

```
// W4_0.cpp : Defines the entry point for the console application.  
//          Wprowadzenie w dziedziczenie klas
```

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

```
//deklaracje klasy bazowej
```

```
class B  
{  
    int i;  
public:  
    int k;  
    int get_i();  
    void set_i(int n);  
};
```

```
//implementacje klasy bazowej
```

```
int B::get_i()  
{  
    return i;  
}
```

```
void B::set_i(int n)  
{  
    i = n;  
}
```

```
//deklaracje klasy pochodnej
```

```
class D : public B  
{  
    int j;  
public:  
    int get_j();  
    void set_j(int n);  
    int mul();  
};
```

```
//implementacje klasy pochodnej
```

```
int D::get_j()  
{
```

```

        return j;
    }

    void D::set_j(int n)
    {
        j = n;
    }

    int D::mul()
    /*=====
    Zwraca val = i*k+j*k
        i - prywatna składowa klasy bazowej
        k - publiczna składowa klasy bazowej
        j - prywatna składowa klasy pochodnej
    =====*/
    {
        int val;
        //val = i*j;    //blad! klasa pochodna nie ma dostępu bezpośredniego
                        //do prywatnej składowej klasy bazowej
        val = j*get_i(); // OK! get_i() - publiczna składowa klasy bazowej
        val += j*k;     //OK! klasa pochodna ma bezpośredni dostęp do
                        //składowych publicznych klasy bazowej

        return val;
    }

    int _tmain(int argc, _TCHAR* argv[])
    {
        D ob;          //deklaryjemy obiekt klasy pochodnej
        ob.set_i(10);   //obiekt klasy pochodnej dziedzichi metode
                        //klasy bazowej
        ob.set_j(20);
        ob.k = 30;      //obiekt klasy pochodnej dziedzichi składowa
                        //klasy bazowej
        //Ustaw tu punkt przerywania i zobacz stan zmiennych obiektu ob
        int result = ob.mul();

        cout << "stan zmiennych obiektu ob:" << endl;
        cout << " i = " << ob.get_i() << endl;
        cout << " j = " << ob.get_j() << endl;
        cout << " k = " << ob.k << endl;
    }
}

```

```
cout << " wynik: " << result << endl;  
system("pause");  
return 0;  
}
```