Sustainable Software Product Lines via Generation

Spencer Smith and Jacques Carette

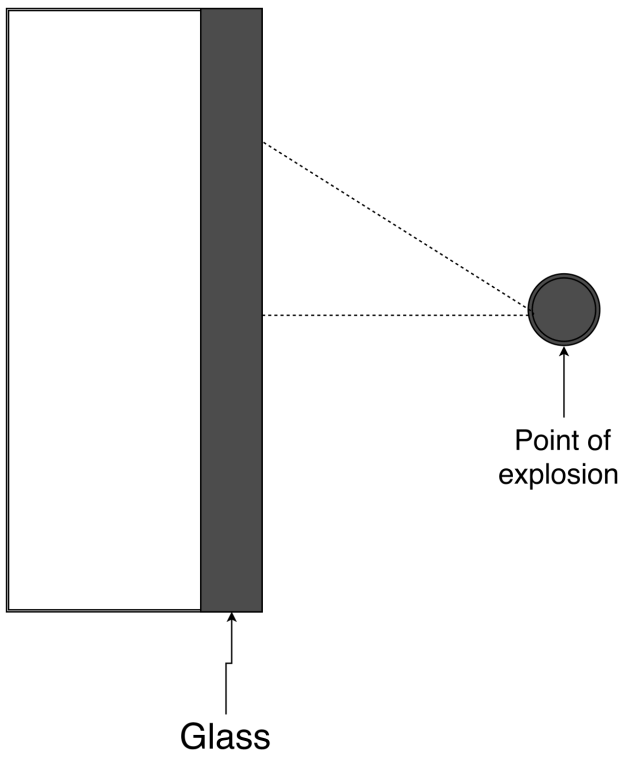
Computing and Software Department
Faculty of Engineering
McMaster University

Huawei 2023 Visit: Feb, 2023

Generate All Things with Drasil

- **Goal** – Improve quality of SCS
- **Idea**
  - Adapt ideas from SE
  - Document requirements, design, verification, etc.
    - Good – improves quality
    - Bad – too much work, inevitable change, too hard to maintain
- **Solution**
  - Capture knowledge once
  - Generate all documentation and code
- **Implement Solution – Drasil**
  - Facilitates change
  - Traceability
  - Reproducibility
  - Sustainability
  - Certification
  - Captures best practices

GlassBR



Given

- dimensions of glass plane
- glass type
- explosion characteristics
- tolerable breakage probability

Predict whether the glass will withstand the explosion

Drasil Inputs:

- Program Name: GlassBR
- Authors: Nikitha K and Spencer S
- Symbols: tolerable load ($\hat{q}_{\text{tol}}$), Risk of failure (B), ...
- Assumptions: Load duration factor constant,
- Data definitions: relation for B, ...
- Design decisions:
    - Modularity (input module),
    - Implementation Type (Program),
    - Logging (Yes),
    - Input Structure (Bundled),
    - Constant Structure (Inlined),
    - Constant Rep (Constants),
    - Real Number Rep (Double),
    - ...

Drasil Inputs:

- Program Name: GlassBR
- Authors: Nikitha K and Spencer S
- Symbols: tolerable load ($\hat{q}_{\text{tol}}$), Risk of failure (B), ...
- Assumptions: Load duration factor constant,
- Data definitions: relation for B, ...
- Design decisions:
    - Modularity (input module),
    - Implementation Type (Program),
    - Logging (Yes),
    - Input Structure (Bundled),
    - Constant Structure (Inlined),
    - Constant Rep (Constants),
    - Real Number Rep (Double),
    - ...

/glassbr
/Website/GlassBR_SRS.html
/Website/GlassBR_SRS.css
/SRS/bibfile.bib
/SRS/Makefile
/SRS/GlassBR_SRS.tex
/SRS/GlassBR_SRS.pdf
/src/python
/src/python/README.md
/src/python/InputParameters.py
/src/python/Calculations.py
/src/python/Makefile
/src/python/doxConfig
...

...
/src/java/GlassBR/Calculations.java
/src/java/Makefile
/src/java/README.md
...
/src/cpp/GlassBR
/src/cpp/ReadTable.cpp
/src/cpp/InputFormat.hpp
/src/cpp/Calculations.cpp
...
/src/swift/Calculations.swift
...
/src/csharp/Control.cs
...

/glassbr
/Website/GlassBR_SRS.html
/Website/GlassBR_SRS.css
/SRS/bibfile.bib
/SRS/Makefile
/SRS/GlassBR_SRS.tex
/SRS/GlassBR_SRS.pdf
/src/python
/src/python/README.md
/src/python/InputParameters.py
/src/python/Calculations.py
/src/python/Makefile
/src/python/doxConfig
...

...
/src/java/GlassBR/Calculations.java
/src/java/Makefile
/src/java/README.md
...
/src/cpp/GlassBR
/src/cpp/ReadTable.cpp
/src/cpp/InputFormat.hpp
/src/cpp/Calculations.cpp
...
/src/swift/Calculations.swift
...
/src/csharp/Control.cs
...

tex

Software Requirements Specification for GlassBR

Nikitha K and Spencer S

html

Table of Symbols

$\hat{q}_{\text{tol}}$

B

...

Introduction

... The software, herein called GlassBR, ...

Assumptions

ldfConstant: LDF is constant, depends on assumed value of t_d and m, ...

Data Definitions

$$B = \frac{k}{(ab)^{m-1}} \left(Eh^2\right)^m \text{LDF} e^J$$

...

README.md

GlassBR

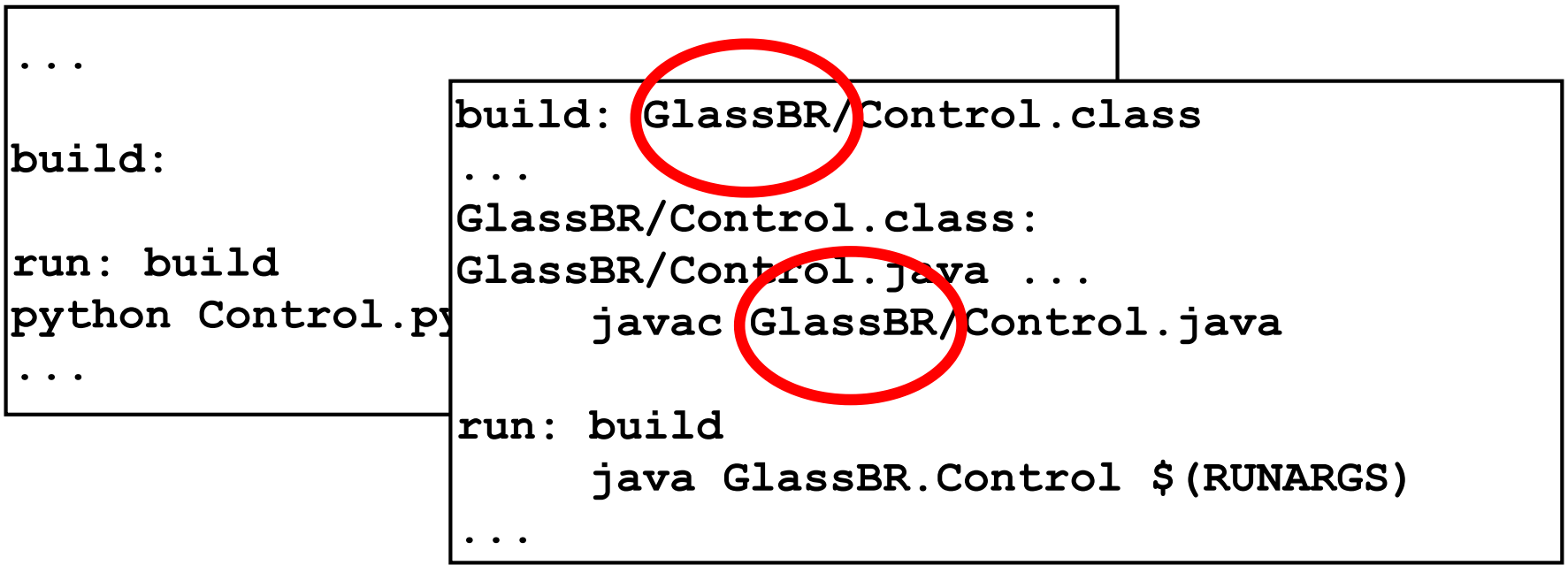
Authors Nikitha K and Spencer S

How to Run the Program: In your terminal command line, enter the same directory as this README file. Then enter the following line

```
make run RUNARGS=input.txt
```

Configuration Files: SDF.txt, TSD.txt must be in the same directory as the executable to run successfully

Versioning: Python Version 3.5.1


...
build:
run: build
python Control.py
...
build: GlassBR/Control.class
...
GlassBR/Control.class:
GlassBR/Control.java ...
javac GlassBR/Control.java
run: build
java GlassBR.Control $(RUNARGS)
...

Calculations.py

```
## \file Calculations.py
# \author Nikitha Krithnan and W. Spencer Smith
# \brie
...
## \bri
# \para
# \para
# \retu
def fur
    out
    pri
    ...
    out

    ret
```

Calculations.java

```
package GlassBR;
/** \file Calculations.java
    \author Nikitha Krithnan and W. Spencer Smith
    \brief Provides functions for calculating the outputs
*/
...
    public static double func_B(InputParameters inParams, double J) throws IOException {
      PrintWriter outfile;
      outfile = new PrintWriter(new FileWriter(new File("log.txt"), true));
      outfile.println("function func_B called with inputs: {");
      ...
      outfile.close();

      return 2.86e-53 /Math.pow(inParams.a * inParams.b, 7.0 - 1.0) *
             Math.pow(7.17e10 * Math.pow(inParams.h, 2.0), 7.0) * inParams.LDF
             * Math.exp(J);
    }
```

J_{tol} in SRS.pdf

Refname	**DD:sdfTol**
Label	Stress distribution factor (Function) based on Pbtol
Symbol	J_{tol}
Units	Unitless
Equation	$$J_{\text{tol}} = \ln\left(\ln\left(\frac{1}{1-P_{\text{btol}}}\right)\frac{\left(\frac{a}{1000}\frac{b}{1000}\right)^{m-1}}{k\left(E\cdot 1000\left(\frac{h}{1000}\right)^2\right)^m LDF}\right)$$
Description	J_{tol} is the stress distribution factor (Function) based on Pbtol (Unitless) P_{btol} is the tolerable probability of breakage (Unitless) a is the plate length (long dimension) (m) b is the plate width (short dimension) (m) m is the surface flaw parameter $(\frac{\text{m}^{12}}{\text{N}^7})$ k is the surface flaw parameter $(\frac{\text{m}^{12}}{\text{N}^7})$ E is the modulus of elasticity of glass (Pa) h is the minimum thickness (m) LDF is the load duration factor (Unitless)

J_{tol} in SRS.tex

```
...
Label & Stress distribution factor (Function) based on Pbtol

\\ \midrule \\
Symbol & ${J_{\text{tol}}}$

\\ \midrule \\
Units & Unitless

\\ \midrule \\
Equation & \begin{displaymath}
           {J_{\text{tol}}}=\ln\left(\ln\left(\frac{1}{1-{P_
               {\text{b}\text{tol}}}}\right) \frac{\left(\
               frac{a}{1000} \frac{b}{1000}\right)^{m-1}}{k \
               left(E\cdot{}1000 \left(\frac{h}{1000}\right)
               ^{2}\right)^{m} LDF}\right)
           \end{displaymath}
\\ \midrule \\
Description & ...
```

J_{tol} in SRS.html

```
...
<th>Equation</th>
<td>
\[{J_{\text{tol}}}=\ln\left(\ln\left(\frac{1}{1-{P_{\text{b}}\
    text{tol}}}}\right) \frac{\left(\frac{a}{1000} \frac{b
    }{1000}\right)^{m-1}}{k \left(E\cdot{}1000 \left(\frac{h
    }{1000}\right)^{2}\right)^{m} LDF}\right)\]
</td>
...
```

J_{tol} in Python

```
## \brief Calculates stress distribution factor (Function)
    based on Pbtol
# \param inParams structure holding the input values
# \return stress distribution factor (Function) based on
    Pbtol
def func_J_tol(inParams):
    outfile = open("log.txt", "a")
    print("function func_J_tol called with inputs: {", file=
        outfile)
    print("  inParams = ", end="", file=outfile)
    print("Instance of InputParameters object", file=outfile)
    print("  }", file=outfile)
    outfile.close()

    return math.log(math.log(1.0 / (1.0 - inParams.P_btol)) *
         ((inParams.a / 1000.0 * (inParams.b / 1000.0)) **
        (7.0 - 1.0) / (2.86e-53 * (7.17e10 * 1000.0 * (
        inParams.h / 1000.0) ** 2.0) ** 7.0 * inParams.LDF)))
```

J_{tol} in Java

```
/** \brief Calculates stress distribution factor (
    Function) based on Pbtol
    \param inParams structure holding the input values
    \return stress distribution factor (Function) based
        on Pbtol
*/
public static double func_J_tol(InputParameters inParams)
     throws IOException {
    PrintWriter outfile;
    outfile = new PrintWriter(new FileWriter(new File("
        log.txt"), true));
    ...
    return Math.log(Math.log(1.0 / (1.0 - inParams.P_btol
        )) * (Math.pow(inParams.a / 1000.0 * (inParams.b
        / 1000.0), 7.0 - 1.0) / (2.86e-53 * Math.pow(7.17
        e10 * 1000.0 * Math.pow(inParams.h / 1000.0, 2.0)
        , 7.0) * inParams.LDF)));
}
```

J_{tol} in Drasil (Haskell)

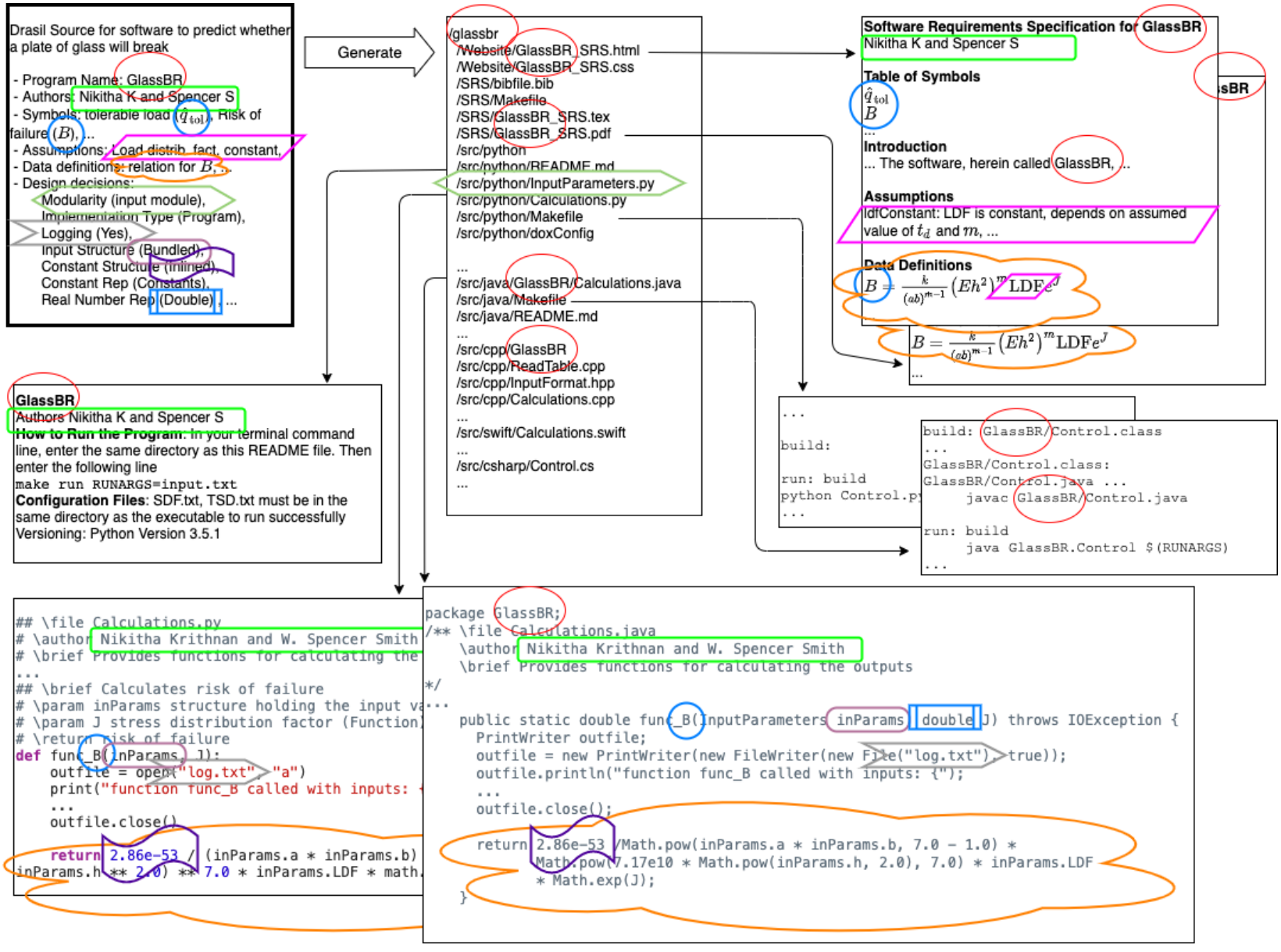
```
tolStrDisFacEq :: Expr
tolStrDisFacEq = ln (ln (recip_ (exactDbl 1 $- sy pbTol))
  `mulRe` (((sy plateLen $/ exactDbl 1000) `mulRe` (sy
     plateWidth $/ exactDbl 1000)) $^ (sy sflawParamM $-
     exactDbl 1) $/
    (sy sflawParamK `mulRe` ((sy modElas `mulRe` exactDbl
       1000 `mulRe`
    square (sy minThick $/ exactDbl 1000)) $^ sy sflawParamM)
         `mulRe` sy lDurFac)))
```

J_{tol} without Unit Conversion

```
tolStrDisFacEq :: Expr
tolStrDisFacEq = ln (ln (recip_ (exactDbl 1 $- sy pbTol))
  `mulRe` ((sy plateLen `mulRe` sy plateWidth) $^ (sy
     sflawParamM $- exactDbl 1) $/
    (sy sflawParamK `mulRe` ((sy modElas `mulRe`
    square (sy minThick)) $^ sy sflawParamM) `mulRe` sy
       lDurFac)))
```

Drasil Inputs:

- Program Name: GlassBR
- Authors: Nikitha K and Spencer S
- Symbols: tolerable load ($\hat{q}_{\text{tol}}$), Risk of failure (B), ...
- Assumptions: Load duration factor constant,
- Data definitions: relation for B, ...
- Design decisions:
    - Modularity (input module),
    - Implementation Type (Program),
    - Logging (Yes),
    - Input Structure (Bundled),
    - Constant Structure (Inlined),
    - Constant Rep (Constants),
    - Real Number Rep (Double),
    - ...


Drasil Source for software to predict whether a plate of glass will break
- Program Name: GlassBR
- Authors: Nikitha K and Spencer S
- Symbols: tolerable load ($\hat{q}_{tol}$), Risk of failure (B), ...
- Assumptions: Load distrib. fact. constant, ...
- Data definitions: relation for B, ...
- Design decisions:
Modularity (input module),
Implementation Type (Program),
Logging (Yes),
Input Structure (Bundled),
Constant Structure (Inlined),
Constant Rep (Constants),
Real Number Rep (Double), ...
Generate
/glassbr
/Website/GlassBR_SRS.html
/Website/GlassBR_SRS.css
/SRS/bibfile.bib
/SRS/Makefile
/SRS/GlassBR_SRS.tex
/SRS/GlassBR_SRS.pdf
/src/python
/src/python/README.md
/src/python/InputParameters.py
/src/python/Calculations.py
/src/python/Makefile
/src/python/doxConfig
...
/src/java/GlassBR/Calculations.java
/src/java/Makefile
/src/java/README.md
...
/src/cpp/GlassBR
/src/cpp/ReadTable.cpp
/src/cpp/InputFormat.hpp
/src/cpp/Calculations.cpp
...
/src/swift/Calculations.swift
...
/src/csharp/Control.cs
...
Software Requirements Specification for GlassBR
Nikitha K and Spencer S
Table of Symbols
$\hat{q}_{tol}$
B
...
Introduction
... The software, herein called GlassBR, ...
Assumptions
ldfConstant: LDF is constant, depends on assumed value of t_d and m, ...
Data Definitions
$B = \frac{k}{(ab)^{m-1}} (Eh^2)^m \text{LDF} e^J$
$B = \frac{k}{(ab)^{m-1}} (Eh^2)^m \text{LDF} e^J$
...
GlassBR
Authors Nikitha K and Spencer S
How to Run the Program: In your terminal command line, enter the same directory as this README file. Then enter the following line
make run RUNARGS=input.txt
Configuration Files: SDF.txt, TSD.txt must be in the same directory as the executable to run successfully
Versioning: Python Version 3.5.1
...
build:
run: build
python Control.py
...
build: GlassBR/Control.class
...
GlassBR/Control.class:
GlassBR/Control.java ...
javac GlassBR/Control.java
run: build
java GlassBR.Control $(RUNARGS)
...
\file Calculations.py
\author Nikitha Krithnan and W. Spencer Smith
\brief Provides functions for calculating the
...
\brief Calculates risk of failure
# \param inParams structure holding the input va
# \param J stress distribution factor (Function)
# \return risk of failure
def func_B(inParams, J):
outfile = open("log.txt", "a")
print("function func_B called with inputs:
...
outfile.close()
return 2.86e-53 / (inParams.a * inParams.b)
inParams.h ** 2.0) ** 7.0 * inParams.LDF * math.
package GlassBR;
/** \file Calculations.java
\author Nikitha Krithnan and W. Spencer Smith
\brief Provides functions for calculating the outputs
*/
...
public static double func_B(InputParameters inParams, double J) throws IOException {
PrintWriter outfile;
outfile = new PrintWriter(new FileWriter(new File("log.txt"), true));
outfile.println("function func_B called with inputs: {");
...
outfile.close();
return 2.86e-53 /Math.pow(inParams.a * inParams.b, 7.0 - 1.0) *
Math.pow(7.17e10 * Math.pow(inParams.h, 2.0), 7.0) * inParams.LDF
* Math.exp(J);
}

Improve Software Qualities

- Capture best practices
- Explore alternatives
- Traceability
- Reproducibility
- Sustainability
- Certifiability
- Reusability

Concluding Remarks

- What Drasil can currently do
  - Explicit relations, LHS = ...
  - Linear first order ODEs
  - Knowledge on rigid body mechanics, heat transfer
  - SRS (LaTeX, html), code (Python, C++, C sharp, Java, Swift), README, Makefile
- Future additions
  - More scientific knowledge: medical imaging, chemistry, more mechanics, etc.
  - More computational knowledge: external libraries, linear systems solvers, higher order ODEs, root finding, etc.
  - More document knowledge: Teaching lessons, academic papers, assurance cases, etc.
  - Jupyter notebooks
  - GitHub Issues