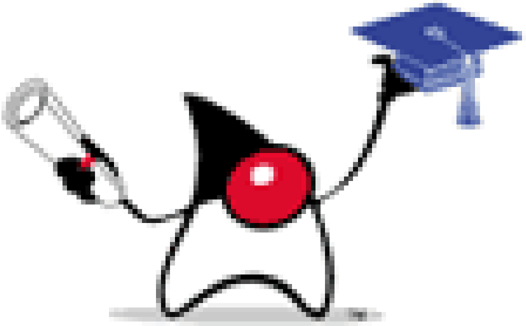
Athens University of Economics & Business
Department of Management Science and Technology

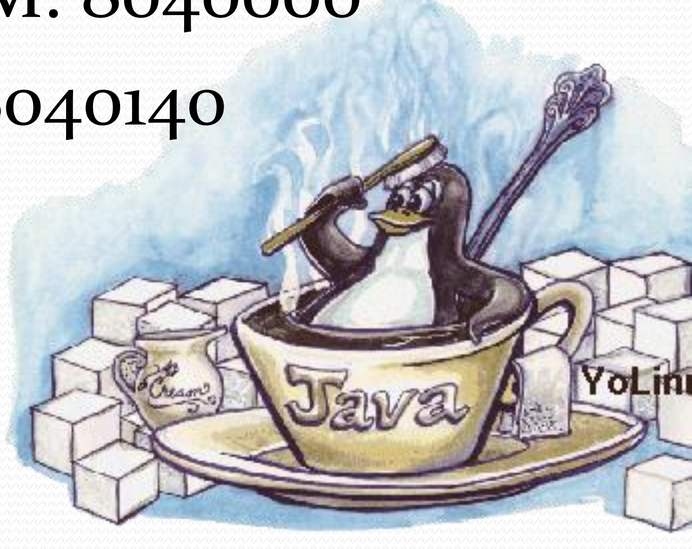
Advanced Topics in Software Engineering

Project Team – jLab

Φοιτητές:

- Παναγιώτης Αδαμόπουλος, Α.Μ. 8040000
- Γεωργία – Βίλμα Τόδρη, Α.Μ. 8040140





jLab

- The jLab project aims to provide a Matlab/Scilab environment
  - with a scripting interpreter implemented in Java
  - with the potential of linking dynamically Java numerical computing code.
- The system will perform very efficiently since the Java class code executes very fast.
- Moreover the potentiality for distributed execution can be explored.

Project Summary

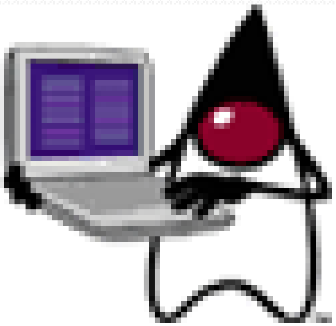
Keywords: Programming Environments, Java, Scientific Software, Scripting, Interpreter, Reflection

License: GNU General Public License (GPL)

Project web site URL: https://jlab.dev.java.net/

Implementation language: java

Platform: totally platform independent- tested on Linux, Solaris and Windows XP and it runs in the same way, on all these different environments, without any change of the code.



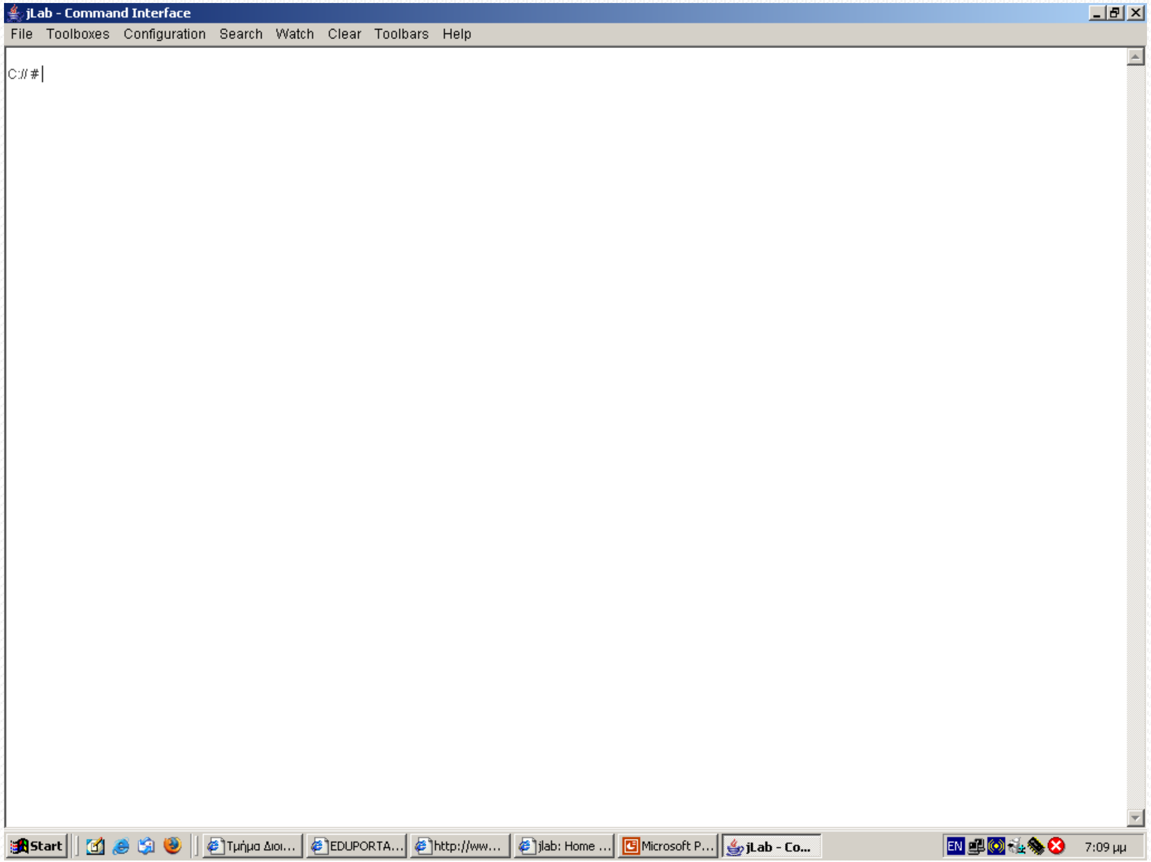
Message from the owner

- The jLab project aims to provide a Matlab/Scilab environment with a scripting interpreter implemented in Java, and with the potential of linking dynamically Java numerical computing code. The system will perform very efficiently since the Java class code executes very fast. Moreover the potentiality for distributed execution can be explored.

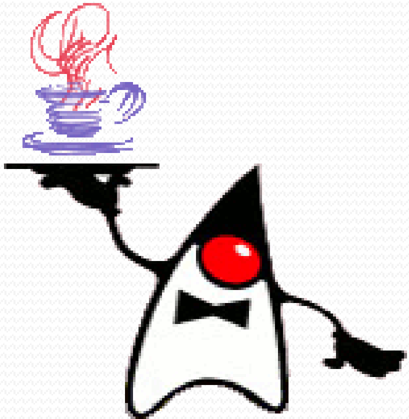
Description

- environment ~ Matlab/Scilab like scripting language that is executed by an interpreter implemented in the Java language.
- This language will support all the basic programming constructs and an extensive set of built in mathematical routines that cover all the basic numerical analysis tasks.
- Moreover, the toolboxes of jLab can be easily implemented in Java and the corresponding classes can be dynamically integrated to the system.
- The efficiency of the Java compiled code can be directly utilized for any computationally intensive operations.
- Since jLab will be coded in pure Java the build from source process is much cleaner, faster, platform independent and less error prone than similar C/C++/Fortran based open source environments (e.g. Scilab, Octave).
- Also the facilities of the Java language for distributed computation will be explored to speed up scientific computations.

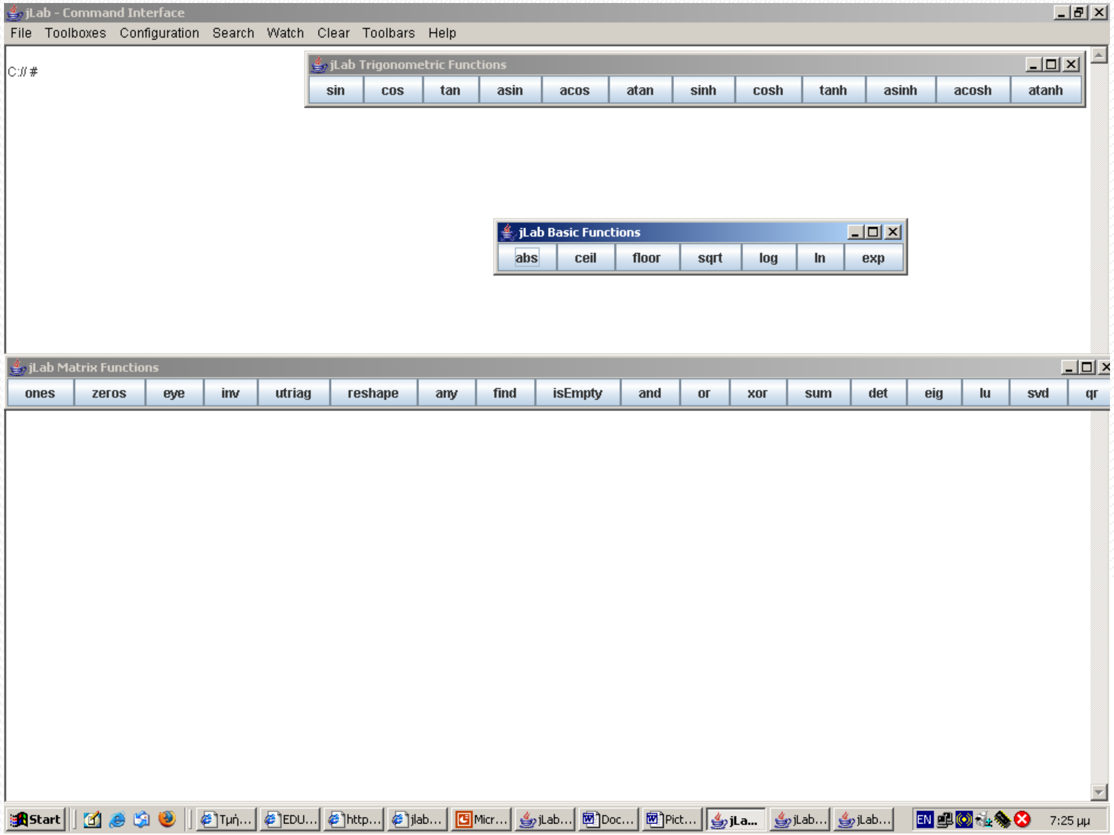
User Interface



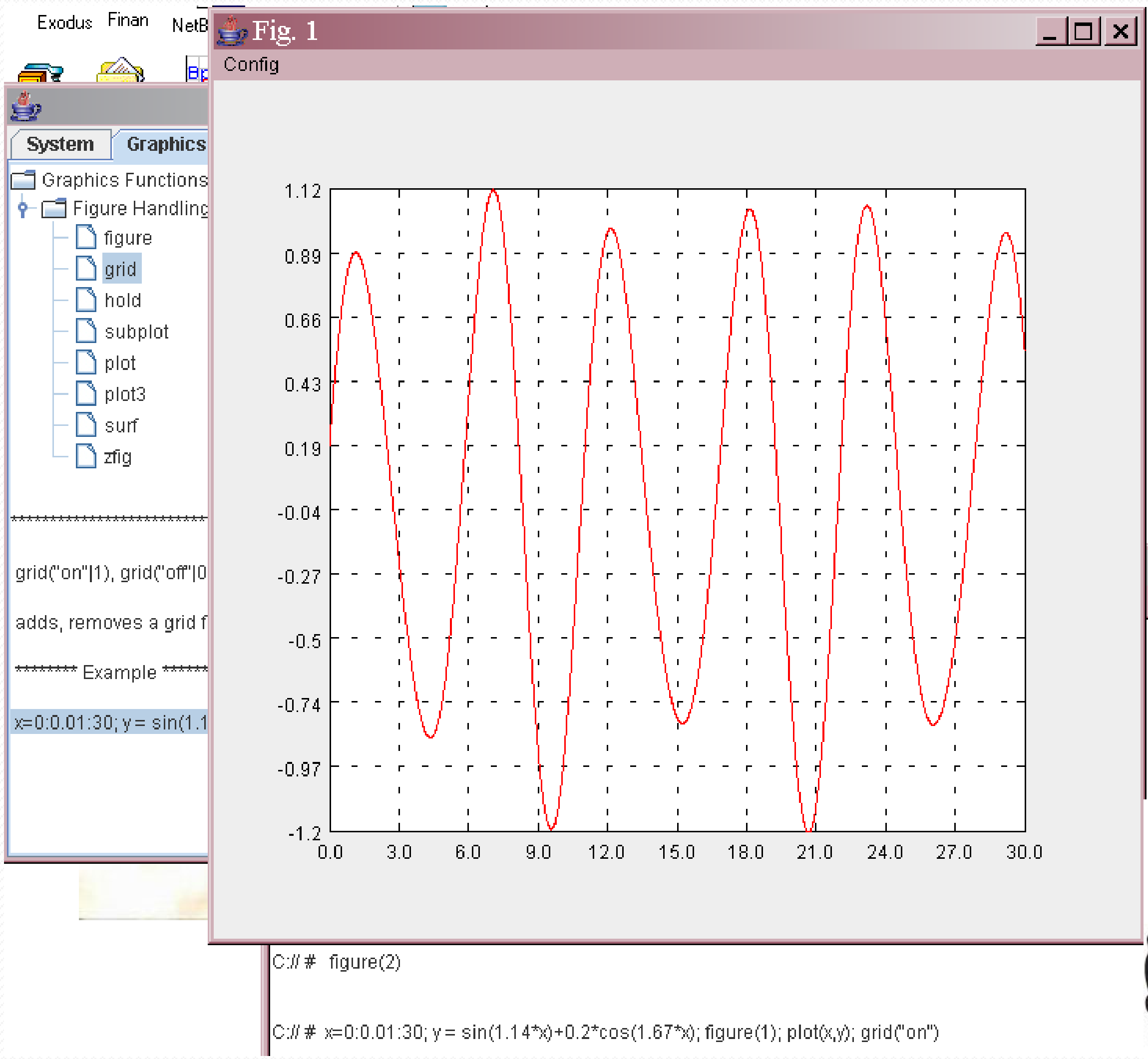
```
C:// #
```



User Interface (2)



```
C:// #
```



****

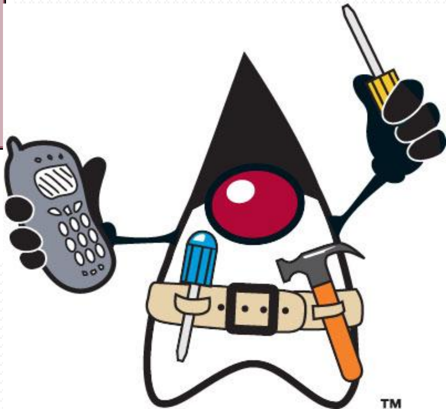
grid("on"|1), grid("off"|0

adds, removes a grid f

******** Example ******

x=0:0.01:30; y = sin(1.1

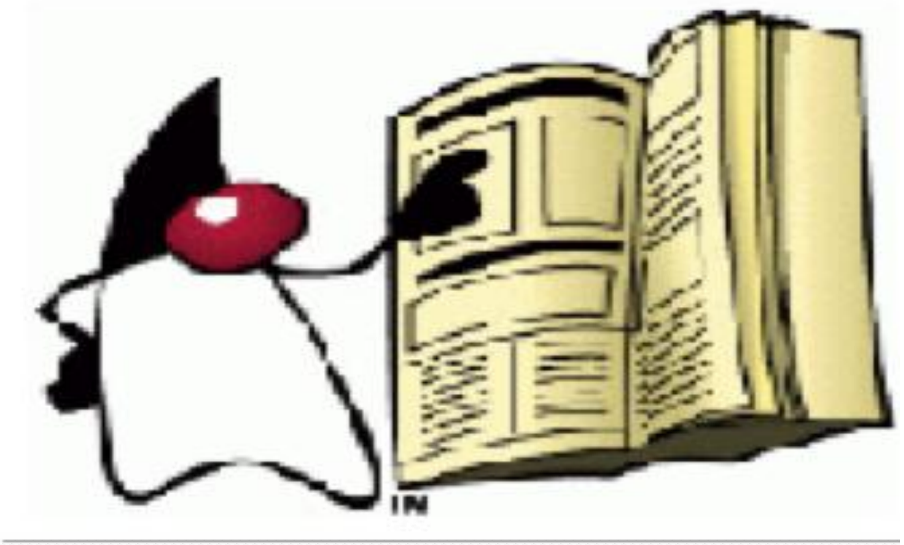
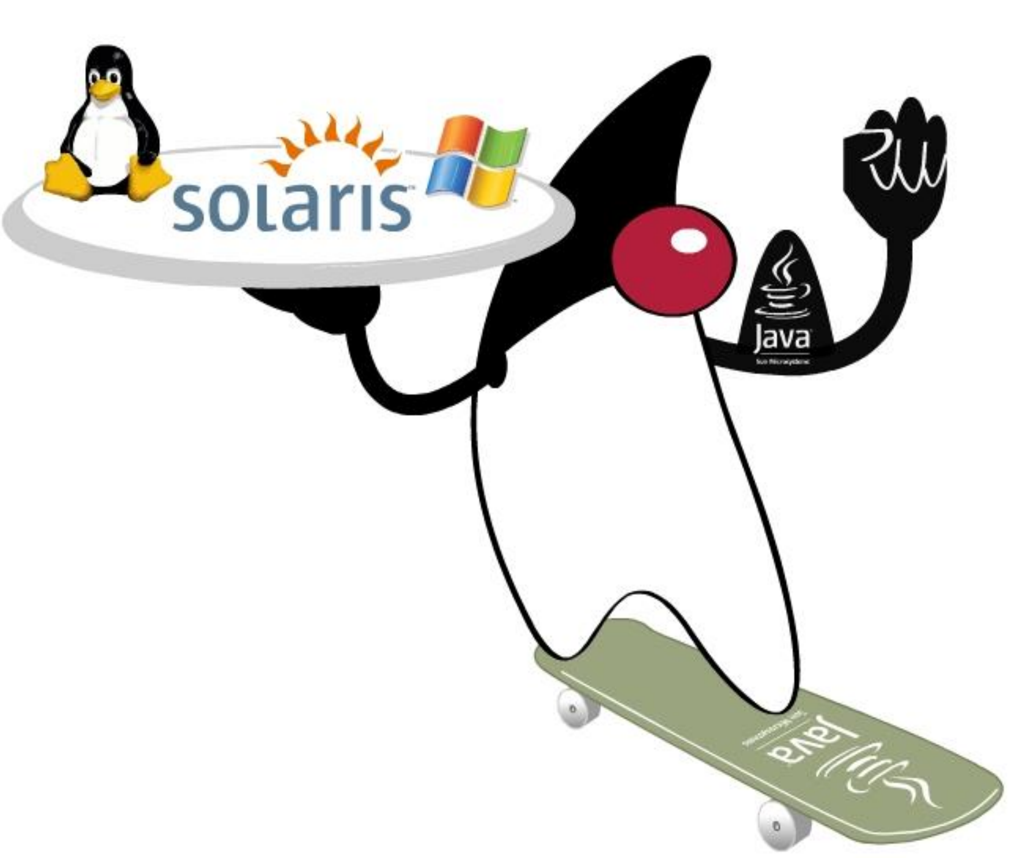
```
C:// #  figure(2)

C:// # x=0:0.01:30; y = sin(1.14*x)+0.2*cos(1.67*x); figure(1); plot(x,y); grid("on")
```



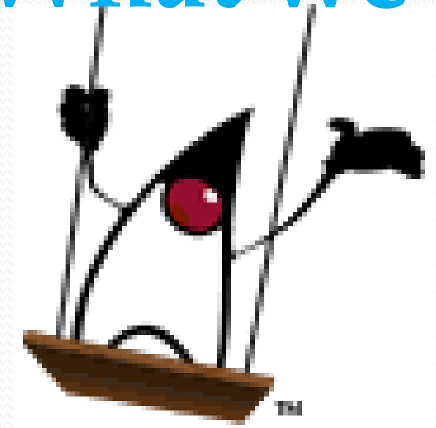
Classes…

Approximate source code size:

- Main project 500+ classes
- Toolbox 130+ classes

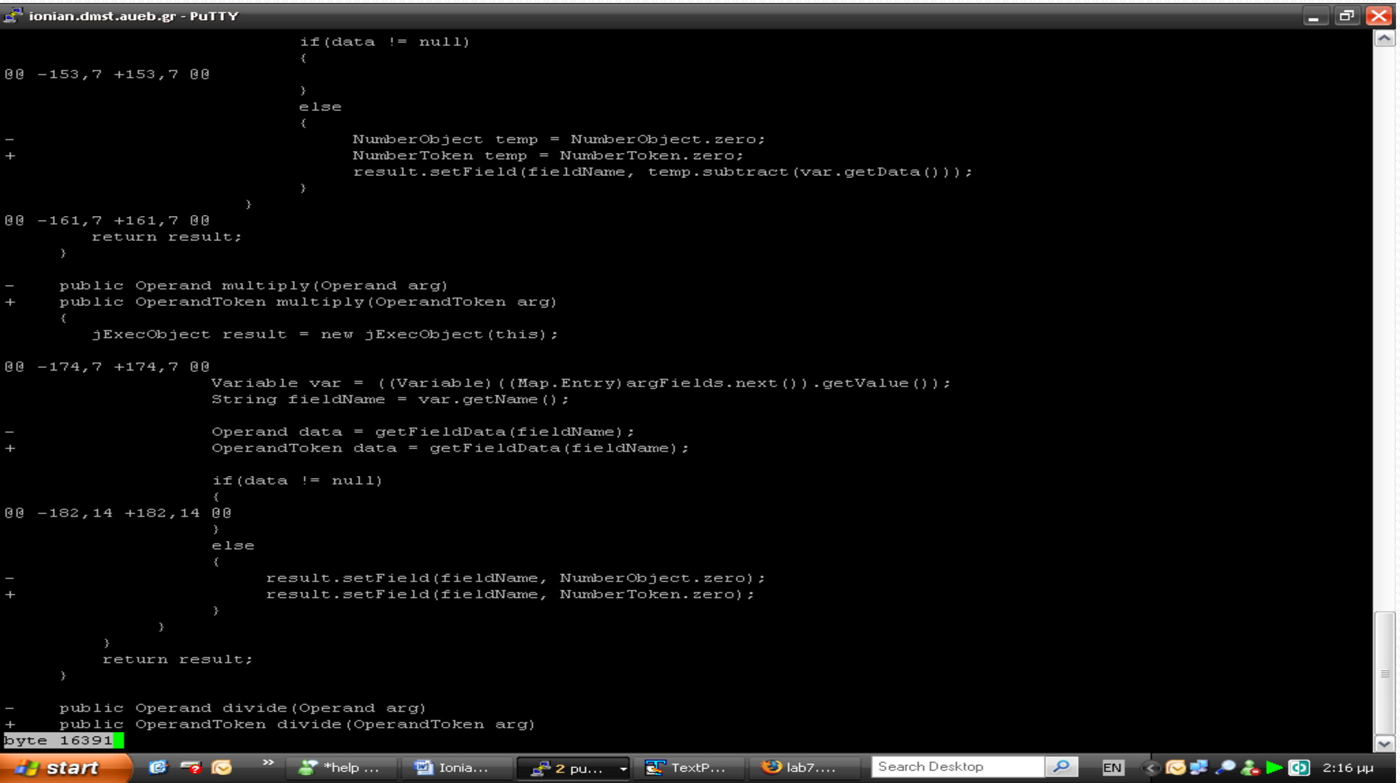
What we planned to add?



- Addition of toolboxes.
  - MathFunctions
    - atan2, IEEEremainder, max, min, pow, random, rint, toDegrees, toRadians
  - Equations
    - first, second (Degree Equations)

Version Control with Subversion – Breadth of changes

- Using this useful tool we can keep track of our changes and different versions of the project. We can see the differences between two versions and examine the summary of changes through time.

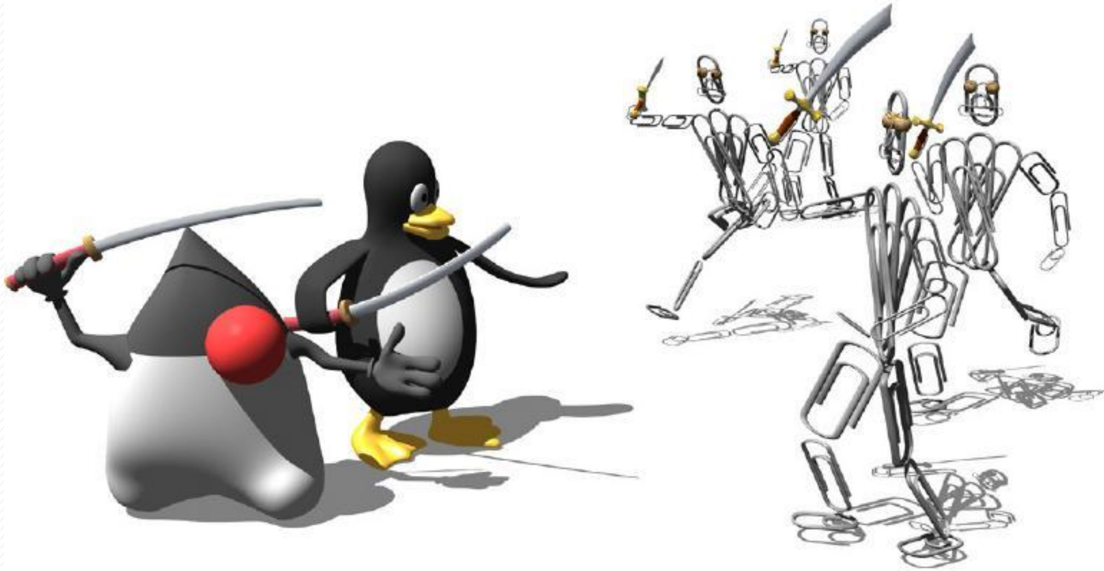


```
                            if(data != null)
                            {
@@ -153,7 +153,7 @@
                            }
                            else
                            {
-                                NumberObject temp = NumberObject.zero;
+                                NumberToken temp = NumberToken.zero;
                                 result.setField(fieldName, temp.subtract(var.getData()));
                            }
                       }
@@ -161,7 +161,7 @@
        return result;
     }

-    public Operand multiply(Operand arg)
+    public OperandToken multiply(OperandToken arg)
     {
        jExecObject result = new jExecObject(this);

@@ -174,7 +174,7 @@
                   Variable var = ((Variable)((Map.Entry)argFields.next()).getValue());
                   String fieldName = var.getName();

-                  Operand data = getFieldData(fieldName);
+                  OperandToken data = getFieldData(fieldName);

                   if(data != null)
                   {
@@ -182,14 +182,14 @@
                   }
                   else
                   {
-                       result.setField(fieldName, NumberObject.zero);
+                       result.setField(fieldName, NumberToken.zero);
                   }
              }
         }
         return result;
     }

-    public Operand divide(Operand arg)
+    public OperandToken divide(OperandToken arg)
byte 16391
```

Our contribution…

Summary of the changes

- AboutGUIDialog
- jExecObject
- OperandToken
- MathFunction
- NumberToken
- FunctionManager
- Equations
- AddSubOperatorToken
- jExec\Tokens\Expression
- jExec\gui\Console
- jExec.Det.*
- svm_predict
- Errors
- svm_train
- load
- FunctionToken
- comments and documentation to m
- .properties

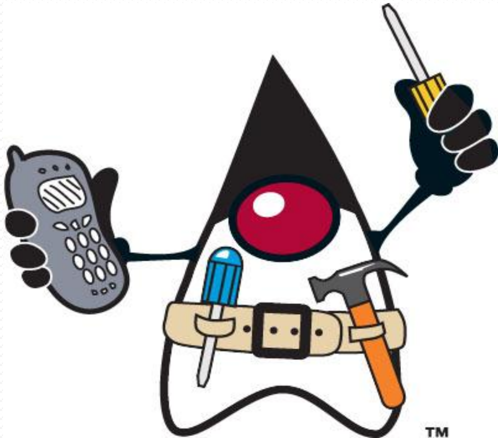


Understanding and documentation of legacy system

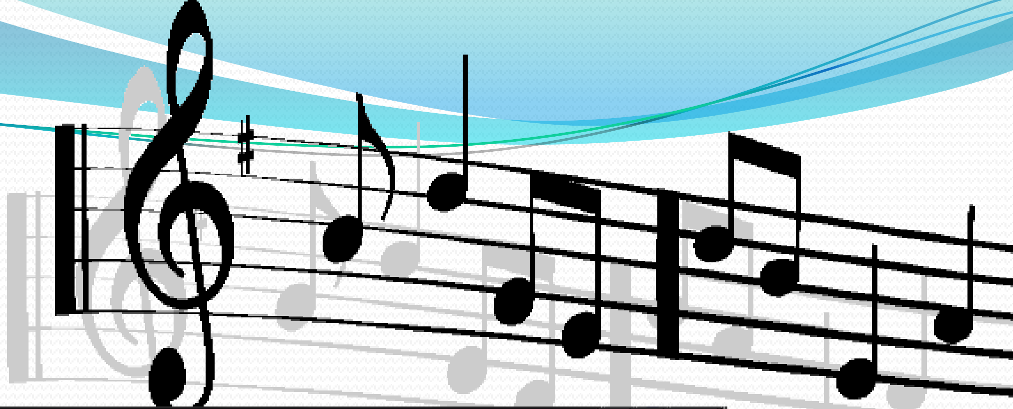
- In gemini (Red Hat linux 9 Server) we can list contents of directories in a tree-like format. In this way we can see how our project is organized.

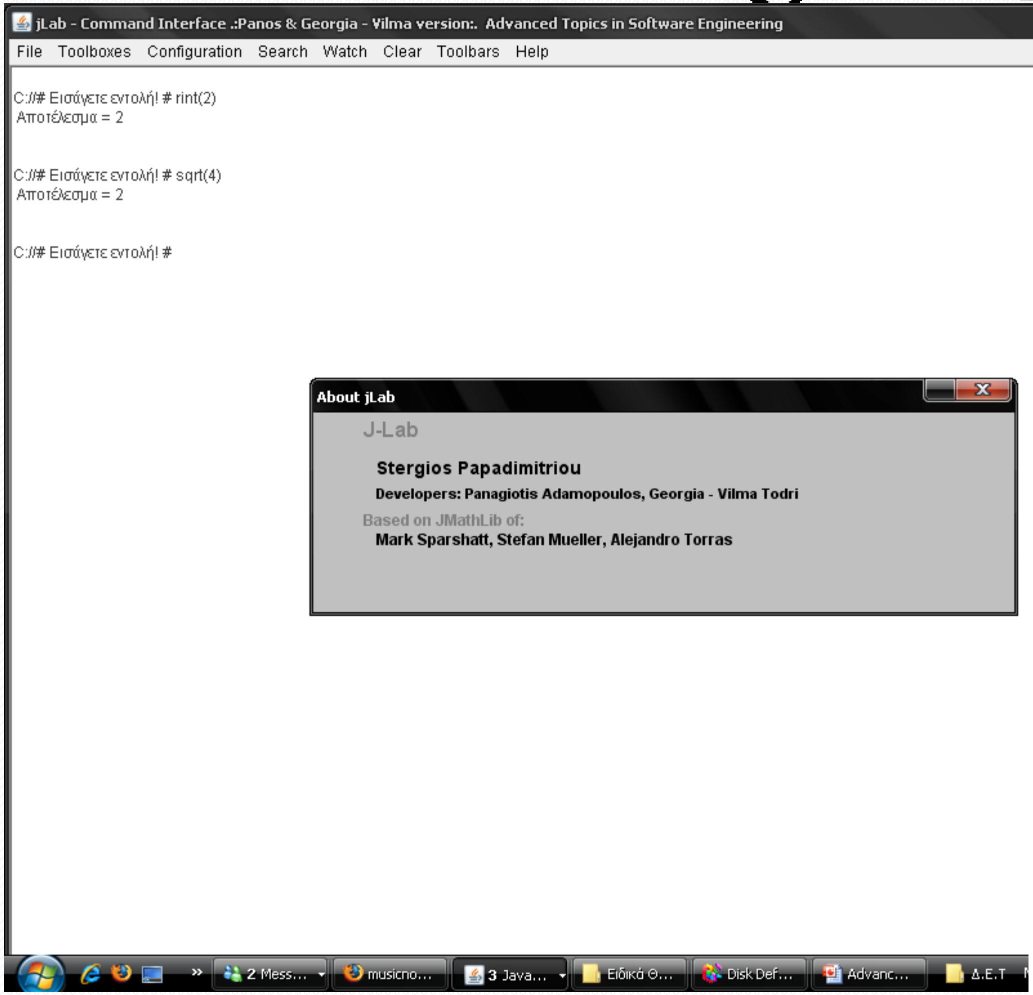
  We execute the command **tree –dl jLabSrc**

  60 directories in JLabSrc!



User Interface





```
C://# Εισάγετε εντολή! # rint(2)
 Αποτέλεσμα = 2


C://# Εισάγετε εντολή! # sqrt(4)
 Αποτέλεσμα = 2


C://# Εισάγετε εντολή! #
```

About jLab

J-Lab

Stergios Papadimitriou

Developers: Panagiotis Adamopoulos, Georgia - Vilma Todri

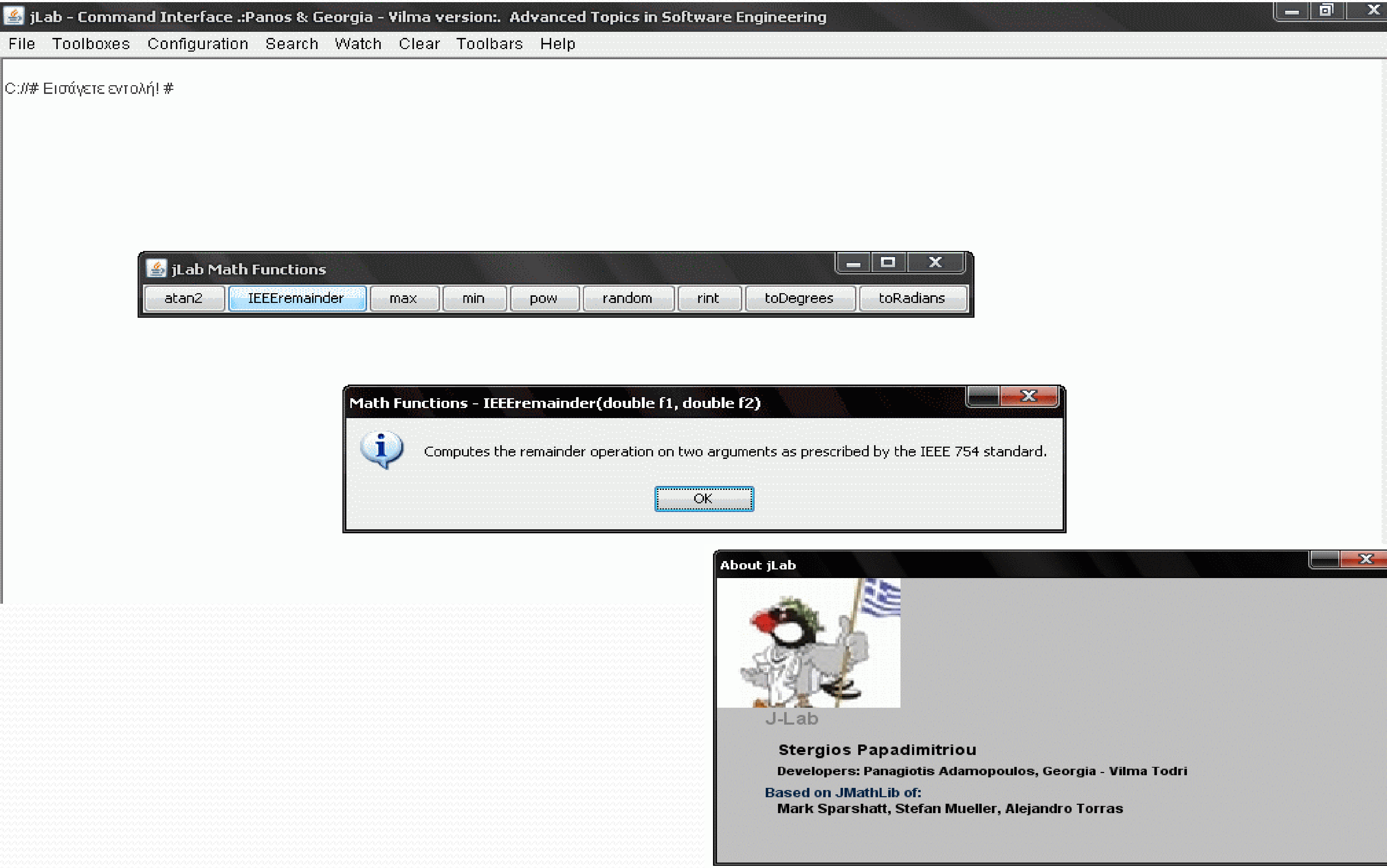
Based on JMathLib of:

Mark Sparshatt, Stefan Mueller, Alejandro Torras

Consistency in Formatting

- Opening brace on separate line.
- Standard values often appear as editor commands.
- Consistent coding style
- Readable structure etc.
- To produce our changes we used various programs. For example, Netbeans IDE, Eclipse IDE and Textpad.

Screenshots of some changes



jLab - Command Interface .:Panos & Georgia - Vilma version:. Advanced Topics in Software Engineering

C://# Εισάγετε εντολή! #

jLab Math Functions

Math Functions - IEEEremainder(double f1, double f2)

Computes the remainder operation on two arguments as prescribed by the IEEE 754 standard.

About jLab

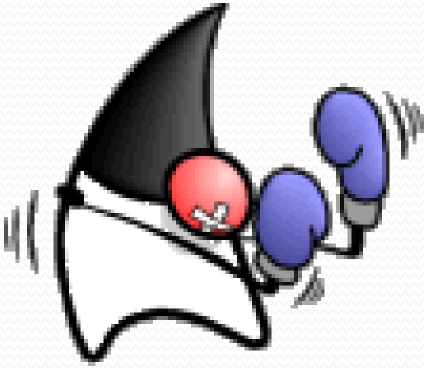
J-Lab

Stergios Papadimitriou

Developers: Panagiotis Adamopoulos, Georgia - Vilma Todri

Based on JMathLib of:
Mark Sparshatt, Stefan Mueller, Alejandro Torras

Detailed changes..



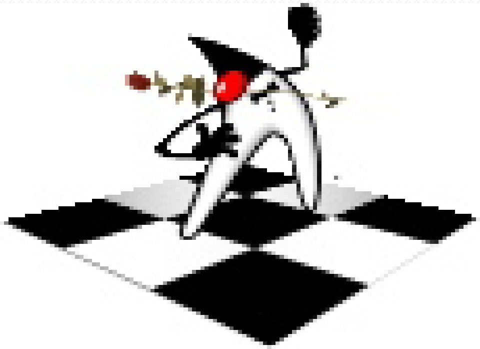
Matrix

- exp : Calculates the exponent of a complex number and takes as arguments the value as an array of double. The result is also an array of double
- floor : Rounds the value of the first operand down to the nearest integer. Takes as argument a double array and return the result as an operand token
- ln : Returns the natural logarithm of value. Takes as argument an array of double and return the result as an array of double too.

Detailed changes.. (2)

- log : Returns the logarithm of value to the base. Takes as argument an array of double and return the result as an array of double too.
- round : Rounds a value to the nearest integer. Takes as argument an array of double and returns the result as an OperandToken
- sqrt : Calculates the sqrt of a complex number. Takes as argument an array of double and return the result as an array of double too.
- sum : Returns the sum of all the elements of a matrix per column. Takes as argument the matrix to sum as an operand.

Detailed changes.. (3)



Math

- atan2: Converts rectangular coordinates (x, y) to polar (r, *theta*)
- IEEEremainder: Computes the remainder operation on two arguments as prescribed by the IEEE 754 standard.
- Max: Returns the greater of two double values.
- Min : Returns the smaller of two double values.
- Pow : Returns the value of the first argument raised to the power of the second argument.

Detailed changes.. (4)

- Random : Returns a double value with a positive sign, greater than or equal to 0.0 and less than 1.0.
- Rint : Returns the double value that is closest in value to the argument and is equal to a mathematical integer.
- toDegrees: Converts an angle measured in radians to an approximately equivalent angle measured in degrees.
- toRadians : Converts an angle measured in degrees to an approximately equivalent angle measured in radians.

Detailed changes.. (5)

- We added many classes and packages that we found in the web…

  For example some financial packages that can be loaded by using the corresponding option.
- We added a midi player so as “welcome” and “error” sounds to be played.
- We changed the user interface (UI).

- Many small changes more…

Code…

```
/**Executes the equation - the code run is based on the index number
   @param operands - the array of parameters
   @return the result of the function as an OperandToken*/
  public OperandToken evaluate(Token[] operands)
  {
     OperandToken result = null;
     String input = operands.toString();
     OperandToken result1 = new NumberToken(0);
     //execute the equation depending on the index
   switch(index)
   {
        case FIRST:
           double a = ((NumberToken)operands[0]).getValue();
           double b = ((NumberToken)operands[3]).getValue();
           double g = ((NumberToken)operands[6]).getValue();
           g = b-g;
           if (a!=0) {
              double temp_result = - b / a;
              result = new NumberToken(temp_result);
           } else { ........
```

Code (2)…

```
public class sqrt extends ExternalElementWiseFunction
{
    public sqrt()    {
      name = "sqrt";
      }
    public double[] evaluateValue(double[] arg)    {
        double[] result = new double[2];
        double   re      = arg[REAL];
        double   im      = arg[IMAG];
        double temp = Math.pow(re, 2) + Math.pow(im, 2);
        double mag  = Math.sqrt(temp);
        if (mag > 0.0)  {
            if (re > 0.0) {
                temp =  Math.sqrt(0.5 * (mag + re));
                re =  temp;
                im =  0.5 * im / temp;
            }
            else
        .....
        result[REAL] = re;
        result[IMAG] = im;
```

Code (3)…

```
/**Calculates the arctangent of a complex number
   * @param arg = the value as an array of double
   * @return the result as an array of double*/
  public OperandToken rint() {
      double[][][] results = new double[sizeY][sizeX][2];
      for (int yy=0; yy<sizeY; yy++) {
          for (int xx=0; xx<sizeX; xx++) {
              results[yy][xx][REAL]       =
  java.lang.Math.rint(values[yy][xx][REAL]);
              results[yy][xx][IMAGINARY] =
  java.lang.Math.rint(values[yy][xx][IMAGINARY]);
          }
      }
      return new NumberToken(results);
  }
```

Fix code

- **Before**

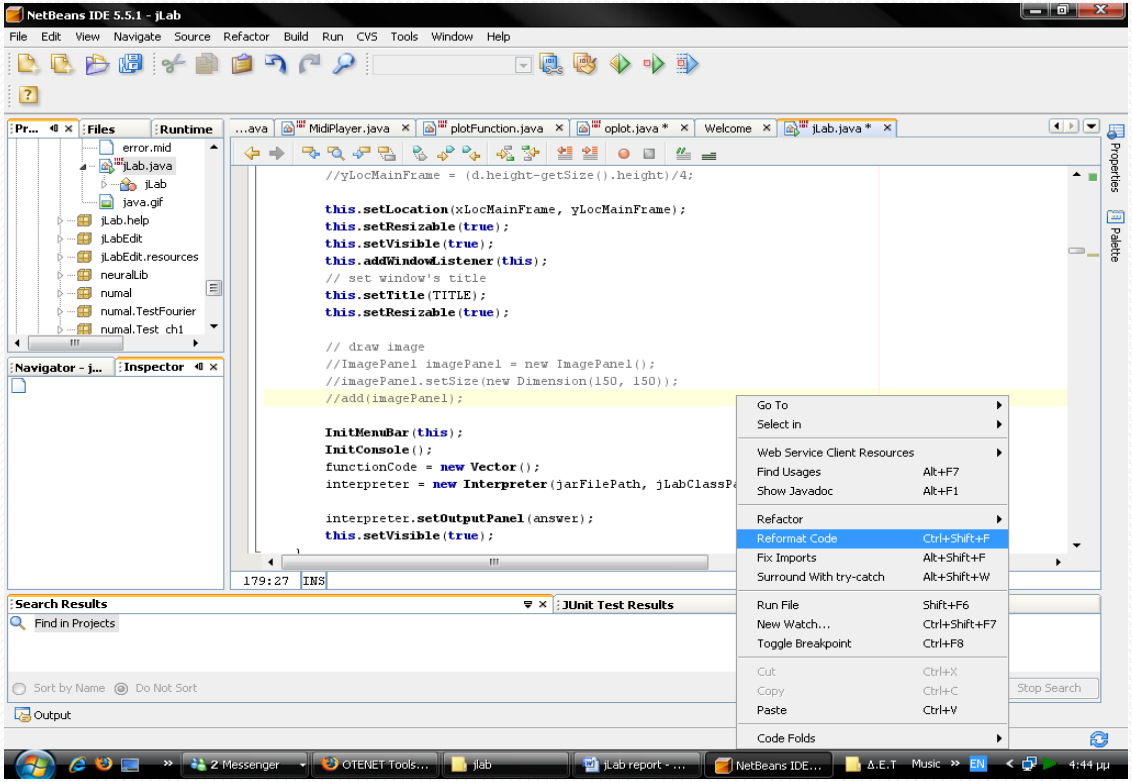
```
Container box = Box.createHorizontalBox();
box.add(bones);box.add(bzeros);box.add(beye);box.add(binv
);box.add(butriag);box.add(breshape);
box.add(bany);box.add(bfind);box.add(bisEmpty);
```

- **After**

```
// A (AWT) container object that contains boxes (other AWT
components).
        Container box = Box.createHorizontalBox();
//  Appends the specified component to the end of this
container.               box.add(bones);
        box.add(bzeros);
        box.add(beye);  ……
```

Formatting Java Source Code

- The IDE automatically formats your code.



```
    //yLocMainFrame = (d.height-getSize().height)/4;

    this.setLocation(xLocMainFrame, yLocMainFrame);
    this.setResizable(true);
    this.setVisible(true);
    this.addWindowListener(this);
    // set window's title
    this.setTitle(TITLE);
    this.setResizable(true);

    // draw image
    //ImagePanel imagePanel = new ImagePanel();
    //imagePanel.setSize(new Dimension(150, 150));
    //add(imagePanel);

    InitMenuBar(this);
    InitConsole();
    functionCode = new Vector();
    interpreter = new Interpreter(jarFilePath, jLabClassPa

    interpreter.setOutputPanel(answer);
    this.setVisible(true);
```

- But in some cases the output wasn’t satisfactory. So, we had to do it manually until we produced an efficient output.

Batch files

```
SET CLASSDIR=..\build\classes
SET SOURCEDIR=jLabSrc
SET JAVAC_OPTS=-classpath %classpath%;dist\jLab.jar;. -d %classdir%

javac %javac_opts% jLab\*.java
javac %javac_opts% jLab\Graph\*.java
javac %javac_opts% jLab\wavelets\*.java
javac %javac_opts% jLab\weka\*.java ...

cd classes
java  -classpath dist\jLab.jar;
```

Integration

In order to make our changes and add functionality to jLab project we had to integrate our source code into the rest of the project.

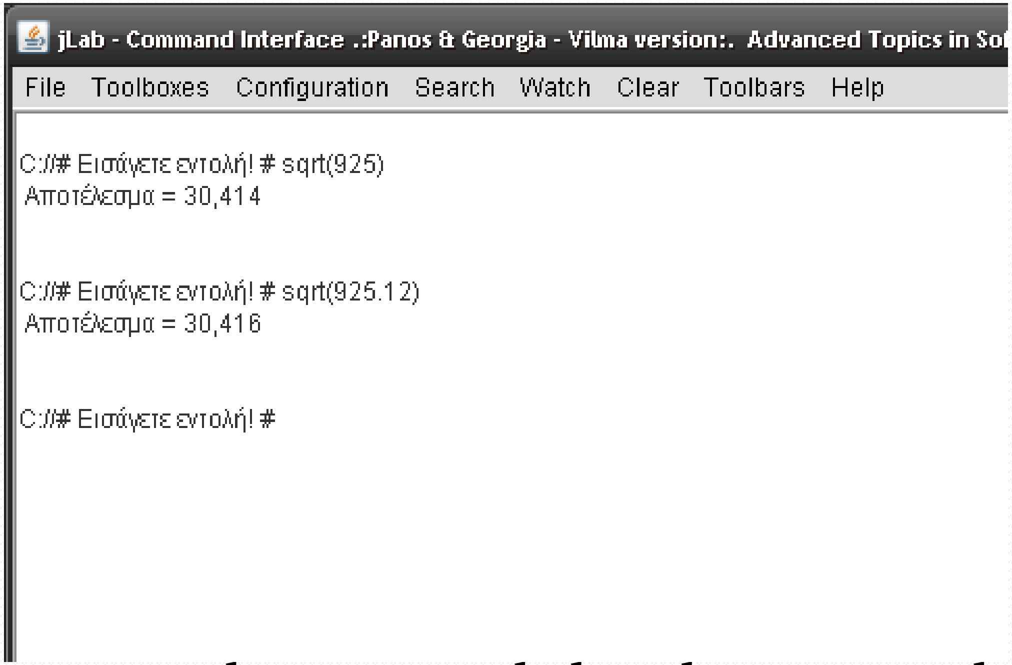
Specifically, as far as matrix functions are concerned, for example, we created and extended the class ExternalElementWiseFunction.java which extends ExternalFunction.java that is a class that already existed and is considered base class for all external function classes.

Moreover, in many cases we added source code in already existing classes without disturbing the legacy project!

Testing

Scenarios

- As far as testing is concerned, we conducted test cases and implement examples of actual use.


jLab - Command Interface .:Panos & Georgia - Vilma version:. Advanced Topics in Sof
File
Toolboxes
Configuration
Search
Watch
Clear
Toolbars
Help

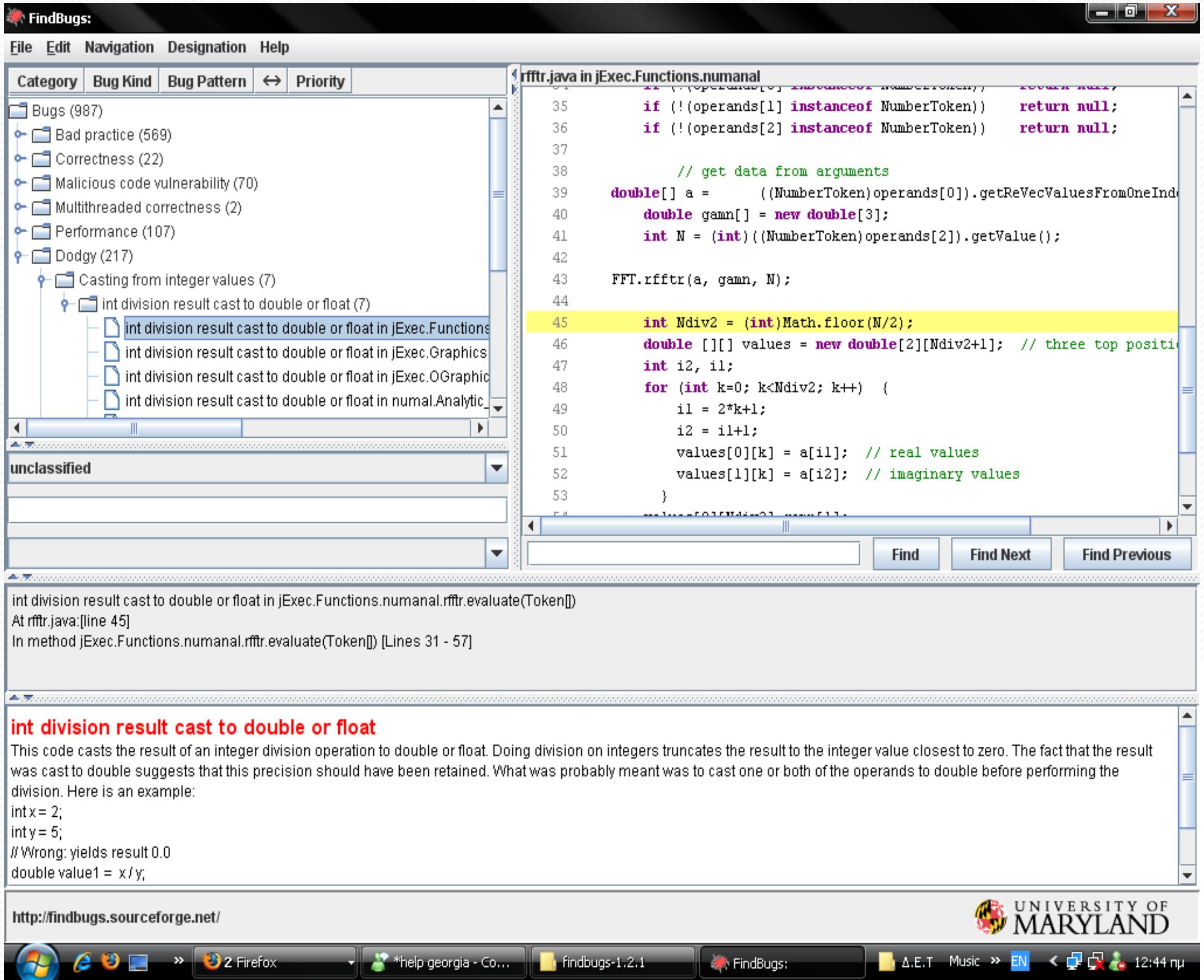
```
C://# Εισάγετε εντολή! # sqrt(925)
 Αποτέλεσμα = 30,414


C://# Εισάγετε εντολή! # sqrt(925.12)
 Αποτέλεσμα = 30,416


C://# Εισάγετε εντολή! #
```

- In this way we have tested the changes and confirm that they work in the way they are supposed to do so.

Debug

Findbugs

- It looks for instances of "bug patterns" --- code instances that are likely to be errors.



Bugs (987)
- Bad practice (569)
- Correctness (22)
- Malicious code vulnerability (70)
- Multithreaded correctness (2)
- Performance (107)
- Dodgy (217)
  - Casting from integer values (7)
    - int division result cast to double or float (7)
      - int division result cast to double or float in jExec.Functions
      - int division result cast to double or float in jExec.Graphics
      - int division result cast to double or float in jExec.OGraphic
      - int division result cast to double or float in numal.Analytic_

rfftr.java in jExec.Functions.numanal

```
35        if (!(operands[1] instanceof NumberToken))    return null;
36        if (!(operands[2] instanceof NumberToken))    return null;
37
38            // get data from arguments
39     double[] a =      ((NumberToken)operands[0]).getReVecValuesFromOneInd
40        double gamn[] = new double[3];
41        int N = (int)((NumberToken)operands[2]).getValue();
42
43     FFT.rfftr(a, gamn, N);
44
45        int Ndiv2 = (int)Math.floor(N/2);
46        double [][] values = new double[2][Ndiv2+1];  // three top positi
47        int i2, i1;
48        for (int k=0; k<Ndiv2; k++)  {
49           i1 = 2*k+1;
50           i2 = i1+1;
51           values[0][k] = a[i1];  // real values
52           values[1][k] = a[i2];  // imaginary values
53         }
```

int division result cast to double or float in jExec.Functions.numanal.rfftr.evaluate(Token[])
At rfftr.java:[line 45]
In method jExec.Functions.numanal.rfftr.evaluate(Token[]) [Lines 31 - 57]

int division result cast to double or float

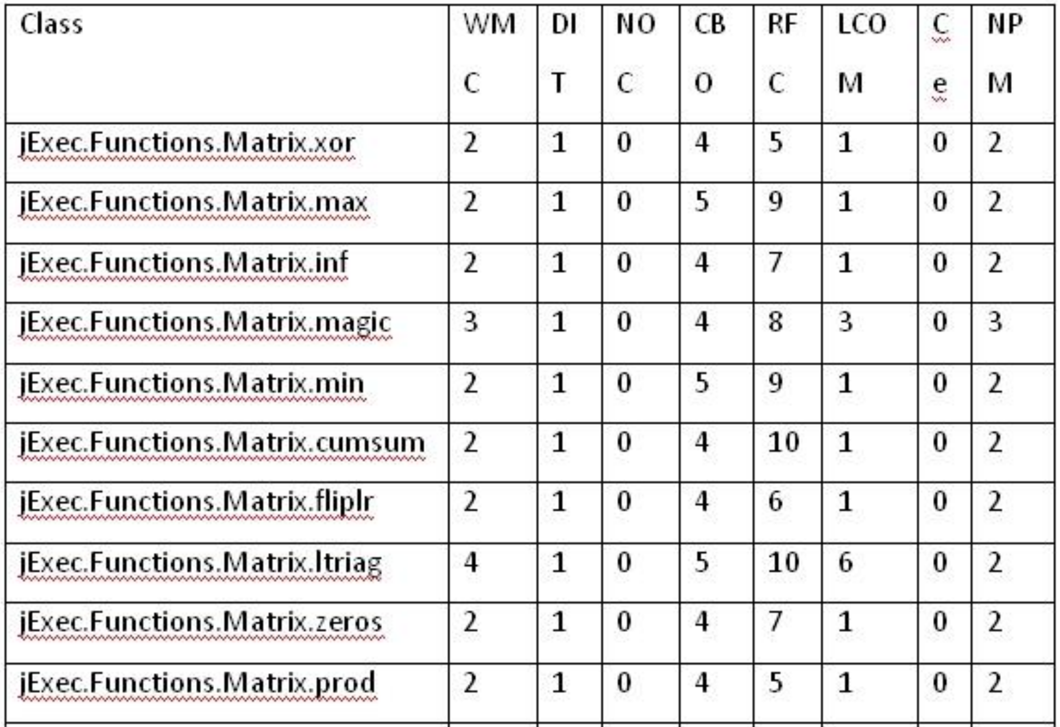
This code casts the result of an integer division operation to double or float. Doing division on integers truncates the result to the integer value closest to zero. The fact that the result was cast to double suggests that this precision should have been retained. What was probably meant was to cast one or both of the operands to double before performing the division. Here is an example:

```
int x = 2;
int y = 5;
// Wrong: yields result 0.0
double value1 =  x / y;
```

http://findbugs.sourceforge.net/

ckjm

- The program *ckjm* calculates Chidamber and Kemerer object-oriented metrics by processing the bytecode of compiled Java files.
- C:\>java -jar ckjm-1.7.jar C:\build\classes\jExec\Functions\Matrix\*.class



Class	WMC	DIT	NOC	CBO	RFC	LCOM	Ce	NPM
jExec.Functions.Matrix.xor	2	1	0	4	5	1	0	2
jExec.Functions.Matrix.max	2	1	0	5	9	1	0	2
jExec.Functions.Matrix.inf	2	1	0	4	7	1	0	2
jExec.Functions.Matrix.magic	3	1	0	4	8	3	0	3
jExec.Functions.Matrix.min	2	1	0	5	9	1	0	2
jExec.Functions.Matrix.cumsum	2	1	0	4	10	1	0	2
jExec.Functions.Matrix.fliplr	2	1	0	4	6	1	0	2
jExec.Functions.Matrix.ltriag	4	1	0	5	10	6	0	2
jExec.Functions.Matrix.zeros	2	1	0	4	7	1	0	2
jExec.Functions.Matrix.prod	2	1	0	4	5	1	0	2

Coordination with the development team – Mails

- **Από:** "root" <sterg@philippos.teikav.edu.gr>

Mpravo gia tin grigori prosarmogi sas ston kodika!!

> Fisiko einai na xathite ston kodika giati einai poliplokos> kai xriazetai prosektiko diavasma.> > Stergios

Documentation

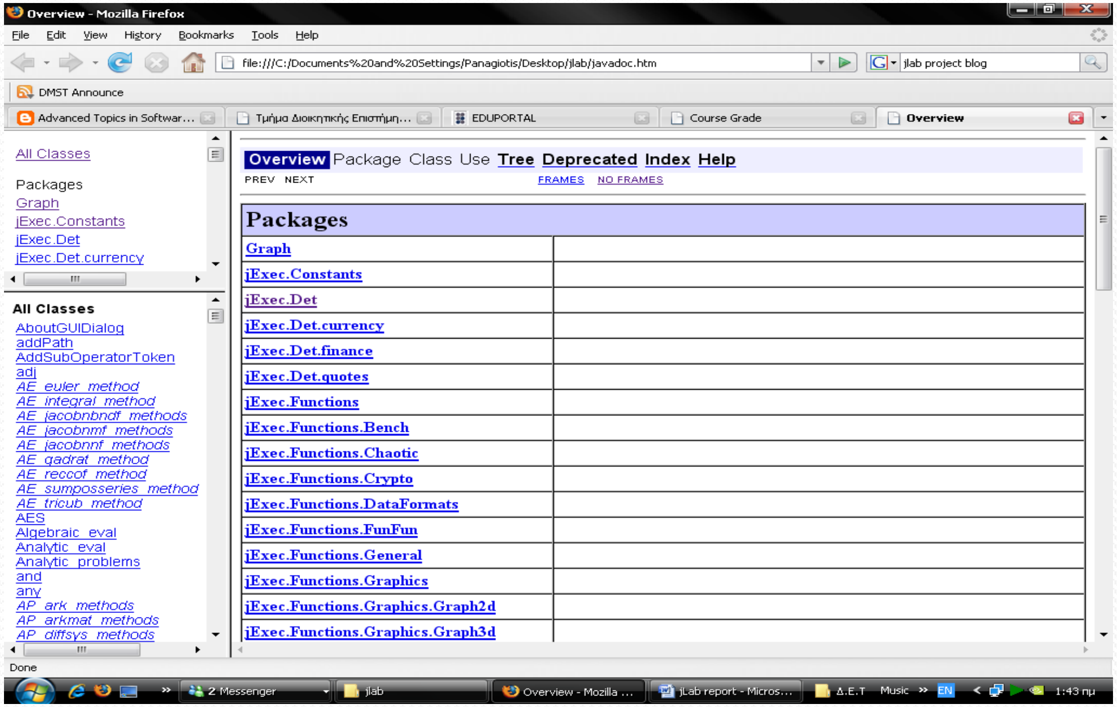
We used Java /** comments that are read by javadoc.

For example:

```
/**Calculates the logarithm of a complex number
   @param arg = the value as an array of double
   @return the result as an array of double*/
   public double[] evaluateValue(double[] arg)
   { …..
```

Documentation (2)

We used a doclet to generate the documentation. The standard doclet generates HTML and is built into the Javadoc tool.



All Classes

Packages
Graph
jExec.Constants
jExec.Det
jExec.Det.currency

All Classes
AboutGUIDialog
addPath
AddSubOperatorToken
adj
AE_euler_method
AE_integral_method
AE_jacobnbndf_methods
AE_jacobnmf_methods
AE_jacobnnf_methods
AE_qadrat_method
AE_reccof_method
AE_sumposseries_method
AE_tricub_method
AES
Algebraic_eval
Analytic_eval
Analytic_problems
and
any
AP_ark_methods
AP_arkmat_methods
AP_diffsys_methods

Packages	
Graph	
jExec.Constants	
jExec.Det	
jExec.Det.currency	
jExec.Det.finance	
jExec.Det.quotes	
jExec.Functions	
jExec.Functions.Bench	
jExec.Functions.Chaotic	
jExec.Functions.Crypto	
jExec.Functions.DataFormats	
jExec.Functions.FunFun	
jExec.Functions.General	
jExec.Functions.Graphics	
jExec.Functions.Graphics.Graph2d	
jExec.Functions.Graphics.Graph3d	

Hopefully, we didn’t reached any XXX (means something is probably wrong here) TODO (marks areas of further work) or FIXME (marks areas of further enhancement) comments.

Blog

We tried to keep our blog up-to-date. So, we did many posts which explain our contribution in jLab project.

Our blog is written mostly in Greek.

The post that is included below is our first post in our blog.

“Αυτό το Blog δημιουργήθηκε με σκοπό να κρατάει αναφορές για την εργασία στο μάθημα «Ειδικά Θέματα Τεχνολογίας Λογισμικού». Η εργασία αφορά στην συνεισφορά μας σε ένα project ανοιχτού κώδικα. θα προσπαθήσουμε να το ανανεώνουμε συνεχώς!”

URL: http://project-jlab.blogspot.com/

Ευχαριστούμε…