
 }  
 **void** Print() **const** {   
 cout<<**"\n\*Класс вертолета\*\n"**;  
 cout<<**"Масса: "**<<mass<<**"\nНазвание: "**<<name<<**"\nКол-во человек в экипаже\n"**<<units<<**"\n"**;  
 }  
 **bool operator**<(**const** HelicopterClass& x) **const** {   
 **return** units < x.units;  
 }  
};  
  
**int** main()  
{  
 RocketClass a;  
 FlyApparatClass\* p = &a;  
 a.GetData(48, **"РАКЕТА №1"**, 20, 3);  
 p->Print();  
  
 *//Вектор для пацанов* vector<HelicopterClass> chop\_array;  
 chop\_array.push\_back(HelicopterClass(10, **"Вертолет №1"**, 12));  
 chop\_array.push\_back(HelicopterClass(4, **"Вертолет №2"**, 5));  
 chop\_array.push\_back(HelicopterClass(9, **"Вертолет №3"**, 16));  
 sort(chop\_array.begin(), chop\_array.end());   
 **for** (**unsigned** i = 0; i < chop\_array.size(); i++)  
 chop\_array[i].Print();  
 **return** 0;  
}

Скриншоты программы:

