```
/**
 * @file        MyDouble.cpp
 * @synopsis    MyDouble-Klasse Definition
 * @author      Jan Tammen (FH Konstanz), <jan.tammen@fh-konstanz.de>
 * @date        2005-03-30
 */

#include "MyDouble.h"

// {{{ Konstruktoren, Destruktor
// ---------------------------------------------------------------- //

/// Default-Konstruktor
MyDouble::MyDouble(void)
{
    this->mNumber = 0.0;
}

MyDouble::MyDouble (double number)
{
    this->mNumber = number;
}

/// Destruktor
MyDouble::~MyDouble ()
{
    /// nothing to do.
}

// ---------------------------------------------------------------- //
// }}}

// {{{ Operatoren
// ---------------------------------------------------------------- //

// {{{ Operatoren: Zuweisung
MyDouble& MyDouble::operator= (const MyDouble& z)
{
    if (&z != this)
    {
        this->mNumber = z.mNumber;
    }

    return (*this);
}
// }}}

// {{{ Operatoren: Addition
MyDouble MyDouble::operator+ (const MyDouble& z) const
{
    MyDouble* s = new MyDouble();
    s->setNumber(this->mNumber + z.mNumber);
    return *s;
}

MyDouble& MyDouble::operator+= (const MyDouble& z)
{
    return (*this) = (*this) + z;
}
// }}}

// {{{ Operatoren: Subtraktion
MyDouble MyDouble::operator- (const MyDouble& z) const
{
    MyDouble* d = new MyDouble();
    d->setNumber(this->mNumber - z.mNumber);
    return *d;
}
```



```
MyDouble& MyDouble::operator-= (const MyDouble& z)
{
    return (*this) = (*this) - z;
}

MyDouble MyDouble::operator- (void) const
{
    MyDouble* res = new MyDouble();
    res->mNumber = this->mNumber*(-1.0);
    return *res;
}
// }}}

// {{{ Operatoren: Multiplikation
MyDouble MyDouble::operator* (const MyDouble& z) const
{
    MyDouble* p = new MyDouble();
    p->setNumber(this->mNumber * z.mNumber);
    return *p;
}

MyDouble& MyDouble::operator*= (const MyDouble& z)
{
    return (*this) = (*this) * z;
}
// }}}

// {{{ Operatoren: Ausgabe
std::ostream& operator<< (std::ostream& s, const MyDouble& z)
{
    s << z.mNumber;
    return s;
}
// }}}

// ---------------------------------------------------------------- //
// }}}

/* vim: set expandtab tabstop=4 shiftwidth=4 softtabstop=4 foldmethod=marker: */
```