CResultMatrix.cpp

Fa felállítása, melyik melyikből öröklődik.

**//CException.h**

**CIDManager.h**-->io/csodaio.h, inteface/IIDManager.h

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

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

interface/IIDManager -->io/csodaio --> PIPA

**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--> PIPA(cidmanager-ben megcsinálva)

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

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

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

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

**CResultmatrix.h**-->IO/CSodaIO.h, interface/IBitMatrix.h, interface/IIDManager.h,

**CRevision.h**

IO/CSodaIO.h -->PIPA

interface/IBitMatrix.h --> PIPA

interface/IIDManager.h-->CSODAIO.h--PIPA

CRevision.h-->io/CSODAIO.h (PIPA), data/Sodalibdefs.h(PIPA)

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

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

enum TestResultType { trtNotExecuted=0, trtFailed=2, trtPassed=3 };

CResultsMatrix()

~CResultsMatrix();

const CRevision<IndexType>& getRevisions() const;

IntVector getRevisionNumbers() const;

const IndexType getNumOfTestcases() const;

const IndexType getNumOfRevisions() const;

TestResultType getResult(const int revision, const String& testcaseName) const;

bool isExecuted(const int revision, const String& testcaseName) const;

bool isPassed(const int revision, const String& testcaseName) const;

void setResult(const int revision, const String& testcaseName, const TestResultType result);

void addOrSetResult(const int revision, const String& testcaseName, const TestResultType result);

void refitMatrixSize();

void addRevisionNumber(const int);

void addTestcaseName(const String&);

void save(const String& filename) const;

void load(const String& filename);

IIDManager\* m\_testcases;

CRevision<IndexType>\* m\_revisions;

IBitMatrix\* m\_exec;

IBitMatrix\* m\_pass;

bool m\_createTestcases;

bool m\_createRevisions;

bool m\_createExecutionBitMatrix;

bool m\_createPassedBitMatrix;

**CResultsMatrix.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:**

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

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

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

CResultsMatrix.cpp-->all functions-->CResultsMatrix.h

CResultsMatrix.cpp-->size()-->IIDManager.h

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

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

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

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

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

CResultsMatrix.cpp-->resize() -->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 valamivel 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**