

## STREAMS

Striim – abstraktsioon viitamaks andmevoole allikast (sisendist) neeluni (väljundini).

Sisendist *võtame, eraldame* baite

Väljundisse *saadame, salvestame, paneme vahele* baite

class **streambuf** – puhverdamine elementaarse formateerimisega

class **ios** – sisend/väljund meetodid; sisaldab viita ( streambuf \* )

class **istream** : virtual public ios;

class **ostream** : virtual public ios;

class **iostream** : public istream, public ostream;

class **istream\_withassign** : public istream

{

public:

istream\_withassign( );

istream& operator = ( istream &);

istream& operator = ( streambuf \*);

};

class **ostream\_withassign** : public ostream;

// = nagu eespool

class **iostream\_withassign** : public iostream;

// = nagu eespool

Failis *iostream.h* on defineeritud objektid:

```
extern istream_withassign cin;  
extern ostream_withassign cout;  
extern ostream_withassign cerr;  
extern ostream_withassign clog;
```

Operaatorid >> ja << on üledefineeritud:

```
int k = 3, m;
```

```
char c = 'X';
```

```
cout << k; cin >> m;
```

Meetodid:

```
cout.put( c );                // sama mis   cout << c
cout.write( (char *) &m, sizeof( m ) ); // baitide väljastamine
cin.get( c );                 // sama mis   cin >> c
cin.read( (char *) &k, sizeof( k ) );  // baitide lugemine
cin.putback( c );             // sümboli “tagasipanemine” sisendvoogu
```

## FAILINDUS

```
class filebuf : public streambuf;
```

```
class ifstream : public istream;
class ofstream : public ostream;
class fstream : public iostream;
```

Näide – faili kopeerimine:

```
#include <fstream.h>
...
char c;
ifstream sisend( “vana.doc” );
if( !sisend ) { errmsg( “Ei saa avada sisendfaili” ); exit( 0 ); };
ofstream valjund( “uus.doc” );
if( !valjund ) { errmsg( “Ei saa avada väljundfaili” ); exit( 0 ); };
while ( sisend.get( c ) && valjund )
    valjund.put( c );
sisend.close();
valjund.close();
```

Faili ei pea avama striimi loomisel:

```
ofstream f;
...
f.open( “esimene.txt” )
...
f.close();
f.open( “teine.txt” );
...
```

Näide – fikseeritud pikkusega kirjete fail

```
class KirjeteFail
{
protected:
    fstream f;
    int kirje_pikkus;
    streampos get_pos, put_pos;          // integer
public:
    KirjeteFail ( char *fnimi, int kp )
    {
        f.open( fnimi, ios::in | ios::out );
        kirje_pikkus = kp;
        get_pos = f.tellg(); put_pos = f.tellp();
    }
    void lugeda( int kirjenr, void *kirje )
    {
        f.seekg( kirjenr * kirje_pikkus, ios::beg );    // ios::cur
        f.read( (char *) kirje, kirje_pikkus );
        get_pos = f.tellg();
    }
    void kirjutada( int kirjenr, void *kirje )
    {
        f.seekp( kirjenr * kirje_pikkus, ios::beg );
        f.write( (char *) kirje, kirje_pikkus );
        put_pos = f.tellp();
    }
    void lisada( void *kirje )
    {
        f.seekp( 0, ios::end );
        f.write( (char *) kirje, kirje_pikkus );
        put_pos = f.tellp();
    }
}
```

Vigade töötlemine:

```
if( !f.good() ) ... // veatöötlustele
```

```
if( f.eof() ) ... // EOF töötlus
```