```
// W9_2.cpp : Defines the entry point for the console application.
//          Wielodziedziczenie:      B1         B2
//                                   *          *
//                                   ************
//                                         *
//                                         D



#include "stdafx.h"
#include <iostream>
using namespace std;

//declaration of class base
class base1
{
      int ib;
public:
      int jb;
      base1(int ii, int jj) { ib = ii; jb = jj; cout << " konstruktor base1\n"; }
      ~base1() { cout << " destruktor base1\n"; }
      int get_ib() { return ib; }
      void show() { cout << "base1 : ib = " << ib << " jb = " << jb << endl; }
};


class base2
{
      int ib;
public:
      int jb;
      base2(int ii, int jj) { ib = ii; jb = jj; cout << " konstruktor base2\n"; }
      ~base2() { cout << " destruktor base2\n"; }
      int get_ib() { return ib; }
      void show() { cout << "base2 : ib = " << ib << " jb = " << jb << endl; }
};

//declaration of class derived
class derived : public base1, public base2
{
      int id;
public:
```

```
      int jd;
      derived(int ii, int jj, int i_bas_1, int j_bas_1, int i_bas_2, int j_bas_2);
      ~derived() { cout << " destruktor derived\n"; }
      int get_id() { return id; }
      void show() {
            cout << "derived : id = " << id << " jd = " << jd << endl;
            base1::show();
            base2::show();
      }
};

//implementation of class derived_1

derived::derived(int ii, int jj, int i_bas_1, int j_bas_1, int i_bas_2, int j_bas_2)
                                                                  : base1(i_bas_1, j_bas_1),
                                                                    base2(i_bas_2, j_bas_2)
/*=============================================================================
Konstruktor klasy derived
=============================================================================*/
{
      id = ii;
      jd = jj;
      cout << "konstruktor derived\n";
}




int _tmain(int argc, _TCHAR* argv[])
{
      derived d(50, 60, 3, 4, 5, 6);
      d.show();

      system("pause");

      return 0;
}
```