|  |
| --- |
| 1、#include<iostream> |
|  | using namespace std; |
|  | class Date |
|  | {private: |
|  | int year; |
|  | int month; |
|  | int day; |
|  | public: |
|  | void SetDate(int ,int ,int ); |
|  | void Display( ); |
|  | int GetYear( ); |
|  | int GetMonth( ); |
|  | int GetDay( ); |
|  | }; |
|  | void Date::SetDate(int y,int m,int d) |
|  | { year=y; |
|  | month=m; |
|  | day=d; |
|  | } |
|  | int Date::GetYear( ) |
|  | { return year; |
|  | } |
|  | int Date::GetMonth( ) |
|  | { return month; |
|  | } |
|  | int Date::GetDay( ) |
|  | { return day; |
|  | } |
|  | void Date::Display() |
|  | { cout << year<< "-" << month << "-" << day << endl; |
|  | } |

|  |
| --- |
| 2、#include<iostream> |
|  | using namespace std; |
|  | class Date |
|  | {private: |
|  | int year; |
|  | int month; |
|  | int day; |
|  | public: |
|  | void SetDate(int y, int m , int d) |
|  | { year = y; |
|  | month = m; |
|  | day = d; |
|  | } |
|  | int GetYear( ) |
|  | { return year; |
|  | } |
|  | int GetMonth( ) |
|  | { return month; |
|  | } |
|  | int GetDay( ) |
|  | { return day; |
|  | } |
|  | void Display( ) |
|  | { cout<< year << "-" << month << "-" << day << endl; |
|  | } |
|  | }; |

|  |
| --- |
| 3、#include <iostream> |
|  | using namespace std; |
|  | /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*类\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/ |
|  | class Date |
|  | { int year, month, day ; |
|  | public : |
|  | Date(int y=2007, int m=1, int d=1) ; |
|  | Date(const Date &date); |
|  |  |
|  | int GetYear( ); |
|  | int GetMonth( ); |
|  | int GetDay( ); |
|  | void Display ( ); |
|  | }; |
|  | /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*构造函数\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/ |
|  | Date::Date(int y, int m, int d) |
|  | { year = y; |
|  | month = m; |
|  | day = d; |
|  | cout<<"Constructor called.\n"; |
|  | } |
|  |  |
|  | Date::Date( const Date &date) |
|  | { year = date.year; |
|  | month = date.month; |
|  | day = date.day; |
|  | cout<<"Copy Constructor called.\n"; |
|  | } |
|  | /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*普通函数\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/ |
|  | int Date:: GetYear( ) |
|  | { return year; |
|  | } |
|  |  |
|  | int Date:: GetMonth( ) |
|  | { return month; |
|  | } |
|  |  |
|  | int Date:: GetDay( ) |
|  | { return day; |
|  | } |
|  | void Date::Display() |
|  | { cout<<year<<"-"<<month<<"-"<<day<<endl; } |

|  |
| --- |
| 4、#include <iostream> |
|  | using namespace std; |
|  | class Date |
|  | { int year, month, day ; |
|  | public : |
|  | Date(int y=2007, int m=1, int d=1) ; |
|  | Date(const Date &date); |
|  | ~Date( ); |
|  | int GetYear( ); |
|  | int GetMonth( ); |
|  | int GetDay( ); |
|  | void Display ( ); |
|  | }; |
|  | Date::Date(int y, int m, int d) |
|  | { year = y; |
|  | month = m; |
|  | day = d; |
|  | cout<<"Constructor called.\n"; |
|  | } |
|  |  |
|  | Date::Date( const Date &date) |
|  | { year = date.year; |
|  | month = date.month; |
|  | day = date.day; |
|  | cout<<"Copy Constructor called.\n"; |
|  | } |
|  |  |
|  | Date::~Date( ) |
|  | {cout<<"Destructor called.\n"; |
|  | } |
|  |  |
|  | int Date:: GetYear( ) |
|  | { return year; |
|  | } |
|  | int Date:: GetMonth( ) |
|  | { return month; |
|  | } |
|  | int Date:: GetDay( ) |
|  | { return day; |
|  | } |
|  | void Date::Display() |
|  | { cout<<year<<"-"<<month<<"-"<<day<<endl; } |

|  |
| --- |
| 5/#include <iostream> |
|  | using namespace std; |
|  | class Date |
|  | { int year, month, day ; |
|  | public : |
|  | Date(int y=2007, int m=1, int d=1) ; |
|  | Date(const Date &date); |
|  | ~Date( ); |
|  | int GetYear( ); |
|  | int GetMonth( ); |
|  | int GetDay( ); |
|  | void Display ( ); |
|  | void ModifyDate( int, int, int); |
|  | }; |
|  | Date::Date(int y, int m, int d) |
|  | { year = y; |
|  | month = m; |
|  | day = d; |
|  | cout<<"Constructor called.\n"; |
|  | } |
|  |  |
|  | Date::Date( const Date &date) |
|  | { year = date.year; |
|  | month = date.month; |
|  | day = date.day; |
|  | cout<<"Copy Constructor called.\n"; |
|  | } |
|  |  |
|  | Date::~Date( ) |
|  | {cout<<"Destructor called.\n"; |
|  | } |
|  |  |
|  | int Date:: GetYear( ) |
|  | { return year; |
|  | } |
|  | int Date:: GetMonth( ) |
|  | { return month; |
|  | } |
|  | int Date:: GetDay( ) |
|  | { return day; |
|  | } |
|  | void Date::Display() |
|  | { cout<<year<<"-"<<month<<"-"<<day<<endl; } |
|  |  |
|  | void Date::ModifyDate(int y, int m, int d) |
|  | { year = y; |
|  | month = m; |
|  | day = d; |
|  | } |

|  |
| --- |
| 6、#include <iostream> |
|  | using namespace std; |
|  | class Date |
|  | { int year, month, day ; |
|  | public : |
|  | Date(int y=2007, int m=1, int d=1) ; |
|  | Date(const Date &date); |
|  | ~Date( ); |
|  | int GetYear( ); |
|  | int GetMonth( ); |
|  | int GetDay( ); |
|  | void Display ( ); |
|  | void ModifyDate( int, int, int); |
|  | friend bool Equal(Date& , Date& ); |
|  | }; |
|  | Date::Date(int y, int m, int d) |
|  | { year = y; |
|  | month = m; |
|  | day = d; |
|  | cout<<"Constructor called.\n"; |
|  | } |
|  |  |
|  | Date::Date( const Date &date) |
|  | { year = date.year; |
|  | month = date.month; |
|  | day = date.day; |
|  | cout<<"Copy Constructor called.\n"; |
|  | } |
|  |  |
|  | Date::~Date( ) |
|  | {cout<<"Destructor called.\n"; |
|  | } |
|  |  |
|  | int Date:: GetYear( ) |
|  | { return year; |
|  | } |
|  | int Date:: GetMonth( ) |
|  | { return month; |
|  | } |
|  | int Date:: GetDay( ) |
|  | { return day; |
|  | } |
|  | void Date::Display() |
|  | { cout<<year<<"-"<<month<<"-"<<day<<endl; } |
|  |  |
|  | void Date::ModifyDate(int y, int m, int d) |
|  | { year = y; |
|  | month = m; |
|  | day = d; |
|  | } |
|  |  |
|  | bool Equal( Date &DateA, Date &DateB) |
|  | {return (DateA.year == DateB.year && DateA.month == DateB.month && DateA.day == DateB.day); |
|  | } |

7、

|  |
| --- |
| #include <iostream> |
|  | using namespace std; |
|  | class Date |
|  | { |
|  | public: |
|  | void SetDate(int , int , int ); |
|  | private: |
|  | int year, month, day; |
|  | public: |
|  | int GetYear( ); |
|  | int GetMonth( ); |
|  | int GetDay( )； |
|  | }; |

8、

|  |
| --- |
| #include"Date1.h" |
|  | int main() |
|  | { |
|  | Date today,\*t; |
|  | today.SetDate(2006,10,17); |
|  | cout<<"One day is:"; |
|  | today.Display(); |
|  | t=&today; |
|  | (\*t).SetDate(2007,12,18); |
|  | cout<<"the other day is:"; |
|  | (\*t).Display(); |
|  | return 0; |
|  | } |

9、

|  |
| --- |
| #include"Date1.h" |
|  | int main() |
|  | { |
|  | Date today,\*t; |
|  | today.SetDate(2006,10,17); |
|  | cout<<"One day is:"; |
|  | today.Display(); |
|  | t=&today; |
|  | t->SetDate(2007,12,18); |
|  | cout<<"the other day is:"; |
|  | t->Display(); |
|  | return 0; |
|  | } |

10、

|  |
| --- |
| #include<iostream> |
|  | using namespace std; |
|  | class Date |
|  | {public: |
|  | int year, month; |
|  | protected: |
|  | int day; |
|  | public: |
|  | int GetDay( ) |
|  | { return day ; |
|  | } |
|  | void Init(int y, int m, int d) |
|  | { year = y; |
|  | month = m; |
|  | day=d; |
|  | } |
|  | int GetYear( ) |
|  | { return year ; |
|  | } |
|  | void SetYear( int y) |
|  | { year = y ; |
|  | } |
|  | }DateA, DateB; |
|  | int main( ) |
|  | { DateA.Init( 2011, 7, 23 ); |
|  | DateB.SetYear( 2012 ); |
|  | cout<<"DateA.year="<<DateA.GetYear( )<<endl; |
|  | cout<<"DateA.day="<<DateA.GetDay( ) <<endl ; |
|  | cout<<"DateA.year="<<DateA.year<<endl; |
|  | cout<<"DateB.year="<<DateB.year<<endl; |
|  | cout<<"DateB.year="<<DateB.GetYear( )<<endl ; |
|  | return 0; |
|  | } |

11

|  |  |
| --- | --- |
|  | return 0; |
|  | } |

11、

|  |
| --- |
| #include<iostream> |
|  | using namespace std; |
|  | class Date |
|  | { |
|  | private: |
|  | int year, month, day; |
|  | public: |
|  | void SetDate(int y, int m, int d); |
|  | void Display( ); |
|  | }; |
|  | void Date:: SetDate(int y, int m, int d) |
|  | { |
|  | year=y; |
|  | month=m; |
|  | day=d; |
|  | } |
|  | void Date:: SetDate(int y, int m, int d) |
|  | { |
|  | this->year=y; |
|  | this->month=m; |
|  | this->day=d; |
|  | } |
|  |  |
|  | void Date::Display( ) |
|  | { |
|  | cout<<"调用该函数的对象的this指针是"; |
|  | cout<<this<<endl; |
|  | cout<<"当前对象year成员的起始地址: "; |
|  | cout<<&year<<endl; |
|  | cout<<"当前对象month成员的起始地址: "; |
|  | cout<<&month<<endl; |
|  | cout<<"year="<<year<<" ,month="<<month<<endl; |
|  | } |
|  | int main() |
|  | { |
|  | Date DateA , DateB; |
|  | DateA.SetDate(2011,7,24); |
|  | DateB.SetDate(2010,10,1); |
|  | cout<<"DateA地址:"<<&DateA<<endl ; |
|  | DateA.Display(); |
|  | cout<<"DateB地址:"<<&DateB<<endl ; |
|  | DateB.Display(); |
|  | return 0; |
|  | } |

12

|  |
| --- |
| #include<iostream> |
|  | using namespace std; |
|  | class Date |
|  | {private: |
|  | int year, month, day; |
|  | public: |
|  | Date(int=2006 , int=10 , int=17 ); |
|  | void Display( ); |
|  | }; |
|  | Date::Date(int y, int m, int d ) |
|  | { cout<<"Executing constructor…\n"; |
|  | year = y; |
|  | month = m; |
|  | day = d; |
|  | } |
|  | void Date::Display() |
|  | { cout<< year << "-" << month << "-" << day << endl; |
|  | } |
|  | int main() |
|  | { Date today;//(2006,10,17); |
|  | cout<<"Today is:"; |
|  | today.Display(); |
|  | return 0; |
|  | } |

13

|  |
| --- |
| #include<iostream> |
|  | using namespace std; |
|  |  |
|  | class Date |
|  | { |
|  | private: |
|  | int year, month, day ; |
|  | public: |
|  | Date(int y=2000, int m=1, int d=1 ); |
|  | void Display(); |
|  | }; |
|  |  |
|  | Date::Date(int y , int m , int d) |
|  | { cout<<"Executing constructor…"<<endl; |
|  | year = y; |
|  | month = m; |
|  | day = d; |
|  | } |
|  |  |
|  | void Date::Display() |
|  | { cout<< year << "-" << month << "-" << day << endl; |
|  | } |
|  |  |
|  | int main() |
|  | { |
|  | Date initiateday; |
|  | Date newday(2006); |
|  | Date today(2006,10,17); |
|  |  |
|  | cout<<"Initiateday is:"; |
|  | initiateday.Display(); |
|  | cout<<"Newday is:"; |
|  | newday.Display(); |
|  | cout<<"Today is:"; |
|  | today.Display(); |
|  | return 0; |
|  | } |

14

|  |
| --- |
| #include "Date3.h" |
|  |  |
|  | Date f(Date Q) |
|  | { Date R(Q) ; |
|  | return Q; |
|  | } |
|  | int main() |
|  | { Date day1(2011,5,26) ; |
|  | Date day3; |
|  |  |
|  | Date day2(day1) ; |
|  | Date day4=day2; |
|  |  |
|  | day3= day2; |
|  |  |
|  | day3=f (day2); |
|  |  |
|  | day3.Display( ); |
|  |  |
|  | return 0; |
|  | } |

15

|  |
| --- |
| #include<iostream> |
|  | using namespace std; |
|  | #include"Date4.h" |
|  | int main() |
|  | { |
|  | Date today; |
|  | Date newday(2011,10,18); |
|  | newday.Display( ); |
|  | return 0; |
|  | } |

16

|  |
| --- |
| #include <string> |
|  | #include <iostream> |
|  | using namespace std; |
|  | class Student |
|  | { |
|  | char \*specialty; |
|  |  |
|  | public: |
|  | Student(char \*pSpec = 0); |
|  | Student(const Student &r); |
|  | ~Student(); |
|  | void Show(); |
|  | }; |
|  | Student::Student(char \*pSpec) |
|  | { |
|  | if (pSpec) |
|  | { |
|  | specialty = new char[strlen(pSpec) + 1]; |
|  | strcpy(specialty, pSpec); |
|  | } |
|  | else specialty = 0; |
|  | } |
|  |  |
|  | Student::Student(const Student &r) |
|  | { |
|  | if (r.specialty) |
|  | { |
|  | specialty = new char[strlen(r.specialty) + 1]; |
|  | strcpy(specialty, r.specialty); |
|  | } |
|  | else specialty = 0; |
|  | } |
|  |  |
|  | Student::~Student() |
|  | { |
|  | if (specialty) |
|  |  |
|  | delete[]specialty; |
|  |  |
|  |  |
|  | } |
|  | void Student::Show() |
|  | { |
|  | cout << "specialty=" << specialty << '\n'; |
|  | } |
|  |  |
|  | int main() |
|  | { |
|  | Student zhang("computer"); |
|  | Student wang(zhang); |
|  | zhang.Show(); |
|  | wang.Show(); |
|  | return 0; |
|  | } |

17

|  |
| --- |
| #include"Date4.h" |
|  | int main() |
|  | { Date array[3]={ Date(2011,5,1), Date(2011,10,1) }; |
|  | for(int i=0; i<3; i++) |
|  | array[i].Display( ); |
|  | return 0; |
|  | } |

18

|  |
| --- |
| #include "Date4.h" |
|  | int main() |
|  | { Date array[3]={ Date(2011,5,1), Date(2011,10,1) }; |
|  | Date \*p=array ; |
|  | for( ; p<array+3; p++) |
|  | p->Display( ); |
|  | return 0; |
|  | } |

19

|  |
| --- |
| #include "Date4.h" |
|  | int main() |
|  | { Date \*q=new Date(2010, 4, 1); |
|  | cout<<"one dynamic object is:"<<endl; |
|  | q->Display( ); |
|  | delete q; |
|  | q=new Date[ 3 ]; |
|  | q[0]=Date(2011, 5, 1); |
|  | q[1]=Date(2011, 10, 1); |
|  | for (int i=0;i<3;i++) |
|  | q[i].Display( ); |
|  | delete [ ]q; |
|  | return 0; |
|  | } |

20

|  |
| --- |
| #include "Date4.h" |
|  | Date DateA( 2010, 1, 1 ), DateB( 2010, 5, 1 ); |
|  | Date &pDate; |
|  | pDate=DateA; |
|  |  |
|  | void f( ) |
|  | { DateA. Display ( ); |
|  | DateB. Display ( ); |
|  | pDate. Display ( ); |
|  | } |
|  | int main() |
|  | { cout<<"original DateA,DateB,pDate:"<<endl; |
|  | f( ); |
|  | pDate = DateB; |
|  | cout<<"after pDate=DateB, DateA,DateB,pDate:"<<endl; |
|  | f( ); |
|  | pDate = Date(2010, 10, 1) ; |
|  | cout<<"after pDate=Date(2010, 10, 1), DateA,DateB,pDate:"<<endl; |
|  | f( ); |
|  | return 0; |
|  | } |

21

|  |
| --- |
| #include"Date5.h" |
|  | void Fun(Date DateVar) |
|  | { DateVar.ModifyDate( 2011, 11, 1 ); |
|  | DateVar. Display ( ); |
|  | } |
|  | int main() |
|  | { Date DateA; |
|  | DateA. Display ( ); |
|  | Fun(DateA); |
|  | cout<<"after calling fun() DateA:"; |
|  | DateA. Display ( ); |
|  | return 0; |
|  | } |

22

|  |
| --- |
| #include"Date5.h" |
|  | void Fun(Date \*pDate) |
|  | { pDate->ModifyDate(2011, 11, 1); |
|  | pDate-> Display ( ); |
|  | } |
|  | int main() |
|  | { Date DateA(2011); |
|  | DateA. Display ( ); |
|  | Fun (&DateA) ; |
|  | cout<<"after calling fun( ) DateA: "; |
|  | DateA. Display ( ); |
|  | return 0; |
|  | } |

23

|  |
| --- |
| #include"Date5.h" |
|  | void Fun(Date &pDate) |
|  | { pDate.ModifyDate(2011, 11, 1); |
|  | pDate.Display( ); |
|  | } |
|  | int main() |
|  | { |
|  | Date DateA(2010); |
|  | DateA.Display( ); |
|  | Fun (DateA) ; |
|  | cout<<"after calling fun() DateA: "; |
|  | DateA.Display( ); |
|  | return 0; |
|  | } |

24

|  |
| --- |
| #include"Date5.h" |
|  |  |
|  | Date &Fun(Date &pDate) |
|  | { |
|  | pDate.ModifyDate( 2012, 5 ,1 ); |
|  | cout<<"reference pDate:\n"; |
|  | pDate.Display( ); |
|  | return pDate; |
|  | } |
|  | int main() |
|  | { Date DateA(2011),tDate; |
|  | cout<<"Before right Fun, DataA:\n"; |
|  | DateA.Display(); |
|  | cout<<"Before right Fun, tDate:\n"; |
|  | tDate.Display(); |
|  | tDate=Fun(DateA); |
|  | cout<<"After right Fun, DateA:\n"; |
|  | DateA.Display( ); |
|  | cout<<"After right Fun, tDate:\n"; |
|  | tDate.Display( ); |
|  | Fun(DateA)=Date(1998,10,5); |
|  | cout<<"After left Fun, DateA:\n"; |
|  | DateA.Display(); |
|  | cout<<"After left Fun, tDate:\n"; |
|  | tDate.Display(); |
|  | return 0; |
|  | } |

25

|  |
| --- |
| #include"Date5.h" |
|  | bool Equal( Date &DateA, Date &DateB) |
|  | { return ( DateA.GetYear( ) == DateB.GetYear( ) |
|  | &&DateA.GetMonth( ) == DateB.GetMonth( ) |
|  | &&DateA.GetDay( ) == DateB.GetDay( ) ); |
|  | } |
|  | int main() |
|  | { |
|  | Date YourBirthday( 1990,10,17 ); |
|  | Date MyBirthday( 1990,10,17 ); |
|  | if (Equal( YourBirthday, MyBirthday) ) |
|  | cout<<"Birthday is the same!\n"; |
|  | else |
|  | cout<<"Birthday is not the same!\n"; |
|  | return 0; |
|  | } |

26

|  |
| --- |
| #include"Date6.h" |
|  | int main() |
|  | { Date YourBirthday(1990 , 10 , 17); |
|  | Date MyBirthday(1990 , 10 , 17); |
|  | if ( Equal(YourBirthday , MyBirthday) ) |
|  | cout<<"Birthday is the same!\n"; |
|  | else |
|  | cout<<"Birthday is not the same!\n"; |
|  | return 0; |
|  | } |

27

|  |
| --- |
| #include<iostream> |
|  | #include<string> |
|  | using namespace std; |
|  | class Student; |
|  | class Date |
|  | { int year, month, day ; |
|  | public : |
|  | Date( int y=2007 , int m=1 , int d=1 ) ; |
|  | void Display (const Student &stu ); |
|  | }; |
|  | class Student |
|  | {private: |
|  | char \*specialty; |
|  | public: |
|  | Student(char \*pSpec); |
|  | ~Student(); |
|  | friend void Date::Display (const Student &); |
|  | }; |
|  | Date::Date(int y,int m,int d) |
|  | { year = y; |
|  | month = m; |
|  | day = d; |
|  | cout<<"Constructor called.\n"; |
|  | } |
|  | void Date:: Display (const Student &stu) |
|  | { cout<< stu.specialty <<endl; |
|  | cout<< year << "-" << month << "-" << day << endl; |
|  | } |
|  |  |
|  | Student::Student(char \*pSpec) |
|  | { if ( pSpec ) |
|  | { specialty = new char[strlen( pSpec ) + 1]; |
|  | strcpy( specialty, pSpec ); |
|  | } |
|  | else specialty = 0; |
|  | } |
|  | Student::~Student( ) |
|  | { if ( specialty ) |
|  | delete [ ]specialty; |
|  | } |
|  |  |
|  | int main( ) |
|  | { Student zhang("computer"); |
|  | Date Birthday( 1990,10,17 ); |
|  | Birthday. Display ( zhang ); |
|  | return 0; |
|  | } |

28.

|  |
| --- |
| #include<iostream> |
|  | #include<string> |
|  | using namespace std; |
|  | class Student; |
|  | class Date |
|  | { int year, month, day; |
|  | public: |
|  | Date(int , int , int ); |
|  | void Display (); |
|  | friend Student; |
|  | }; |
|  | class Student |
|  | {private: |
|  | char \*specialty; |
|  | char \*name; |
|  | public: |
|  | Student(char \*pn, char \*pSpec); |
|  | ~Student(); |
|  | void Display (Date &); |
|  | }; |
|  | Date::Date(int y,int m,int d):year(y),month(m),day(d) |
|  | { } |
|  | void Date:: Display ( ) |
|  | { cout<<year<<"-"<<month<<"-"<<day<<endl; |
|  | } |
|  | Student::Student(char \*pn, char \*pSpec) |
|  | { if(pSpec) |
|  | { specialty=new char[ strlen( pSpec ) + 1]; |
|  | strcpy( specialty, pSpec ); |
|  | } |
|  | else specialty = 0; |
|  | if ( pn ) |
|  | { name = new char[ strlen( pn ) + 1]; |
|  | strcpy(name, pn); |
|  | } |
|  | else name = 0; |
|  | } |
|  | Student::~Student() |
|  | { if ( specialty ) delete [ ]specialty; |
|  | if( name ) delete [ ]name; |
|  | } |
|  | void Student:: Display ( Date& obj ) |
|  | { cout<<name<<endl; |
|  | cout<<specialty<<endl; |
|  | obj. Display () ; |
|  | cout<<"Date's another format is:"; |
|  | cout<<obj.month<<"\_"<<obj.day<<"\_"<<obj.year<<endl; |
|  |  |
|  | } |
|  | int main() |
|  | { Student boy( "zhang", "computer" ); |
|  | Date birthday(1990,10,17); |
|  | boy. Display ( birthday ) ; |
|  | return 0; |
|  | } |