CCoverageMatrix.cpp

Fa felállítása, melyik melyikből öröklődik.

**data/CoverageMatrix.h**

**data/CBitMatrix.h**

**data/CIDManager.h**

**//exception/CException.h**

**CIDManager.h-->io/CSoDAio.h, interface/IIDManager.h**

io/csodaio.h -->io/cbinaryio.h, data/Sodalibdefs.h

io/cbinaryyio.h -->data/sodalibdefs.h PIPA

interface/IIDManager -->io/csodaio -->megvan

**CBitMatrix.h -->interface/IBitMatrix.h, data/CBitlist.h**

interface/IBitMatrix.h -->io/CSODAIO.h, io/CbBitreader.h, io/CBitwriter.h, IBitList.h

io/CSODAIO.h--> megvan

io/CBitReader.h-->io/CBinaryio.h-->megvan

io/CbitWriter.h-->io/CBinaryio.h-->megvan

IBitList.h-->CSoDAIO(megvan), CBitReader(megvan), CBitWriter(megvan), Sodalibdefs(megvan) PIPA

data/CbitList.h---> interface/IBitlist.h-->megvan

**CoverageMatrix.h-->** interface/IBitMatrix.h, interface/IIDManager.h

interface/IBitMatrix.h-->megvan

interface/IIDManager.h-->megvan

///////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

**CResultsMatrix.h-->used functions() and etc:**

CCoverageMatrix();

~CCoverageMatrix();

StringVector getTestcases(const String& codeElementName) const;

StringVector getCodeElements(const String& testcaseName) const;

const IndexType getNumOfTestcases() const;

const IndexType getNumOfCodeElements() const;

bool getRelation(const String& testcaseName, const String& codeElementName) const;

void setRelation(const String& testcaseName, const String& codeElementName, const bool isCovered=true);

void addOrSetRelation(const String& testcaseName, const String& codeElementName, const bool isCovered=true);

void refitMatrixSize();

void addTestcaseName(const String&);

void addCodeElementName(const String&);

void save(const String& filename) const;

void load(const String& filename);

IIDManager\* m\_testcases;

IIDManager\* m\_codeElements;

IBitMatrix\* m\_data;

bool m\_createTestcases;

bool m\_createCodeElements;

bool m\_createBitMatrix;

CCoverageMatrix(const CCoverageMatrix&);

CCoverageMatrix& operator=(const CCoverageMatrix&);

**CCoverageMatrix.cpp**

**-header alapján itt is kitörölni ami nem kell, ami megmaradt azokat a megvalósításokat kellett megvizsgálni, hogy melyik rész, honnan jön, melyik headerből, odaírni melléjük is kommentbe.**

**így egyszerűen átláthatóvá válik a lebontott projekt. Ez alapján:**

CCoverageMatrix.cpp-->vectors, etc. -->SoDALibDefs.h

CCoverageMatrix.cpp-->new CIDManager()-->CIDManager.h

CCoverageMatrix.cpp-->size()-->IIDManager.h, CBitList.h, CIDManager.h

CCoverageMatrix.cpp-->getIDList()-->IIDManager.h

CCoverageMatrix.cpp-->add()-->IIDManager.h

CCoverageMatrix.cpp-->containsValue()-->IIDManager.h

CCoverageMatrix.cpp-->new CBitMatrix()-->CBitMatrix.h

CCoverageMatrix.cpp-->all functions-->CCoverageMatrix.h

CCoverageMatrix.cpp-->push\_back()-->IBitList.h

CCoverageMatrix.cpp-->get()-->IBitMatrix.h

CCoverageMatrix.cpp-->set()-->IBitMatrix.h

CCoverageMatrix.cpp-->resize()—IbitMatrix.h

CCoverageMatrix.cpp-->getNumOfRows()-->IBitMatrix.h

CCoverageMatrix.cpp-->getNumOfCols()-->IBitMatrix.h

**Ezekbe a headerekbe belelépve, kikommentelni azokat amik nem kellenek ezekhez a functionokhoz (lehet hogy később mégis kelleni fog így elég csak a kommentelést visszavonni), csökkentve így a headerek méretét, ezáltal az interfaceba is kevesebb dolgot kell felülvizsgálni.**

**Utolsó lépés: felesleges #include-okat kitörölni/átrendezni a használandó headerekből.**

Mindegyik fájlnak szinte ugyanaz a gyökérpontja: CSoDAio.h, annak a CBinaryIO.h, annak pedig SoDALibDefs.h

**A CSoDAio. , CBinaryIO.h elméletileg nem kell.**

**A SoDALibDefs mindenképp**