

## **//funktion.h**

```
#ifndef FUNKTION_H
#define FUNKTION_H

class ostream;

class Funktion {
public:
    virtual double operator() (double x) const;
    virtual void ausgebenAuf(ostream&) const;
    virtual Funktion* kopie() const = 0;
};

ostream& operator<< (ostream&, const Funktion&);

#endif
```

## **//polynomfkt.h**

```
#ifndef POLYNOMFKT_H
#define POLYNOMFKT_H

class ostream;
#include <funktion.h>
#include <polynom.h>

class PolynomFunktion : public Funktion {
private:
    Polynom mPolynom;
public:
    PolynomFunktion(const Polynom& p);
    virtual double operator() (double x) const;
    virtual void ausgebenAuf(ostream&) const;
    virtual Funktion* kopie() const;
};

#endif
```

## **//dekorierer.h**

```
#ifndef DEKORIERER_H
#define DEKORIERER_H

class ostream;
#include <funktion.h>

class Dekorierer : public Funktion {
private:
    Funktion* mFunktion;
    Dekorierer(const Funktion* const);
//    const Dekorierer& operator= (const Dekorierer&);
public:
    Dekorierer(const Funktion&);
    Dekorierer(const Dekorierer&);
    const Dekorierer& operator= (const Dekorierer&);
    virtual double operator() (double x) const;
    virtual void ausgebenAuf(ostream&) const;
    virtual Funktion* kopie() const;
    virtual ~Dekorierer();
};

#endif
```

## //betrag.h

```
#ifndef BETRAG_H
#define BETRAG_H

class ostream;
class funktion;
#include <dekorierer.h>
#include <polynomfkt.h>

class Betrag : public Dekorierer {
public:
    Betrag(const Funktion& = PolynomFunktion(x));
    virtual double operator() (double x) const;
    virtual void ausgebenAuf(ostream&) const;
    virtual Funktion* kopie() const;
};

#endif
```

## //kehrwert.h

```
#ifndef KEHRWERT_H
#define KEHRWERT_H

class ostream;
class funktion;
#include <dekorierer.h>

class Kehrwert : public Dekorierer {
public:
    Kehrwert(const Funktion&);
    virtual double operator() (double x) const;
    virtual void ausgebenAuf(ostream&) const;
    virtual Funktion* kopie() const;
};

#endif
```

## //spiegelhor.h

```
#ifndef SPIEGELHOR_H
#define SPIEGELHOR_H

class ostream;
class funktion;
#include <dekorierer.h>

class SpiegelungHorizontal : public Dekorierer {
private:
    double mVersatz;
public:
    SpiegelungHorizontal(const Funktion&, double versatz = 0.);
    virtual double operator() (double x) const;
    virtual void ausgebenAuf(ostream&) const;
    virtual Funktion* kopie() const;
};

#endif
```