

```

/**
 * @file      SourceCompressor.h
 * @synopsis   Klassen-Deklaration SourceCompressor
 * @author    Jan Tammen (FH Konstanz), <jan.tammen@fh-konstanz.de>
 * @date      2005-04-14
 */

#ifndef SOURCECOMPRESSOR_H
#define SOURCECOMPRESSOR_H

#include <string>
#include <iostream>
#include <fstream>
#include <vector>
#include <map>

using namespace std;

class SourceCompressor
{
public:
    /// Konstruktor: Quelldateistrom oeffnen
    SourceCompressor (string src);
    SourceCompressor (void);
    ~SourceCompressor (void);

    /// Start des Automaten
    void compress ();
    /// Soll die Ausgabe auch auf cout erscheinen?
    void setShowOnScreen (bool b) { mShowOnScreen = b; }

private:
    bool      mShowOnScreen;          /// Ausgabe auch auf dem Schirm?
    string     mSrcFilename;          /// Quelldateiname
    string     mDestFilename;         /// Zieldateiname
    ifstream  mSrcFilestream;         /// Zieldateistrom
    ofstream  mDestFilestream;        /// Quelldateistrom
    string     mOutBuffer;            /// Ausgabepuffer

    /// Zustaende & Aktionen des Automaten
    enum AutData {UNDEF,
        STATE_BASE, STATE_ZK, STATE_ZKE, STATE_ZKW,
        STATE_KO__QM, STATE_KO, STATE_AST__KO, STATE_KOE__QM,
        STATE_ZAHL, STATE_SZF, STATE_NAME, STATE_STRING,
        ACTION_STO, ACTION_OUT, ACTION_OUTSTO,
        ACTION_STOOUT, ACTION_NOOP,
        ACTION_CLEAR};

    /// Zeichengruppen
    enum CharGroup {CGROUP_BU, CGROUP_ZI, CGROUP_SZ, CGROUP_WZ,
        CGROUP_APO, CGROUP_QUOT, CGROUP_SLASH,
        CGROUP_AST, CGROUP_BSLASH, CGROUP_SONST,
        CGROUP_EOL, CGROUP_EOF};

    /// Steuermatrix
    vector< vector< vector<AutData> > > controlMatrix;

    /// Steuermatrix fuellen
    void buildControlMatrix (void);
    /// Zeichengruppe bestimmen
    CharGroup findCharGroup (char c) const;
    /// Aktion (Ausgabe auf Datei) durchfuehren
    void performAction (const int action, char& c);
};

#endif

```