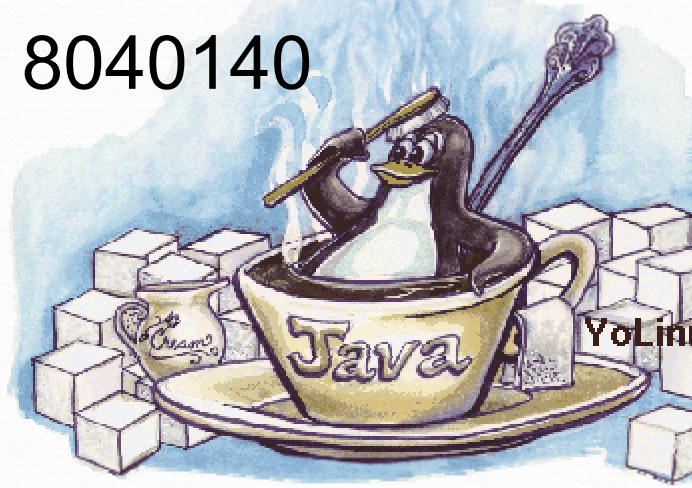
Athens University of Economics & Business
Department of Management Science and Technology

Advanced Topics in Software Engineering

Project Team – jLab

Φοιτητές:

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



jLab - A little bit of revision

- 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.

Description–A little bit of revision

- 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.

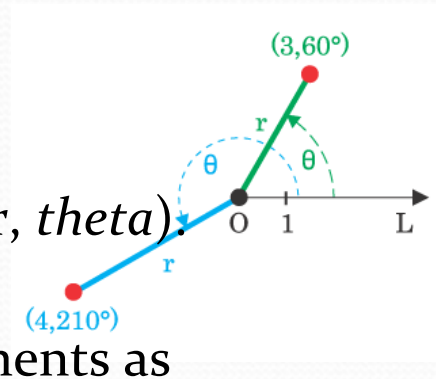
What to add???

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

Last minute…..!

- Play alert / error sounds

Math Functions



- atan2(double y, double x) :
  Converts rectangular coordinates (x, y) to polar (r, *theta*).
- IEEEremainder(double f1, double f2) :
  Computes the remainder operation on two arguments as prescribed by the IEEE 754 standard.
- max(double a, double b) :
  Returns the greater of two double values.
- min(double a, double b) :
  Returns the smaller of two double values.
- pow(double a, double b)
  Returns the value of the first argument raised to the power of the second argument.

and....

- random_()

  Returns a double value with a positive sign, greater than or equal to 0.0 and less than 1.0.

- rint(double a) :

  Returns the double value that is closest in value to the argument and is equal to a mathematical integer.

- toDegrees(double angrad)

  Converts an angle measured in radians to an approximately equivalent angle measured in degrees.

- toRadians(double angdeg)

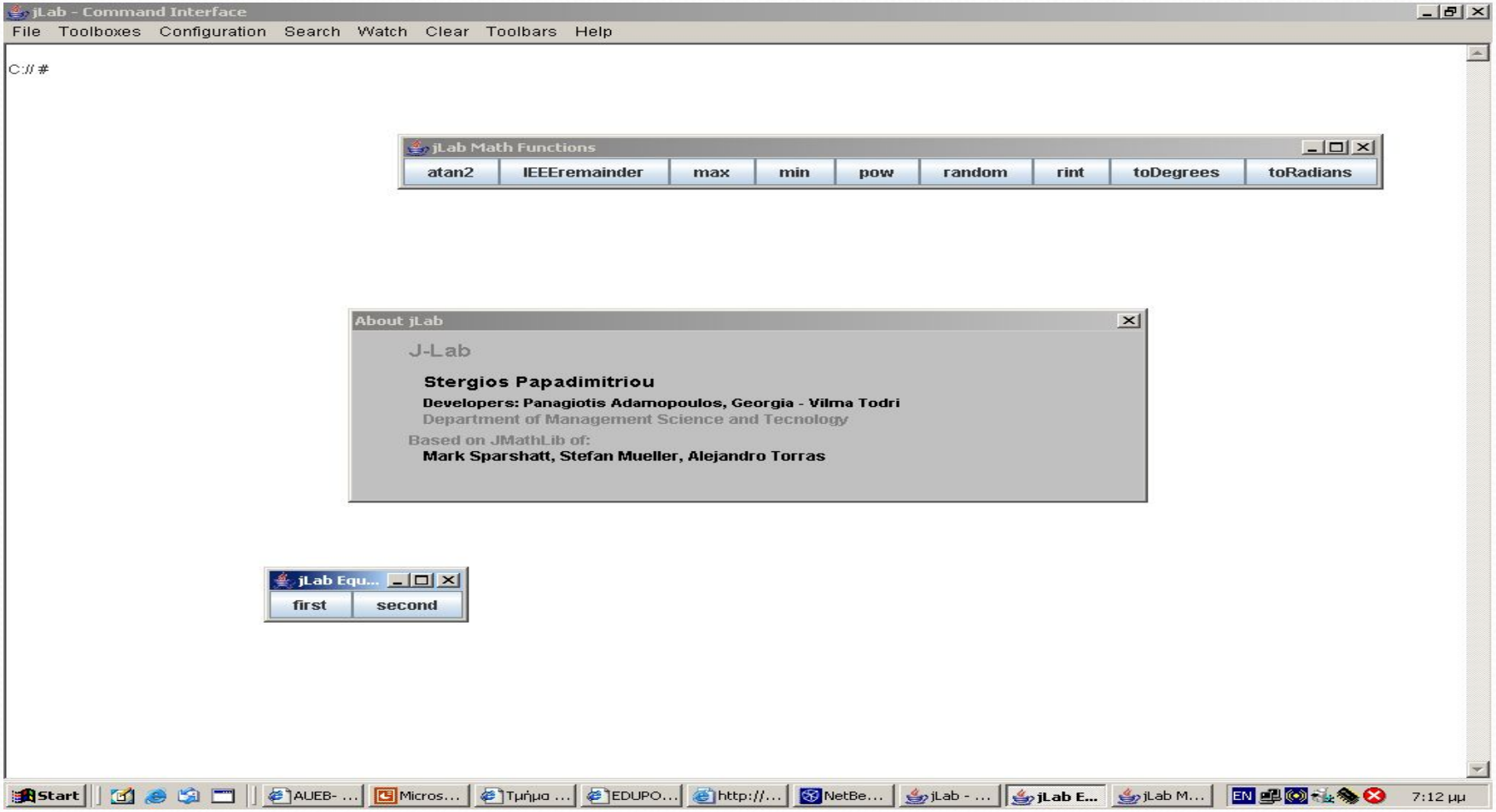
  Converts an angle measured in degrees to an approximately equivalent angle measured in radians.

Our contribution…

Which classes we are going to “implement”..

- Equations
- MathFunction
- NumberToken
- FunctionManager
- ExecObject
- OperandToken
- AboutGUIDialog
- jLab
- MathFunctionToolbar
- EquationFunctionToolbar

User Interface



Code…

$$x = -\frac{b}{2a} \pm \frac{\sqrt{b^2 - 4ac}}{2a} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}.$$

```
/**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 { ……..
```

Ευχαριστούμε..

Ερωτήσεις;;;