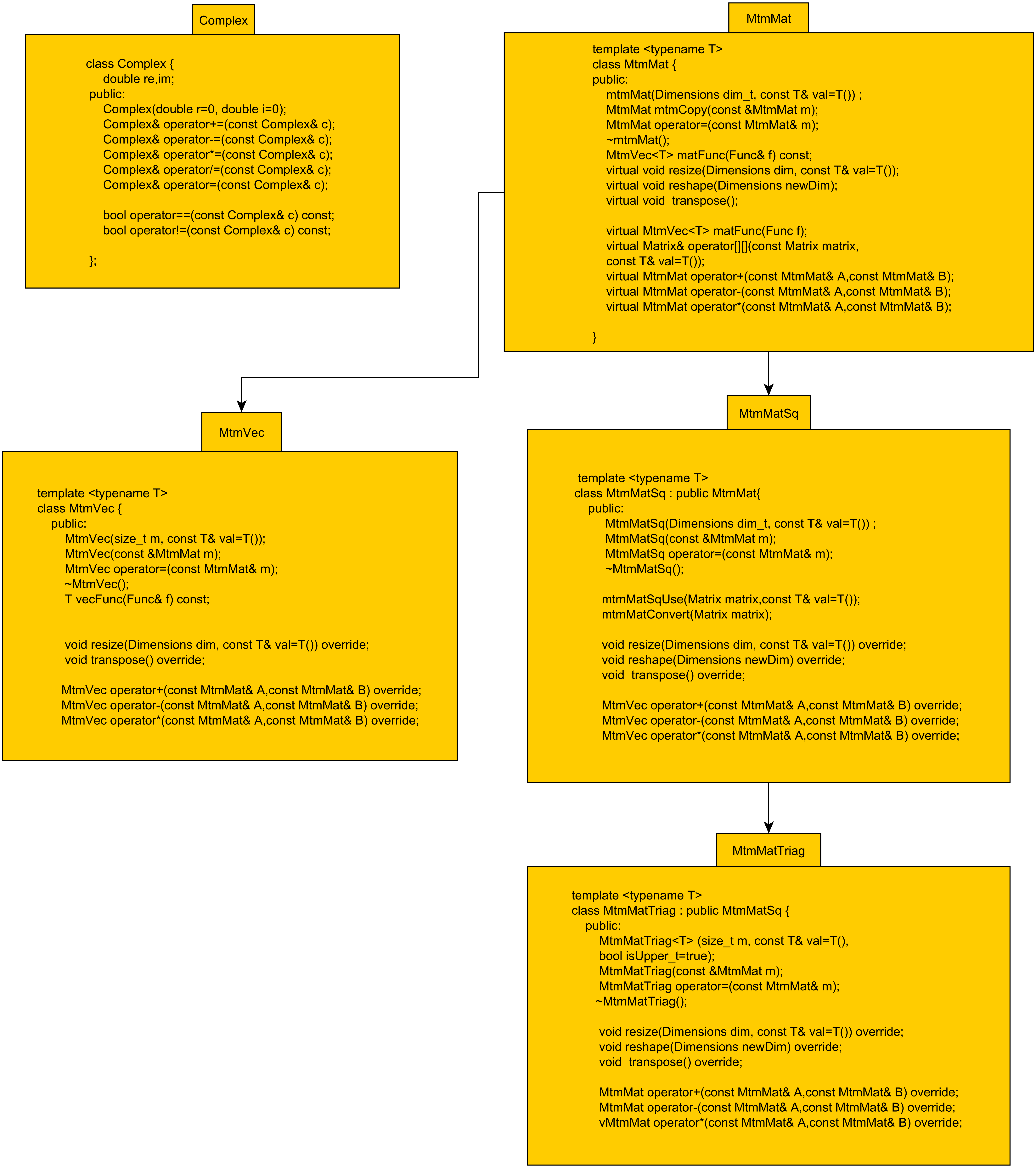

Complex

```
class Complex {
    double re,im;
 public:
    Complex(double r=0, double i=0);
    Complex& operator+=(const Complex& c);
    Complex& operator-=(const Complex& c);
    Complex& operator*=(const Complex& c);
    Complex& operator/=(const Complex& c);
    Complex& operator=(const Complex& c);

    bool operator==(const Complex& c) const;
    bool operator!=(const Complex& c) const;

 };
```

MtmMat

```
template <typename T>
class MtmMat {
public:
    mtmMat(Dimensions dim_t, const T& val=T()) ;
    MtmMat mtmCopy(const &MtmMat m);
    MtmMat operator=(const MtmMat& m);
    ~mtmMat();
    MtmVec<T> matFunc(Func& f) const;
    virtual void resize(Dimensions dim, const T& val=T());
    virtual void reshape(Dimensions newDim);
    virtual void  transpose();

    virtual MtmVec<T> matFunc(Func f);
    virtual Matrix& operator[][](const Matrix matrix,
    const T& val=T());
    virtual MtmMat operator+(const MtmMat& A,const MtmMat& B);
    virtual MtmMat operator-(const MtmMat& A,const MtmMat& B);
    virtual MtmMat operator*(const MtmMat& A,const MtmMat& B);

}
```

MtmVec

```
template <typename T>
class MtmVec {
    public:
        MtmVec(size_t m, const T& val=T());
        MtmVec(const &MtmMat m);
        MtmVec operator=(const MtmMat& m);
        ~MtmVec();
        T vecFunc(Func& f) const;


        void resize(Dimensions dim, const T& val=T()) override;
        void transpose() override;

       MtmVec operator+(const MtmMat& A,const MtmMat& B) override;
       MtmVec operator-(const MtmMat& A,const MtmMat& B) override;
       MtmVec operator*(const MtmMat& A,const MtmMat& B) override;
```

MtmMatSq

```
 template <typename T>
class MtmMatSq : public MtmMat{
    public:
         MtmMatSq(Dimensions dim_t, const T& val=T()) ;
         MtmMatSq(const &MtmMat m);
         MtmMatSq operator=(const MtmMat& m);
         ~MtmMatSq();

        mtmMatSqUse(Matrix matrix,const T& val=T());
        mtmMatConvert(Matrix matrix);

        void resize(Dimensions dim, const T& val=T()) override;
        void reshape(Dimensions newDim) override;
        void  transpose() override;

        MtmVec operator+(const MtmMat& A,const MtmMat& B) override;
        MtmVec operator-(const MtmMat& A,const MtmMat& B) override;
        MtmVec operator*(const MtmMat& A,const MtmMat& B) override;
```

MtmMatTriag

```
template <typename T>
class MtmMatTriag : public MtmMatSq {
    public:
        MtmMatTriag<T> (size_t m, const T& val=T(),
        bool isUpper_t=true);
        MtmMatTriag(const &MtmMat m);
        MtmMatTriag operator=(const MtmMat& m);
       ~MtmMatTriag();

        void resize(Dimensions dim, const T& val=T()) override;
        void reshape(Dimensions newDim) override;
        void  transpose() override;

        MtmMat operator+(const MtmMat& A,const MtmMat& B) override;
        MtmMat operator-(const MtmMat& A,const MtmMat& B) override;
        vMtmMat operator*(const MtmMat& A,const MtmMat& B) override;
```