

integer.cpp

```
#include <iostream>
using namespace std;

class integer {    // int - Variable in class verpacken
//    private:    // private ist default
    int i; // this->i bzw. (*this).i
public:
    integer(int i=0):i(i){ // Konstruktor und Defaultkonstruktor
        cout<<"integer-Objekt i = "<<this->i<<endl;
    }

    int get(){ return i; }

    void set(int i=0){ this->i = i; }

    static integer add(integer i1, integer i2){ // Wertkopien von i1 und i2
// alternativ und explizit: Konstruktor-Aufruf:
//    return integer(i1.i + i2.i);
// Umwandlung int nach integer, Aufruf Konstruktor implizit
        return i1.get()+i2.get();
    }
};

int main(){
    auto k = 0;          // C++11: da 0 vom Typ int ist auch k vom Typ int
    decltype(k) j = 5; // C++11: da k vom Typ int ist auch j vom Typ int

    integer i0(5), i1=4;    // 2 (alternative) Initialisierungen von Objekten
    cout<<"i0.i = "<<i0.get()<<endl;
    cout<<"i0.i + i0.i = "<<integer::add(i0, i0).get()<<endl; //static-Methode add
    integer i3 = integer::add(i0, i0);    // Initialisierung von i3
    cout<<"i3.i = "<<i3.get()<<endl;
    i0.set(22);
    cout<<"i0.i = "<<i0.get()<<endl;
    cin.get();
}

/*
integer-Objekt i = 5
integer-Objekt i = 4
i0.i = 5
integer-Objekt i = 10
i0.i + i0.i = 10
integer-Objekt i = 10
i3.i = 10
i0.i = 22
*/
```