

Präfix- und Postfix-Operatoren: Inkrement.cpp

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

class day {
public:  enum Tag { son, mon, die, mit, don, fre, sam };
private: Tag t;                                //Datenmember
public:
    day(day::Tag t = sam):t(t){}                //Konstruktor

    //Praefix-Operator ++ mit spezifischer Signatur:
    day &operator++(){
        t = (t == sam ) ? son : Tag(t+1);
        return *this;
    }

    //Postfix-Operator ++ mit spezifischer Signatur:
    day operator++(int i){                        //i als dummy
        day dd(*this);
        t = (t == sam ) ? son : Tag(t+1);
        return dd;
    }

    Tag get(){ return t; }
};

//Ueberladener Operator << fuer Objekte vom Typ day
ostream &operator<<(ostream &cout, day &d){
    if(&d)switch(d.get()){
        case day::mon: cout<<"mon\n"; break;
        case day::die: cout<<"die\n"; break;
        case day::mit: cout<<"mit\n"; break;
        case day::don: cout<<"don\n"; break;
        case day::fre: cout<<"fre\n"; break;
        case day::sam: cout<<"sam\n"; break;
        case day::son: cout<<"son\n"; break;
        default: cout<<"undefined\n"; break;
    }
    else cout<<"d not defined\n";
    return cout;
}

void main(){
    day d;                // optional:    day d(day::sam)
    cout<<d<<endl;        // Ausgabe:      sam
    ++d;                  // Wahl:         d.operator++();
    cout<<d<<endl;        // Ausgabe:      son
    cout<<d++<<endl;      // Wahl,Ausg.:  d.operator++(0),son
    cout<<d<<endl;        // Ausgabe:      mon
    cin.get();
}
```