

Summary Report for the SINBAD Search Tool Project

June 2012

Prepared by
Alice Cunha da Silva, Federal University of Rio de Janeiro (UFRJ)

Work performed as part of the author's two-month practicum assignment at ORNL under mentorship by L. C. Leal through the ORNL Nuclear Science and Engineering Directorate Educational Outreach Program. Financial support for the author provided by UFRJ.

DOCUMENT AVAILABILITY

Reports produced after January 1, 1996, are generally available free via the U.S. Department of Energy (DOE) Information Bridge.

Web site <http://www.osti.gov/bridge>

Reports produced before January 1, 1996, may be purchased by members of the public from the following source.

National Technical Information Service
5285 Port Royal Road
Springfield, VA 22161
Telephone 703-605-6000 (1-800-553-6847)
TDD 703-487-4639
Fax 703-605-6900
E-mail info@ntis.gov
Web site <http://www.ntis.gov/support/ordernowabout.htm>

Reports are available to DOE employees, DOE contractors, Energy Technology Data Exchange (ETDE) representatives, and International Nuclear Information System (INIS) representatives from the following source.

Office of Scientific and Technical Information
P.O. Box 62
Oak Ridge, TN 37831
Telephone 865-576-8401
Fax 865-576-5728
E-mail reports@osti.gov
Web site <http://www.osti.gov/contact.html>

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

Reactor and Nuclear Systems Division

Summary Report for The SINBAD Search Tool Project

Alice Cunha da Silva, Federal University of Rio de Janeiro

Work performed as part of the author's two-month practicum assignment at ORNL under mentorship by L. C. Leal through the ORNL Nuclear Science and Engineering Directorate Educational Outreach Program. Financial support for the author provided by UFRJ.

Date Published: June 2012

Prepared by
OAK RIDGE NATIONAL LABORATORY
P.O. Box 2008
Oak Ridge, Tennessee 37831-6285
managed by
UT-BATTELLE, LLC
for the
U.S. DEPARTMENT OF ENERGY
under contract DE-AC05-00OR22725

CONTENTS

| | Page |
|---|-------------|
| LIST OF FIGURES | v |
| ACKNOWLEDGMENTS | vii |
| ABSTRACT..... | ix |
| 1. INTRODUCTION..... | 11 |
| 1.1 EXISTING SEARCH PROCEDURE | 11 |
| 1.2 NEW SEARCH TOOL..... | 11 |
| 1.3 THE INTEGRATED DEVELOPMENT ENVIRONMENT (IDE) | 11 |
| 1.4 PROCEDURES | 11 |
| 2. SUMMARY AND FUTURE WORK..... | 19 |
| 3. REFERENCES..... | 21 |
| APPENDIX A : EXAMPLE OF THE PREVIOUS SEARCH METHOD | A-1 |
| APPENDIX B : LIST OF THE SINBAD SEARCH TOOL CODE, WRITTEN IN JAVA LANGUAGE ON THE NetBeans IDE 7.1 PLATFORM..... | B-1 |

LIST OF FIGURES

| | Page |
|------------|--|
| Fig. 1. | Front page of the SINBAD Search Tool..... 12 |
| Fig. 2. | Materials. 13 |
| Fig. 3. | Searching results. 14 |
| Fig. 4. | Example of a benchmark page..... 15 |
| Fig. 5. | Input link..... 16 |
| Fig. 6. | Input page. 17 |
| Fig. A.1. | Previous search tool. A-3 |
| Fig. A.2. | Previous search tool 2. A-4 |
| Fig. A.3. | Previous search tool 3. A-5 |
| Fig. A.4. | Previous search tool 4. A-6 |
| Fig. A.5. | Previous search tool 5. A-7 |
| Fig. A.6. | Previous search tool 6. A-8 |
| Fig. A.7. | Previous search tool 7. A-9 |
| Fig. A.8. | Previous search tool 8. A-10 |
| Fig. A.9. | Previous search tool 9. A-11 |
| Fig. A.10. | Previous search tool 10. A-12 |
| Fig. A.11. | Previous search tool 11. A-13 |
| Fig. A.12. | Previous search tool 12. A-14 |

ACKNOWLEDGMENTS

This work was performed as part of an educational practicum assignment with the Nuclear Data and Criticality Safety Group of the Reactor and Nuclear Systems Division of the Oak Ridge National Laboratory (ORNL). The author is particularly indebted to Luiz Leal and Mike Dunn (ORNL) and Su Jian (Federal University of Rio de Janeiro) for their support. The author would also like to thank Robert Lefebvre and Dorothea Wiarda (ORNL) for their invaluable help.

ABSTRACT

The Shielding Integral Benchmark Archive Database (SINBAD) Search Tool has been developed to serve as an interface with the SINBAD database to facilitate a simple and quick means of searching for information related to experimental benchmark problems. The Search Tool is written in Java and provides a better and efficient way to retrieve information from the SINBAD database. Searches can be performed quickly and easily. With regard to improvements, users are no longer required to know the name of the benchmarks to search the database. Instead, a search can be performed by specifying the experimental facility, constituents of the experimental benchmark, etc. In summary, a new powerful database search tool has been developed for SINBAD.

1. INTRODUCTION

The Shielding Integral Benchmark Archive Database^{1,2} (SINBAD) includes benchmark experiments relevant to reactor shielding, fusion, and accelerator shielding applications. The database stores and makes available information related to high-quality benchmark experiments, which is useful for validating radiation transport codes and nuclear data. Several benchmark experiments have been incorporated into SINBAD. Every year new experiments are proposed for inclusion in the SINBAD database. The Radiation Safety Information Computational Center (RSICC)³ at Oak Ridge National Laboratory (ORNL) is responsible for collecting, evaluating, and distributing computer codes and databases pertinent to radiological study, including SINBAD.

The purpose of this work is to describe a tool, the SINBAD Search Tool that can be used for the simple and quick selection of documents and benchmarks resident in the SINBAD database.

1.1 EXISTING SEARCH PROCEDURE

The SINBAD system distributed by RSICC includes a very limited search tool that was developed in the HTML language. To find needed information with the existing tool, one must know the precise name of the benchmark (i.e., the user will have to look at each name of the benchmarks in the database). An example of the use of this tool is given in Appendix A.

1.2 NEW SEARCH TOOL

To improve the searching procedure of information in the SINBAD database, a new tool has been developed in the programming language Java.⁴ This language offers several features including the ability to execute procedures in various operational systems. After examining the structure of the SINBAD database, it was decided that the Java language was suitable for the project.

1.3 THE INTEGRATED DEVELOPMENT ENVIRONMENT (IDE)

For this project the NetBeans IDE 7.1⁵ was chosen because it is a free open-source Integrated Development Environment for software development, with all the tools needed to create professional desktop, enterprise, web, and mobile applications with the Java platform.

1.4 PROCEDURES

The search tool developed in this work consists of a graphical user interface (GUI) that permits the users to choose from a frame of options the information that better qualifies the benchmark search. As a result, the SINBAD Search Tool returns a list of benchmark names that matches the user's request. Examples of the search procedure using the search tool are displayed in Figs.1–6. The first page of the search tool is shown in Fig. 1. As can be seen, there are four boxes labeled Type, Material, Laboratory, and Geometry. Descriptions of each parameter are as follows:

Type: The user chooses the type of benchmark application, namely, Reactor Shielding, Fusion Neutronic Shielding, or Accelerator Shielding, of interest. If “All” is selected, all three options are retrieved simultaneously.

Material: The user provides the name of the material of interest. More than one material may be specified as shown in Fig. 2. This is the only field in which the user can choose more than one option. The method of doing so is operating-system dependent. On a Windows machine, the user must press the “Ctrl” key while pressing the second, third, and other choices. Selecting “Any” retrieves all options simultaneously.

Laboratory: The user provides the name of the laboratory responsible for performing the benchmark experiments of interest. Selecting “All” retrieves all options simultaneously.

Geometry: The user specifies the type of geometry of interest. Selecting “Any” retrieves all options simultaneously.

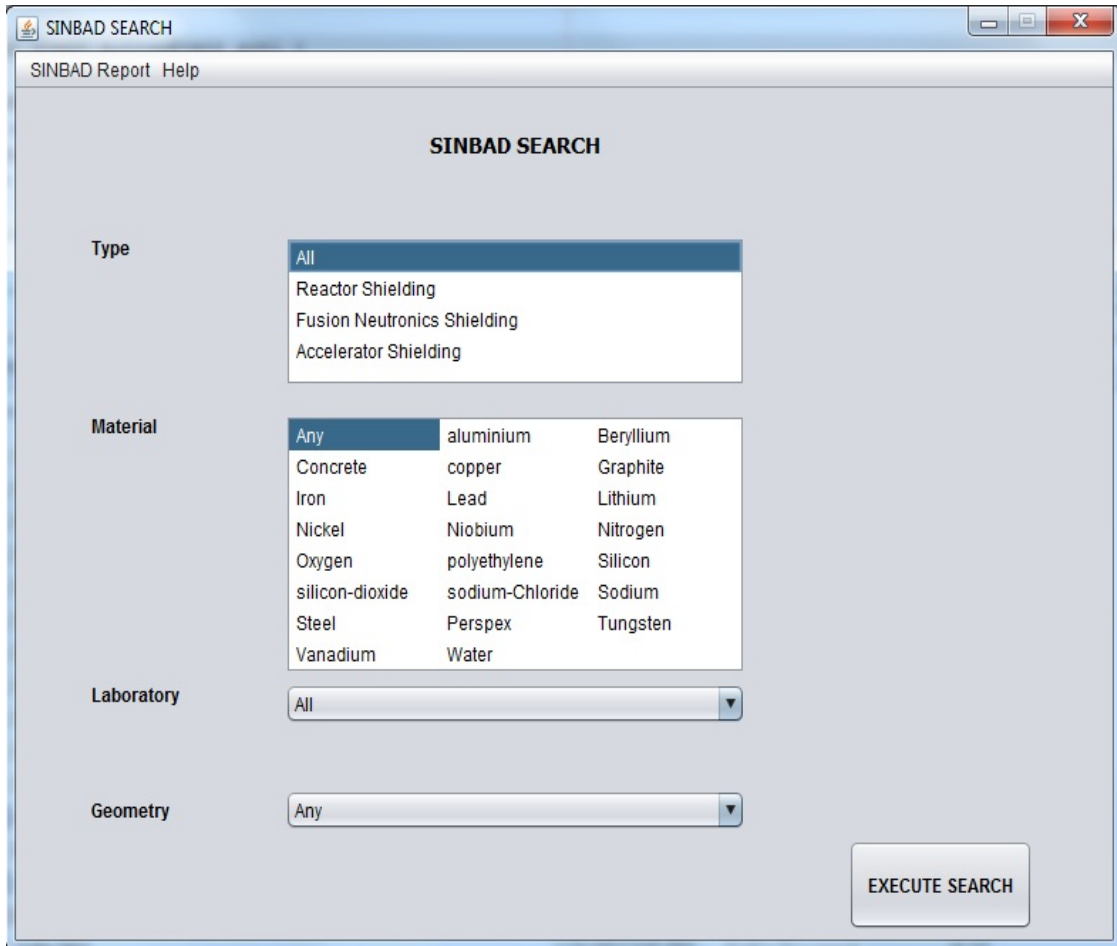


Fig. 1. Front page of the SINBAD Search Tool.

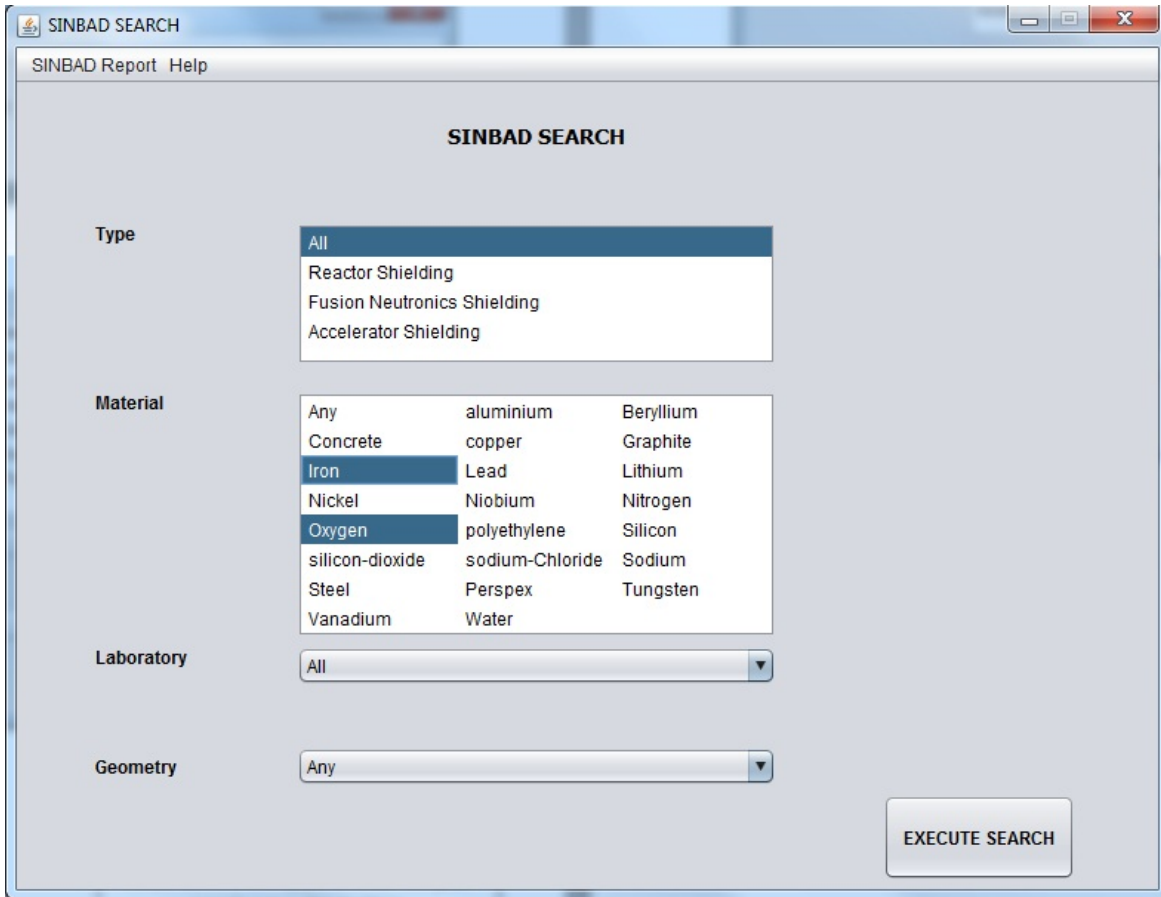


Fig. 2. Materials.

After the search parameters are defined and the “Execute Search” button is pressed, a frame that lists the benchmark names that are well matched with the user’s requested choices is presented as shown in Fig. 3.

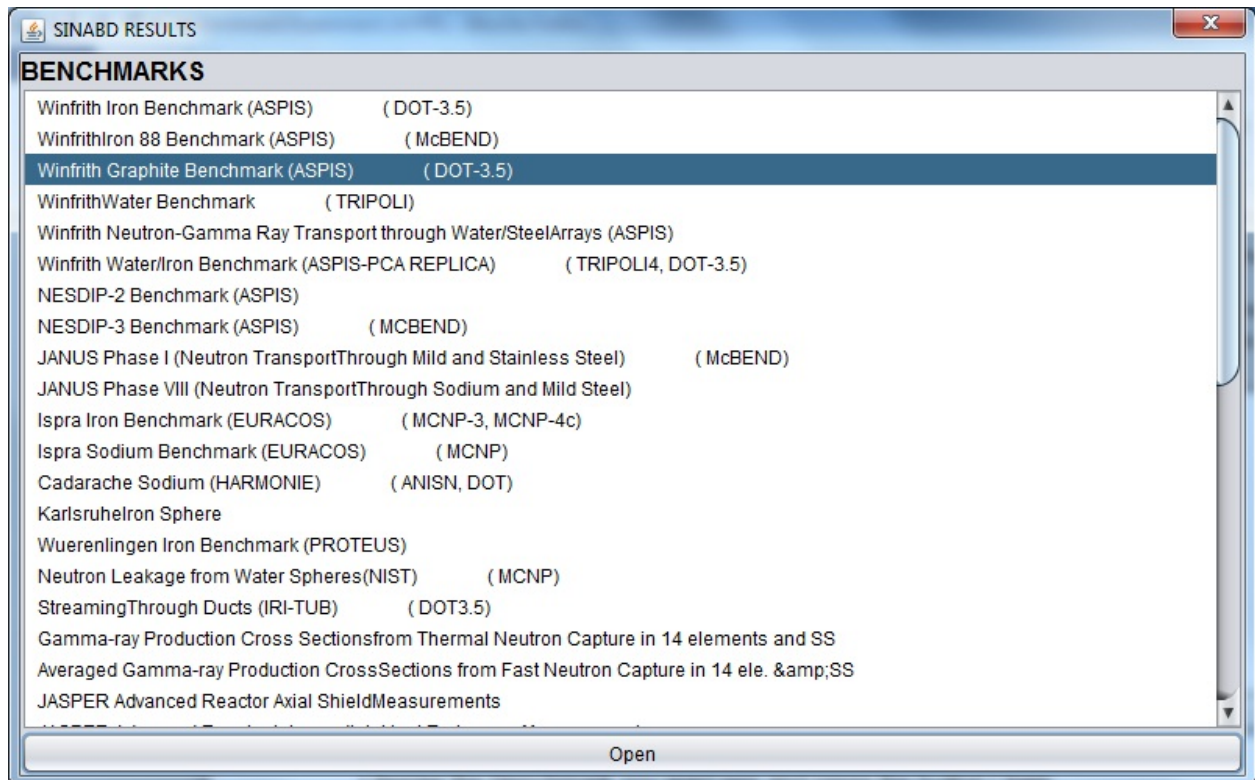


Fig. 3. Searching results.

For some benchmark experiments present in the SINBAD database, also included is an input description for some computer codes. Input data for more than one code system may exist for a single benchmark. If so, this information will be displayed in parentheses beside the benchmark name. This feature is shown in Fig. 3.

After the search is complete, the user may be able to further refine the search and choose the specific benchmark experiment needed. This is done by selecting the benchmark experiment inquired and pressing the button "Open." The benchmark abstract will open in the default browser as shown in Fig. 4.

Any available benchmark computer code input is indicated on the page with the extension .INP as shown in Fig. 5. As an example, the input to the code DOT is specified by "dot35.INP." The contents can be displayed by clicking on "dot35.INP," which will open a page in the default browser of the user's computer as shown in Fig. 6.

SINBAD ABSTRACT NEA-1517/36

Winfrith Graphite Benchmark Experiment (ASPIS)

1. Name of Experiment:

Winfrith Graphite Benchmark Experiment (ASPIS)

2. Purpose and Phenomena Tested:

Determination of the accuracy of methods used to calculate the neutron component of nuclear heating. Threshold reaction rates were measured up to 0.7 m in graphite.

3. Description of the Source and Experimental Configuration:

The source is a fission plate constructed of 93% enriched uranium aluminium alloy driven by a thermal flux from the extended graphite reflector of the NESTOR reactor. The energy spectrum of the source is the one of neutrons emitted from the fission of U-235. The absolute source strength is determined by fission product counting and the spatial distribution via detailed low energy flux mapping with activation detectors.
The graphite assembly had lateral dimensions 180 cm x 190 cm and total length was 177.32 cm. It was built from graphite block of various sizes. The concrete of the approximate thickness 76 cm encases the whole assembly. The detectors were placed in the central block in the cylindrical plug, inserted in 6.45 cm radius hole along the major axis of the block.

4. Measurement System and Uncertainties:

The detectors used were:

| Detector | Diameter (mm) | Thickness (mm) | Mass (g) | Counting System | Systematic Error (%) |
|----------------|------------------|-------------------|-------------|--------------------|-------------------------|
| Al-27(n,alpha) | 50 | 3.1 | 16.72 | Ge detector | 4.0 |
| S-32(n,p) | 38.1 | 2.41 | 5 | Plastic scint. | 4.0 |
| In-115(n,n') | 38 | 1.63 | 12.79 | GeLi detector | 3.0 |
| Rh-103(n,n') | 12.7 | 0.015 | 0.20 | NaI | 3.0 |

The uncertainties (1 sigma) may be taken as uncorrelated, and derive essentially from the absolute calibration of the counting system.

Fig. 4. Example of a benchmark page.

9. References:

- [1] M.D. Carter, P.C. Miller, A. Packwood,
The ASPIS Graphite Benchmark Experiment. Part 1 - Experimental Data
and Preliminary Results, NEACRP-A-630 (1984)
- [2] N. Sasamoto, K. Sakurai, A. Tsubosaka, H. Narita, M. Takemura,
K. Hayashi, Analysis of the ASPIS Graphite Benchmark Experiment with
Discrete Ordinates and Monte Carlo Codes,
NEACRP Specialists' Meeting on Shielding Benchmarks, Paris (1986).
- [3] Alan F. Avery, AEA-RS-5628, private communication.
- [4] G. A. Wright, A. Avery, M. J. Grimstone, H. F. Locke, S. Newbon,
Benchmarking of the JEFF2.2 Data Library for Shielding Applications,
Proceedings, 8th International Conference on Radiation Shielding,
April 24-28, 1994, Arlington, Texas, U.S.A., vol.2, p.816.

10. Data and Format:

| Filename | Size[bytes] | Content |
|--------------------------------|-------------|--|
| 1 ASC-ABS.HTM | 7.462 | This information file. |
| 2 ASC-EXP.HTM | 20.668 | Description of Experiment. |
| 3 dot35.INP | 4.684 | Input data for DOT-3.5 2-D Model Sn code . |
| 4 ASPC-1V.TIF | 27.464 | Figure 1: Cross-section of the ASPIS graphite benchmark experiment. (high quality) |
| 5 ASPC-2V.TIF | 17.362 | Figure 2: Detail of the fission plate. (high quality) |
| 6 ASPC-3V.TIF | 17.696 | Figure 3: Section through the ASPIS graphite benchmark model. (high quality) |
| 7 ASPC-4V.TIF | 16.136 | Figure 4: RZ geometry of the ASPIS graphite benchmark for the DOT 3.5 calculation. (high quality) |
| 8 ASPC-1V.gif | 20.266 | Figure 1: Cross-section of the ASPIS graphite benchmark experiment. (preview) |
| 9 ASPC-2V.gif | 12.183 | Figure 2: Detail of the fission plate. (preview) |
| 10 ASPC-3V.gif | 14.797 | Figure 3: Section through the ASPIS graphite benchmark model. (preview) |
| 11 ASPC-4V.gif | 11.482 | Figure 4: RZ geometry of the ASPIS graphite benchmark for the DOT 3.5 calculation. (preview) |
| 12 WIN C 1.pdf | 2.237.854 | Reference |
| 13 WIN C 2.pdf | 458.475 | Reference |
| 14 WIN C 3.pdf | 789.823 | Reference |
| 15 WIN C 4.pdf | 578.980 | Reference |

Figures describing the geometry of the experiment are included in
TIFF file format and GIF (preview) format.

Tables in [Asc-exp.htm](#):
one table of axial dimensions and of chemical composition of materials
present in the configuration, 3 tables describing the source, and
4 tables of the measured reaction rates.

Fig. 5. Input link.

IV. Input data for the DOT 3.5 calculation, using the WIMS 2 fission spectrum, for the ASPIS graphite benchmark. Taken from [1], simplified geometry as shown on Fig.4.

```

-----
DOT - MC COMPARISON FOR ASPIS GRAPHITE BENCHMARK
O
61$$
/  A02 A03 IZM  IM  JM  IGM  IHT  IHS  ITL  M01
   0  3  10  25  37  120  3  4  123  0
/
/  MCR MTP  MT  IPS  A04  IGE  B01  B02  B04  B03
   25  0  25  1  48  1  1  0  0  0
/
/  D05 S04 G07  FXT  I04  IP1  S02  IZ  JZ  IZC
   1  1  100  0  0  2  0  0  0  1
/
/  IB4 ISC  IZ3  M07  M06  IZ1  IZ2  IB5  IB6  IZ4
   0  0  0  5  3  0  0  0  0  0
/
/  IB2 M05  IB1  IP3  IAFT  IP4  IS2  IS3  IZ5  IZ6
   1  5  7  2R 0  5R 0
/
/  IMG IP2  IB3  ITI  IFLUX  IGMI  IA03I  IA04I  ISRCE  IGIXS  IPRT
   2  20  4  0  22  120  0  0  F  0
/
62$$
8R 0 2 5R 0
63**
7.60587E+10 0.0001 0.0 0.0 F 0.0
T
/  SYMMETRIC S 8 QUADRATURE SET
/  DIRECTION COSINES (MU-ETA) (48+48)
7*
0 -30861- 5 0 -21822- 5 0 +21822- 5 0 -61721- 5 0 -57735- 5 0 -21822- 5 1
0 +21822- 5 0 +57735- 5 0 -81650- 5 0 -78680- 5 0 -57735- 5 0 -21822- 5 2
0 +21822- 5 0 +57735- 5 0 +78680- 5 0 -97590- 5 0 -95119- 5 0 -78680- 5 3
0 -57735- 5 0 -21822- 5 0 +21822- 5 0 +57735- 5 0 +78680- 5 0 +95119- 5 4
0 -30861- 5 0 -21822- 5 0 +21822- 5 0 -61721- 5 0 -57735- 5 0 -21822- 5 5
0 +21822- 5 0 +57735- 5 0 -81650- 5 0 -78680- 5 0 -57735- 5 0 -21822- 5 6
0 +21822- 5 0 +57735- 5 0 +78680- 5 0 -97590- 5 0 -95119- 5 0 -78680- 5 7
0 -57735- 5 0 -21822- 5 0 +21822- 5 0 +57735- 5 0 +78680- 5 0 +95119- 5 8
3R-95119- 5 5R-78680- 5 7R-57735- 5 9R-21822- 5 3R+95119- 5 5R+78680- 5 1
7R+57735- 5 9R+21822- 5 2
T
/  QUADRATURE WEIGHTS (S8-48 ANGLES)
6*
0 + 0+ 0 2R+30247- 6 0 + 0+ 0 4R+22685- 6 0 + 0+ 0 0 +22685- 6 1
0 +23148- 6 2R+22685- 6 0 +23148- 6 0 +22685- 6 0 + 0+ 0 0 +30247- 6 2
0 +22685- 6 0 +22685- 6 2R+30247- 6 0 +22685- 6 0 +22685- 6 0 +30247- 6 3
0 + 0+ 0 2R+30247- 6 0 + 0+ 0 4R+22685- 6 0 + 0+ 0 0 +22685- 6 4
0 +23148- 6 2R+22685- 6 0 +23148- 6 0 +22685- 6 0 + 0+ 0 0 +30247- 6 5
0 +22685- 6 0 +22685- 6 2R+30247- 6 0 +22685- 6 0 +22685- 6 0 +30247- 6 6
T

```

Fig. 6. Input page.

2. SUMMARY AND FUTURE WORK

To facilitate searches for information in the SINBAD database, a new tool written in the Java language has been developed. This tool, called the SINBAD search tool, provides a better and efficient way to retrieve information from the SINBAD database. Searches can be performed quickly and easily. The users are no longer required to know the name of the benchmarks for which they are searching. Instead, the search is carried out by requiring information such as the facility where an experiment was done, the constituents of the experimental benchmark, and so on.

In future versions of the SINBAD search tool, features can be added to improve the program:

- The HTML files that contain the benchmarks abstracts could be standardized.
- An interface to run a code using the .INP files could be provided (e.g., add the ability to run MCNP code when the corresponding .INP for MCNP is given for the benchmark in the SINBAD database).
- Additional types of geometry could be added to the geometry field in the Sinbad Search interface (e.g., SLAB).
- More categories of incident particles (neutrons, electrons, and protons) could be added.
- Energy range could be added as a search criterion.
- Presently only three types of benchmark categories exist (reactor shielding, accelerator shielding, and fusion neutronic shielding), but the capability to facilitate the addition of more benchmark categories could be added without the need to reprogram the source code.

3. REFERENCES

1. OECD Nuclear Energy Agency Data Bank SINBAD,
<http://www.nea.fr/html/science/shielding/sinbad/sinbadis.htm>.
2. I. Kodeli, E. Sartori, and B. Kirk. "SINBAD Shielding Benchmark Experiments Status and Planned Activities," American Nuclear Society 14th Biennial Topical Meeting of the Radiation Protection and Shielding Division, Carlsbad, New Mexico, USA, April 3–6, 2006.
3. Radiation Safety Information Computational Center (RSICC),
<http://www-rsicc.ornl.gov/rsiccnew/BENCHMARKS.htm>.
4. Java.sun.com.
5. NetBeans Integrated Development Environment – IDE, <http://netbeans.org/index.html>.

**APPENDIX A:
EXAMPLE OF THE PREVIOUS SEARCH METHOD**

SINBAD INDEX - Sorted by Shielding

Reactor Shielding (45)

[Winfrith Iron Benchmark \(ASPIS\)](#)
[Winfrith Iron 88 Benchmark \(ASPIS\)](#)
[Winfrith Graphite Benchmark \(ASPIS\)](#)
[Winfrith Water/Iron Benchmark \(ASPIS-PCA REPLICA\)](#)
[Winfrith Water Benchmark](#)
[Winfrith Neutron-Gamma Ray Transport through Water/Steel Arrays \(ASPIS\)](#)
[NESDIP-2 Benchmark \(ASPIS\)](#)
[NESDIP-3 Benchmark \(ASPIS\)](#)
[JANUS Phase I \(Neutron Transport Through Mild and Stainless Steel\)](#)
[JANUS Phase VIII \(Neutron Transport Through Sodium and Mild Steel\)](#)
[Ispra Sodium Benchmark \(EURACOS\)](#)
[Ispra Iron Benchmark \(EURACOS\)](#)
[Cadarache Sodium \(HARMONIE\)](#)
[Karlsruhe Iron Sphere](#)
[Wuerenlingen Iron Benchmark \(PROTEUS\)](#)
[Neutron Leakage from Water Spheres \(NIST\)](#)
[Streaming Through Ducts \(IRI-TUB\)](#)
[Gamma-ray Production Cross Sections from Thermal Neutron Capture in 14 elements and SS](#)
[Averaged Gamma-ray Production Cross Sections from Fast Neutron Capture in 14 ele. & SS](#)
[JASPER Advanced Reactor Axial Shield Measurements](#)
[JASPER Advanced Reactor Intermediate Heat Exchanger Measurements](#)
[JASPER Advanced Reactor Radial Shield Measurements](#)
[ORNL TSF Iron Broomstick](#)
[ORNL TSF Oxygen Broomstick](#)
[ORNL TSF Nitrogen Broomstick](#)
[ORNL TSF Sodium Broomstick](#)
[ORNL TSF Stainless Steel Broomstick](#)
[ORNL Neutron Transport Through Iron and SS - Part I](#)
[ORNL Neutron Transport in Thick Sodium](#)
[Pool Critical Assembly-Pressure Vessel Facility Benchmark](#)
[University of Illinois Iron Sphere \(CF-252\)](#)
[University of Tokyo-YAYOI Iron Slab](#)
[Radiation field parameters for pressure vessel monitoring in NRI LR-0 VVER-440 reactor](#)
[Radiation field parameters for pressure vessel monitoring in NRI LR-0 VVER-1000 reactor](#)
[Balakovo-3 VVER-1000 Ex-vessel Neutron Dosimetry Benchmark](#)
[VENUS-3 LWR-PVS Benchmark](#)
[H.B. Robinson-2 Pressure Vessel](#)

Fig. A.1. Previous search tool.

[Photon Leakage Spectra from Al, Ti, Fe, Cu, Zr, Pb, U238 Spheres](#)
[Photon Spectra from H2O, SiO2 and NaCl](#)
[IPPE Th shell with 14 MeV and Cf-252 source neutrons](#)
[Baikal-1 Skyshine Benchmark Experiment](#)
[NAIÁDE 1 Graphite Benchmark \(60cm\)](#)
[NAIÁDE 1 Iron Benchmark \(60cm\)](#)
[NAIÁDE 1 Light Water Benchmark \(60cm\)](#)
[SNL Polyethylene-Reflected Plutonium Metal Spheres: Subcritical Neutron and Gamma Measurements](#)

Fusion Neutronics Shielding (29)

[Nickel Sphere \(OKTAVIAN\)](#)
[Iron Sphere \(OKTAVIAN\)](#)
[Aluminium Sphere \(OKTAVIAN\)](#)
[Silicon Sphere \(OKTAVIAN\)](#)
[Tungsten Sphere \(OKTAVIAN\)](#)
[FNS Experimental data for fusion neutronics benchmark](#)
[FNS Integral Experiment on Graphite Cylindrical Assembly](#)
[FNS Liquid Oxygen](#)
[FNS Vanadium Cube](#)
[FNS Tungsten](#)
[FNS Skyshine](#)
[FNS Dogleg Duct Streaming](#)
[FNG-SS Shield \(integral\)](#)
[FNG-ITER Blanket Bulk Shield \(integral\)](#)
[FNG/TUD ITER Blanket Bulk Shield \(spectra measurements\)](#)
[FNG-ITER Neutron Streaming \(integral\)](#)
[FNG-ITER Dose Rate Experiment](#)
[FNG Silicon Carbide \(integral\)](#)
[FNG/TUD Silicon Carbide \(spectra measurements\)](#)
[FNG Tungsten \(integral\)](#)
[FNG/TUD Tungsten \(spectra measurements\)](#)
[TUD Iron Slab Experiment](#)
[IPPE Vanadium Shells](#)
[IPPE Iron Shells](#)
[ORNL 14-MeV Neutron SS/Borated Poly Slab](#)
[University of Illinois Iron Sphere \(D-T\)](#)
[KANT Spherical Beryllium Shells](#)
[MEPhI empty slits streaming experiment](#)
[Juelich Li Metal Blanket Experiment](#)

Accelerator Shielding (23)

[Transmission Through Shielding Materials of Neutrons and Photons Generated by 52 MeV Protons](#)
[Transmission Through Shielding Materials of Neutrons and Photons Generated by 65 MeV Protons](#)
[Transmission of Medium Energy Neutrons Through Concrete Shields \(AVF Cyclotron\)](#)
[Neutron Production from Thick Targets of Carbon, Iron, Copper, and Lead by 30- and 52-MeV Protons](#)

Fig. A.2. Previous search tool 2.

[TIARA 40 and 65 MeV Neutron Transmission Through Iron, Concrete and Polyethylene](#)
[Radioactivity induced by GeV-Protons and Spallation Neutrons using AGS accelerator](#)
[Intermediate and High-Energy Accelerator Shielding Benchmarks](#)
[ROESTI I, II and III \(CERN\)](#)
[CERF Bonner Sphere Spectrometer Response to Charged Hadrons](#)
[CERF Radionuclide Production](#)
[CERF Residual Dose Rates](#)
[CERF Neutron Energy Spectra behind Shielding of a 120 GeV/c Hadron Beam Facility](#)
[CERN 200 and 400 GeV/c protons activation experiments](#)
[RIKEN Quasi-monoenergetic Neutron Field in 70-210 MeV Energy Range](#)
[KENS p-500 MeV shielding experiment using 4m Concrete at KEK](#)
[HIMAC experiments with He, C, Ne, Ar, Fe, Xe and Si ions on C, Al, Cu & Pb targets](#)
[HIMAC High energy Neutron \(<800 MeV\) Measurements in Iron](#)
[HIMAC High energy Neutron \(<800 MeV\) Measurements in Concrete](#)
[BEVALAC Experiment with Nb Ions on Nb & Al Targets](#)
[MSU experiment with He & C ions on Al target](#)
[Neutron Spectra Generated by 590-MeV Protons on a Thick Pb Target](#)
[ISIS Deep-Penetration Neutrons through Concrete and Iron Shields using p-800 MeV](#)
[Simulation of the lineal energy distribution of the energy deposition in biological cells, TEPC-FLUKA Comparison](#)

SINBAD INDEX - Sorted by Directory Name

[\[30_52mev \] Neutron Production from Thick Targets of Carbon, Iron, Copper, and Lead by 30- and 52-MeV Protons](#)
[\[52p \] Transmission Through Shielding Materials of Neutrons and Photons Generated by 52 MeV Protons](#)
[\[65p \] Transmission Through Shielding Materials of Neutrons and Photons Generated by 65 MeV Protons](#)
[\[ags \] Radioactivity induced by GeV-Protons and Spallation Neutrons using AGS accelerator](#)
[\[asp_fe \] Winfrith Iron Benchmark \(ASPIS\)](#)
[\[asp_fe88 \] Winfrith Iron 88 Benchmark \(ASPIS\)](#)
[\[asp_grap \] Winfrith Graphite Benchmark \(ASPIS\)](#)
[\[asp_h2o \] Winfrith Water Benchmark](#)
[\[asp_ng \] Winfrith Neutron-Gamma Ray Transport through Water/Steel Arrays \(ASPIS\)](#)
[\[avf75mev \] Transmission of Medium Energy Neutrons Through Concrete Shields \(AVF Cyclotron\)](#)
[\[balakovo \] Balakovo-3 VVER-1000 Ex-vessel Neutron Dosimetry Benchmark](#)
[\[berp_poly_2009 \] Polyethylene-Reflected Plutonium Metal Sphere: Subcritical Neutron and Gamma Measurements](#)
[\[bevalac \] BEVALAC Experiment with Nb Ions on Nb & Al Targets](#)
[\[cerf_ac5 \] CERF Radionuclide Production](#)
[\[cerf_bss \] CERF Bonner Sphere Spectrometer Response to Charged Hadrons](#)
[\[cerf_dr3 \] CERF Residual Dose Rates](#)
[\[cerf_sp \] CERF Neutron Energy Spectra behind Shielding of a 120 GeV/c Hadron Beam Facility](#)
[\[cern200 \] CERN 200 and 400 GeV/c protons activation experiments](#)

Fig. A.3. Previous search tool 3.

[\[eurac fe \] Ispra Iron Benchmark \(EURACOS\)](#)
[\[eurac na \] Ispra Sodium Benchmark \(EURACOS\)](#)
[\[FNG BLKT \] FNG-ITER Blanket Bulk Shield \(integral\)](#)
[\[fng_dose \] FNG-ITER Dose Rate Experiment](#)
[\[fng_str \] FNG-ITER Neutron Streaming \(integral\)](#)
[\[fng_sic \] FNG Silicon Carbide \(integral\)](#)
[\[fns \] FNS Experimental data for fusion neutronics benchmark](#)
[\[FNG SS \] FNG-SS Shield \(integral\)](#)
[\[fng_w \] FNG Tungsten \(integral\)](#)
[\[fns_c \] FNS Integral Experiment on Graphite Cylindrical Assembly](#)
[\[fns_duct \] FNS Dogleg Duct Streaming](#)
[\[fns_o \] FNS Liquid Oxygen](#)
[\[fns_sky \] FNS Skyshine](#)
[\[fns_v \] FNS Vanadium Cube](#)
[\[fns_w \] FNS Tungsten](#)
[\[harmo_na \] Cadarache Sodium \(HARMONIE\)](#)
[\[hbr2 \] H.B. Robinson-2 Pressure Vessel](#)
[\[himac \] HIMAC experiments with He, C, Ne, Ar, Fe, Xe and Si ions on C, Al, Cu & Pb targets](#)
[\[himac800fe \] HIMAC High energy Neutron \(<800 MeV\) Measurements in Iron](#)
[\[himac800conc \] HIMAC High energy Neutron \(<800 MeV\) Measurements in Concrete](#)
[\[iheas \] Intermediate and High-Energy Accelerator Shielding Benchmarks](#)
[\[ILL FE \] University of Illinois Iron Sphere \(Cf-252\)](#)
[\[ILL FE \] University of Illinois Iron Sphere \(D-T\)](#)
[\[ippe_v \] IPPE Vanadium Shells](#)
[\[ippe-fe \] IPPE Iron Shells](#)
[\[ippe_th \] IPPE Th shell with 14 MeV and Cf-252 source neutrons](#)
[\[iri_tub \] Streaming Through Ducts \(IRI-TUB\)](#)
[\[isis800 \] ISIS Deep-Penetration Neutrons through Concrete and Iron Shields using p-800 MeV](#)
[\[janus_1 \] JANUS Phase I \(Neutron Transport Through Mild and Stainless Steel\)](#)
[\[janus_8 \] JANUS Phase VIII \(Neutron Transport Through Sodium and Mild Steel\)](#)
[\[JAS_AX \] JASPER Advanced Reactor Axial Shield Measurements](#)
[\[JAS_IHX \] JASPER Advanced Reactor Intermediate Heat Exchanger Measurements](#)
[\[JAS_RAD \] JASPER Advanced Reactor Radial Shield Measurements](#)
[\[juelich_li \] Juelich Li Metal Blanket Experiment](#)
[\[kant \] Spherical Beryllium Shells](#)
[\[kens500 \] KENS p-500 MeV shielding experiment using 4m Concrete at KEK](#)
[\[kfk_fe \] Karlsruhe Iron Sphere](#)
[\[msu \] MSU experiment with He & C ions on Al target](#)
[\[mephi \] MEPhI empty slits streaming experiment](#)
[\[nai60fec \] NAIÁDE 1 Graphite Benchmark \(60cm\)](#)
[\[nai60h2o \] NAIÁDE 1 Light Water Benchmark \(60cm\)](#)
[\[nai60fec \] NAIÁDE 1 Iron Benchmark \(60cm\)](#)
[\[nesdip_2 \] NESDIP-2 Benchmark \(ASPIS\)](#)
[\[nesdip_3 \] NESDIP-3 Benchmark \(ASPIS\)](#)
[\[nist_h2o \] Neutron Leakage from Water Spheres \(NIST\)](#)

Fig. A.4. Previous search tool 4.

[\[oktav_al \] Aluminium Sphere \(OKTAVIAN\)](#)
[\[oktav_fe \] Iron Sphere \(OKTAVIAN\)](#)
[\[oktav_ni \] Nickel Sphere \(OKTAVIAN\)](#)
[\[oktav_si \] Silicon Sphere \(OKTAVIAN\)](#)
[\[oktav_w \] Tungsten Sphere \(OKTAVIAN\)](#)
[\[PCA_PV \] Pool Critical Assembly-Pressure Vessel Facility Benchmark](#)
[\[pca_repl \] Winfrith Water/Iron Benchmark \(ASPIS-PCA REPLICA\)](#)
[\[prote_fe \] Wuorenlingen Iron Benchmark \(PROTEUS\)](#)
[\[psi590 \] Neutron Spectra Generated by 590-MeV Protons on Thick Pb Target](#)
[\[rfnc_ph \] Photon Leakage Spectra from Al, Ti, Fe, Cu, Zr, Pb, U238 Spheres](#)
[\[rfnc_ph2 \] Photon Spectra from H2O, SiO2 and NaCl](#)
[\[riken \] RIKEN Quasi-monoenergetic Neutron Field in 70-210 MeV Energy Range](#)
[\[rosti \] ROESTI I, II and III \(CERN\)](#)
[\[SB2_GAM \] Gamma-ray Production Cross Sections from Thermal Neutron Capture in 14 elements and SS](#)
[\[SB3_GAM \] Averaged Gamma-ray Production Cross Sections from Fast Neutron Capture in 14 ele. & SS](#)
[\[SB5_FUS \] ORNL 14-MeV Neutron SS/Borated Poly Slab](#)
[\[SDT1 \] ORNL TSF Iron Broomstick](#)
[\[SDT2 \] ORNL TSF Oxygen Broomstick](#)
[\[SDT3 \] ORNL TSF Nitrogen Broomstick](#)
[\[SDT4 \] ORNL TSF Sodium Broomstick](#)
[\[SDT5 \] ORNL TSF Stainless Steel Broomstick](#)
[\[SDT11 \] ORNL Neutron Transport Through Iron and SS - Part I](#)
[\[SDT12 \] ORNL Neutron Transport in Thick Sodium](#)
[\[skyshine \] Baikal-1 Skyshine Benchmark Experiment](#)
[\[tepc-fluka \] Simulation of the lineal energy distribution of the energy deposition in biological cells, TEPC-FLUKA Comparison](#)
[\[tiara \] TIARA 40 and 65 MeV Neutron Transmission Through Iron, Concrete and Polyethylene](#)
[\[tud_fe \] TUD Iron Slab Experiment](#)
[\[tud_fng \] FNG/TUD ITER Blanket Bulk Shield \(spectra measurements\)](#)
[\[tud_sic \] FNG/TUD Silicon Carbide \(spectra\)](#)
[\[tud_w \] FNG/TUD Tungsten \(spectra measurements\)](#)
[\[venus3 \] VENUS-3 LWR-PVS Benchmark](#)
[\[vver440 \] Radiation field parameters for pressure vessel monitoring in NRI LR-0 VVER-440 reactor](#)
[\[vver1000 \] Radiation field parameters for pressure vessel monitoring in NRI LR-0 VVER-1000 reactor](#)
[\[YAYOI_FE \] University of Tokyo-YAYOI Iron Slab](#)

SINBAD INDEX - Sorted by Laboratory

[\[AGH-UST, Poland \] Juelich Li Metal Blanket Experiment](#)
[\[ARCS, Austria \] Simulation of the lineal energy distribution of the energy deposition in biological cells, TEPC-FLUKA Comparison](#)
[\[Cadarache, France - Harmonie \] Cadarache Sodium](#)

Fig. A.5. Previous search tool 5.

[\[CEA, France | NAIADE 1 Graphite Benchmark \(60cm\)](#)
[\[CEA, France | NAIADE 1 Iron Benchmark \(60cm\)](#)
[\[CEA, France | NAIADE 1 Light Water Benchmark \(60cm\)](#)
[\[CERN, Switzerland | ROESTI I, II and III](#)
[\[CERN, Switzerland | CERF Bonner Sphere Spectrometer Response to Charged Hadrons](#)
[\[CERN, Switzerland | CERF Radionuclide Production](#)
[\[CERN, Switzerland | CERF Residual Dose Rates](#)
[\[CERN, Switzerland | CERF Neutron Energy Spectra behind Shielding of a 120 GeV/c Hadron Beam Facility](#)
[\[CERN, Switzerland | CERN 200 and 400 GeV/c protons activation experiments](#)
[\[FNS/JAEA, Japan | FNS Experimental data for fusion neutronics benchmark](#)
[\[FNS/JAEA, Japan | FNS Integral Experiment on Graphite Cylindrical Assembly](#)
[\[FNS/JAEA, Japan | FNS Liquid Oxygen](#)
[\[FNS/JAEA, Japan | FNS Vanadium Cube](#)
[\[FNS/JAEA, Japan | FNS Tungsten](#)
[\[FNS/JAEA, Japan | FNS Skyshine](#)
[\[FNS/JAEA, Japan | FNS Dogleg Duct Streaming](#)
[\[FNG/ENEA, Italy | FNG-SS Shield \(integral meas.\)](#)
[\[FNG/ENEA, Italy | FNG-ITER Blanket Bulk Shield \(integral meas.\)](#)
[\[FNG/ENEA, Italy | FNG-ITER Neutron Streaming \(integral\)](#)
[\[FNG/ENEA, Italy | FNG-ITER Dose Rate Experiment](#)
[\[FNG/ENEA, Italy | FNG Silicon Carbide \(integral\)](#)
[\[FNG/ENEA, Italy | FNG Tungsten \(integral\)](#)
[\[FNG/Italy, TUD/Germany | FNG/TUD ITER Blanket Bulk Shield \(spectra measurements\)](#)
[\[FNG/Italy, TUD/Germany | FNG/TUD Silicon Carbide \(spectra\)](#)
[\[FNG/Italy, TUD/Germany | FNG/TUD Tungsten \(spectra measurements\)](#)
[\[TUD, Germany | TUD Iron Slab Experiment](#)
[\[FzK, Germany | Karlsruhe Iron Sphere](#)
[\[FzK, Germany | KANT Spherical Beryllium Shells](#)
[\[IPPE - FzK | IPPE Vanadium Shells](#)
[\[IPPE, Russia | IPPE Iron Shells](#)
[\[IPPE, Russia | IPPE Th shell with 14 MeV and Cf-252 source neutrons](#)
[\[IRI - TUB | Streaming Through Ducts](#)
[\[ISPRA Univ. of Pavia, Italy - EURACOS II | Ispra Iron Benchmark](#)
[\[ISPRA Univ. of Pavia, Italy - EURACOS II | Ispra Sodium Benchmark](#)
[\[JAEA, Japan | TIARA 40 and 65 MeV Neutron Transmission Through Iron, Concrete and Polyethylene](#)
[\[JAEA, Japan | Radioactivity induced by GeV-Protons and Spallation Neutrons using AGS accelerator](#)
[\[JAEA, Japan | Intermediate and High-Energy Accelerator Shielding Benchmarks](#)
[\[KEK/KENS, Japan | KENS p-500 MeV shielding experiment using 4m Concrete at KEK](#)
[\[LBNL, USA | BEVALAC Experiment with Nb Ions on Nb & Al Targets](#)
[\[MEPhI, Russian Fed. | MEPhI empty slits streaming experiment](#)
[\[MSU, USA | Experiment with He & C ions on Al target](#)
[\[NIRS, Japan | HIMAC experiments with He, C, Ne, Ar, Fe, Xe and Si ions on C, Al, Cu & Pb targets](#)
[\[NIRS, Japan | HIMAC High energy Neutron \(<800 MeV\) Measurements in Iron](#)
[\[NIRS, Japan | HIMAC High energy Neutron \(<800 MeV\) Measurements in Concrete](#)

Fig. A.6. Previous search tool 6.

[\[NIST, USA \] Neutron Leakage from Water Spheres](#)
[\[NRC, USA \] H.B. Robinson-2 Pressure Vessel](#)
[\[NRI, Rez \] Radiation field parameters for pressure vessel monitoring in NRI LR-0 VVER-440 reactor](#)
[\[NRI, Rez \] Radiation field parameters for pressure vessel monitoring in NRI LR-0 VVER-1000 reactor](#)
[\[ORNL, USA \] Gamma-ray Production Cross Sections from Thermal Neutron Capture in 14 elements and SS](#)
[\[ORNL, USA \] Averaged Gamma-ray Production Cross Sections from Fast Neutron Capture in 14 ele. & SS](#)
[\[ORNL, USA \] JASPER Advanced Reactor Axial Shield Measurements](#)
[\[ORNL, USA \] JASPER Advanced Reactor Intermediate Heat Exchanger Measurements](#)
[\[ORNL, USA \] JASPER Advanced Reactor Radial Shield Measurements](#)
[\[ORNL, USA \] ORNL TSF Iron Broomstick](#)
[\[ORNL, USA \] ORNL TSF Oxygen Broomstick](#)
[\[ORNL, USA \] ORNL TSF Nitrogen Broomstick](#)
[\[ORNL, USA \] ORNL TSF Sodium Broomstick](#)
[\[ORNL, USA \] ORNL TSF Stainless Steel Broomstick](#)
[\[ORNL, USA \] ORNL Neutron Transport Through Iron and SS - Part I](#)
[\[ORNL, USA \] ORNL Neutron Transport in Thick Sodium](#)
[\[ORNL, USA \] Pool Critical Assembly-Pressure Vessel Facility Benchmark](#)
[\[ORNL, USA \] ORNL 14-MeV Neutron SS/Borated Poly Slab](#)
[\[PSI, Swiss \] Neutron Spectra Generated by 590-MeV Protons on Thick Pb Target](#)
[\[RAL, England \] ISIS Deep-Penetration Neutrons through Concrete and Iron Shields using p-800 MeV](#)
[\[RDIPE, Russia \] Baikal-1 Skyshine Benchmark Experiment](#)
[\[RFNC, Russia \] Photon Leakage Spectra from Al, Ti, Fe, Cu, Zr, Pb, U238 Spheres](#)
[\[RFNC, Russia \] Photon Spectra from H₂O, SiO₂ and NaCl](#)
[\[RIKEN, Japan \] Quasi-monoenergetic Neutron Field in 70-210 MeV Energy Range](#)
[\[SNL, USA \] Polyethylene-Reflected Plutonium Metal Sphere HPGe](#)
[\[SNL, USA \] Polyethylene-Reflected Plutonium Metal Sphere NPOD](#)
[\[SNL, USA \] Polyethylene-Reflected Plutonium Metal Sphere SNAP](#)
[\[SCK/CEN, Belgium \] VENUS-3 LWR-PVS Benchmark](#)
[\[SEC NRS/FZR, Russia/Germany \] Balakovo-3 VVER-1000](#)
[\[University of Illinois, USA \] University of Illinois Iron Sphere \(D-T\)](#)
[\[University of Illinois, USA \] University of Illinois Iron Sphere \(CF-252\)](#)
[\[Univ. of Osaka, Japan - OKTAVIAN \] Aluminium Sphere](#)
[\[Univ. of Osaka, Japan - OKTAVIAN \] Iron Sphere](#)
[\[Univ. of Osaka, Japan - OKTAVIAN \] Nickel Sphere](#)
[\[Univ. of Osaka, Japan - OKTAVIAN \] Silicon Sphere](#)
[\[Univ. of Osaka, Japan - OKTAVIAN \] Tungsten Sphere](#)
[\[Univ. of Osaka, Japan - AVF Cyclotron \] Transmission Through Shielding Materials of Neutrons and Photons Generated by 65 MeV Protons](#)
[\[Univ. of Osaka, Japan - AVF Cyclotron \] Transmission of Medium Energy Neutrons Through Concrete Shields \(AVF Cyclotron\)](#)
[\[Univ. of Osaka and Univ. of Tokyo, Japan \] Neutron Production from Thick Targets of Carbon, Iron, Copper, and Lead by 30- and 52-MeV Protons](#)
[\[Univ. of Tokyo, Japan - INS \] Transmission Through Shielding Materials of Neutrons and Photons Generated by 52 MeV Protons](#)
[\[Univ. of Tokyo, Japan - YAYOI \] University of Tokyo-YAYOI Iron Slab](#)
[\[Winfrith, England - ASPIS \] Winfrith Iron Benchmark](#)
[\[Winfrith, England - ASPIS \] Winfrith Iron 88 Benchmark](#)
[\[Winfrith, England - ASPIS \] Winfrith Graphite Benchmark](#)
[\[Winfrith, England \] Winfrith Water Benchmark](#)

Fig. A.7. Previous search tool 7.

[\[Winfrith, England - ASPIS \] Winfrith Neutron-Gamma Ray Transport through Water/Steel Arrays](#)
[\[Winfrith, England - ASPIS \] JANUS Phase I \(Neutron Transport Through Mild and Stainless Steel\)](#)
[\[Winfrith, England - ASPIS \] JANUS Phase VIII \(Neutron Transport Through Sodium and Mild Steel\)](#)
[\[Winfrith, England - ASPIS \] NESDIP-2 Benchmark](#)
[\[Winfrith, England - ASPIS \] NESDIP-3 Benchmark](#)
[\[Winfrith, England - ASPIS \] Winfrith Water/Iron Benchmark](#)
[\[Wuerenlingen, Switzerland - PROTEUS \] Wuerenlingen Iron Benchmark](#)

SINBAD INDEX - Sorted by Year

[\[1960 \] NAIÁDE 1 Graphite Benchmark \(60cm\)](#)
[\[1960 \] NAIÁDE 1 Iron Benchmark \(60cm\)](#)
[\[1960 \] NAIÁDE 1 Light Water Benchmark \(60cm\)](#)
[\[1968 \] ORNL TSF Iron Broomstick](#)
[\[1968 \] ORNL TSF Oxygen Broomstick](#)
[\[1968 \] ORNL TSF Nitrogen Broomstick](#)
[\[1968 \] ORNL TSF Sodium Broomstick](#)
[\[1968 \] ORNL TSF Stainless Steel Broomstick](#)
[\[1969 \] Gamma-ray Production Cross Sections from Thermal Neutron Capture in 14 elements and SS](#)
[\[1969 \] Averaged Gamma-ray Production Cross Sections from Fast Neutron Capture in 14 ele. & SS](#)
[\[1974 \] ORNL Neutron Transport Through Iron and SS - Part I](#)
[\[1974 \] ORNL Neutron Transport in Thick Sodium](#)
[\[1975 \] University of Illinois Iron Sphere \(CF-252\)](#)
[\[1975 \] University of Illinois Iron Sphere \(D-T\)](#)
[\[1975 \] Karlsruhe Iron Sphere](#)
[\[~1975 \] Winfrith Iron Benchmark \(ASPIS\)](#)
[\[1976 \] University of Tokyo-YAYOI Iron Slab](#)
[\[1976-1984 \] Juelich Li Metal Blanket Experiment](#)
[\[1978 \] Cadarache Sodium \(HARMONIE\)](#)
[\[1979 \] ORNL 14-MeV Neutron SS/Borated Poly Slab](#)
[\[1979 \] Neutron Spectra Generated by 590-MeV Protons on Thick Pb Target](#)
[\[1980 \] Pool Critical Assembly-Pressure Vessel Facility Benchmark](#)
[\[1981 \] Transmission Through Shielding Materials of Neutrons and Photons Generated by 52 MeV Protons](#)
[\[1982 \] Neutron Production from Thick Targets of Carbon, Iron, Copper, and Lead by 30- and 52-MeV Protons](#)
[\[1983 \] CERN 200 and 400 GeV/c protons activation experiments](#)
[\[1983-1991 \] FNS Experimental data for fusion neutronics benchmark](#)
[\[1984 \] H.B. Robinson-2 Pressure Vessel](#)
[\[1984 \] Winfrith Graphite Benchmark \(ASPIS\)](#)
[\[1984 \] Winfrith Water Benchmark](#)
[\[1984 \] Winfrith Water/Iron Benchmark \(ASPIS-PCA REPLICIA\)](#)

Fig. A.8. Previous search tool 8.

[\[1984 \] Wuerenlingen Iron Benchmark \(PROTEUS\)](#)
[\[1984 \] FNS Integral Experiment on Graphite Cylindrical Assembly](#)
[\[1985 \] NESDIP-3 Benchmark \(ASPIS\)](#)
[\[1986 \] Ispra Iron Benchmark \(EURACOS\)](#)
[\[1986 \] Ispra Sodium Benchmark \(EURACOS\)](#)
[\[1986 \] JANUS Phase I \(Neutron Transport Through Mild and Stainless Steel\)](#)
[\[1987 \] JASPER Advanced Reactor Radial Shield Measurements](#)
[\[1987 \] Winfrith Neutron-Gamma Ray Transport through Water/Steel Arrays \(ASPIS\)](#)
[\[1987 \] OKTAVIAN Silicon Sphere](#)
[\[1987 \] OKTAVIAN Tungsten Sphere](#)
[\[1987-1992 \] IPPE Th shell with 14 MeV and Cf-252 source neutrons](#)
[\[1988 \] OKTAVIAN Iron Sphere](#)
[\[1988 \] OKTAVIAN Aluminium Sphere](#)
[\[1988 \] VENUS-3 LWR-PVS Benchmark](#)
[\[1989 \] FNG-SS Shield \(integral meas.\)](#)
[\[1989 \] FNS Liquid Oxygen](#)
[\[1989-1991 \] ROESTI I, II and III \(CERN\)](#)
[\[1989-1995 \] IPPE Iron Shells](#)
[\[~1990 \] Radiation field parameters for pressure vessel monitoring in NRI LR-0 VVER-440 reactor](#)
[\[~1990 \] Radiation field parameters for pressure vessel monitoring in NRI LR-0 VVER-1000 reactor](#)
[\[1990 \] Neutron Leakage from Water Spheres \(NIST\)](#)
[\[1990 \] JANUS Phase VIII \(Neutron Transport Through Sodium and Mild Steel\)](#)
[\[1990 - 1994 \] KANT Spherical Beryllium Shells](#)
[\[1991 \] Streaming Through Ducts \(IRI-TUB\)](#)
[\[1991 \] Transmission Through Shielding Materials of Neutrons and Photons Generated by 65 MeV Protons](#)
[\[1991 \] Transmission of Medium Energy Neutrons Through Concrete Shields \(AVF Cyclotron\)](#)
[\[1991 \] JASPER Advanced Reactor Axial Shield Measurements](#)
[\[1991 \] BEVALAC Experiment with Nb Ions on Nb & Al Targets](#)
[\[1992 \] JASPER Advanced Reactor Intermediate Heat Exchanger Measurements](#)
[\[1992 \] Photon Leakage Spectra from Al, Ti, Fe, Cu, Zr, Pb, U238 Spheres](#)
[\[1993 \] Winfrith Iron 88 Benchmark \(ASPIS\)](#)
[\[1993 \] OKTAVIAN Nickel Sphere](#)
[\[1993 \] FNS Tungsten](#)
[\[1993 \] MSU experiment with He & C ions on Al target](#)
[\[1994 \] NESDIP-2 Benchmark \(ASPIS\)](#)
[\[1994-1995 \] Balakovo-3 VVER-1000](#)
[\[1995 \] TUD Iron Slab Experiment](#)
[\[1995 \] FNG-ITER Blanket Bulk Shield \(integral meas.\)](#)
[\[1995-1998 \] MEFH empty slits streaming experiment](#)
[\[1996 \] FNG/TUD ITER Blanket Bulk Shield \(spectra measurements\)](#)
[\[1996 \] FNS Vanadium Cube](#)
[\[1996, 1997 \] TIARA 40 and 65 MeV Neutron Transmission Through Iron, Concrete and Polyethylene](#)
[\[1996, 1997 \] IPPE Vanadium Shells](#)
[\[1996-1998 \] Baikal-1 Skyshine Benchmark Experiment](#)

Fig. A.9. Previous search tool 9.

[\[1997-1998 \] FNG-ITER Neutron Streaming \(integral\)](#)
[\[1997 - 1999 \] HIMAC experiments with He, C, Ne, Ar, Fe, Xe and Si ions on C, Al, Cu & Pb targets](#)
[\[1998 \] ISIS Deep-Penetration Neutrons through Concrete and Iron Shields using p-800 MeV](#)
[\[~1998? \] RIKEN Quasi-monoenergetic Neutron Field in 70-210 MeV Energy Range](#)
[\[2000-2001 \] FNG-ITER Dose Rate Experiment](#)
[\[2001 \] FNG Silicon Carbide \(integral\)](#)
[\[2001 \] FNG/TUD Silicon Carbide \(spectra measurements\)](#)
[\[2001 \] HIMAC High energy Neutron \(<800 MeV\) Measurements in Concrete](#)
[\[2001 \] Radioactivity induced by GeV-Protons and Spallation Neutrons using AGS accelerator](#)
[\[2002 \] FNG Tungsten \(integral\)](#)
[\[2002 \] FNG/TUD Tungsten \(spectra measurements\)](#)
[\[2002 \] FNS Skyshine](#)
[\[2002 \] HIMAC High energy Neutron \(<800 MeV\) Measurements in Iron](#)
[\[2002-2004 \] KENS p-500 MeV shielding experiment using 4m Concrete at KEK](#)
[\[2003 \] CERF Bonner Sphere Spectrometer Response to Charged Hadrons](#)
[\[2003 \] CERF Residual Dose Rates](#)
[\[2003 \] Simulation of the lineal energy distribution of the energy deposition in biological cells, TEPC-FLUKA Comparison](#)
[\[~2003 \] CERF Radionuclide Production](#)
[\[2004 \] FNS Dogleg Duct Streaming](#)
[\[2004 \] CERF Neutron Energy Spectra behind Shielding of a 120 GeV/c Hadron Beam Facility](#)
[\[2006 \] Photon Spectra from H₂O, SiO₂ and NaCl](#)
[\[2009 \] Polyethylene-Reflected Plutonium Metal Sphere: Subcritical Neutron and Gamma Measurements](#)
[\[? \] Intermediate and High-Energy Accelerator Shielding Benchmarks](#)

SINBAD INDEX - Sorted by Material

[\[Air \] Streaming Through Ducts \(IRI-TUB\)](#)
[\[Air \] FNG-ITER Neutron Streaming \(integral\)](#)
[\[Air \] FNS Dogleg Duct Streaming](#)
[\[Air \] FNS Skyshine](#)
[\[Air \] Baikal-1 Skyshine Benchmark Experiment](#)
[\[Air \] RIKEN Quasi-monoenergetic Neutron Field in 70-210 MeV Energy Range](#)
[\[Air \] Simulation of the lineal energy distribution of the energy deposition in biological cells, TEPC-FLUKA Comparison](#)
[\[Air \] CERF Bonner Sphere Spectrometer Response to Charged Hadrons](#)
[\[Aluminium \] Aluminium Sphere \(OKTAVIAN\)](#)
[\[Air \] MEPHI empty slits streaming experiment](#)
[\[Aluminium \] MSU experiment with He & C ions on Al target](#)
[\[Beryllium \] KANT Spherical Beryllium Shells](#)
[\[Concrete \] HIMAC High energy Neutron \(<800 MeV\) Measurements in Concrete](#)
[\[Concrete \] KENS p-500 MeV shielding experiment using 4m Concrete at KEK](#)
[\[Concrete \] Transmission of Medium Energy Neutrons Through Concrete Shields \(AVF Cyclotron\)](#)

Fig. A.10. Previous search tool 10.

[\[Copper \] CERN 200 and 400 GeV/c protons activation experiments](#)
[\[Copper \] CERF Neutron Energy Spectra behind Shielding of a 120 GeV/c Hadron Beam Facility](#)
[\[Graphite \] NAIÁDE 1 Graphite Benchmark \(60cm\)](#)
[\[Graphite \] Winfrith Graphite Benchmark \(ASPIS\)](#)
[\[Graphite \] FNS Integral Experiment on Graphite Cylindrical Assembly](#)
[\[Iron \] NAIÁDE 1 Iron Benchmark \(60cm\)](#)
[\[Iron \] Karlsruhe Iron Sphere](#)
[\[Iron \] Winfrith Iron Benchmark \(ASPIS\)](#)
[\[Iron \] Wuerenlingen Iron Benchmark \(PROTEUS\)](#)
[\[Iron \] Ispra Iron Benchmark \(EURACOS\)](#)
[\[Iron \] Iron Sphere \(OKTAVIAN\)](#)
[\[Iron \] Winfrith Iron 88 Benchmark \(ASPIS\)](#)
[\[Iron \] Iron Slab Experiment \(TUD\)](#)
[\[Iron \] University of Tokyo-YAYOI Iron Slab](#)
[\[Iron \] ORNL TSF Iron Broomstick](#)
[\[Iron \] University of Illinois Iron Sphere \(CF-252\)](#)
[\[Iron \] University of Illinois Iron Sphere \(D-T\)](#)
[\[Iron \] IPPE Iron Shells](#)
[\[Iron \] HIMAC High energy Neutron \(<800 MeV\) Measurements in Iron](#)
[\[Iron, Concrete \] ISIS Deep-Penetration Neutrons through Concrete and Iron Shields using p-800 MeV](#)
[\[Iron, Concrete, Polyethylene \] TIARA 40 and 65 MeV Neutron Transmission Through Iron, Concrete and Polyethylene](#)
[\[Iron, Lead \] ROESTI I, II and III \(CERN\)](#)
[\[Lead \] Neutron Spectra Generated by 590-MeV Protons on Thick Pb Target](#)
[\[Lithium \] Juelich Li Metal Blanket Experiment](#)
[\[Nickel \] Nickel Sphere \(OKTAVIAN\)](#)
[\[Niobium, Aluminum \] BEVALAC Experiment with Nb Ions on Nb & Al Targets](#)
[\[Nitrogen \] ORNL TSF Nitrogen Broomstick](#)
[\[Oxygen \] FNS Liquid Oxygen](#)
[\[Oxygen \] ORNL TSF Oxygen Broomstick](#)
[\[Plutonium \] Polyethylene-Reflected Plutonium Metal Sphere: Subcritical Neutron and Gamma Measurements HPGe](#)
[\[Plutonium \] Polyethylene-Reflected Plutonium Metal Sphere: Subcritical Neutron and Gamma Measurements NPOD](#)
[\[Plutonium \] Polyethylene-Reflected Plutonium Metal Sphere: Subcritical Neutron and Gamma Measurements SNAP](#)
[\[Silicon \] Silicon Sphere \(OKTAVIAN\)](#)
[\[Silicon Carbide \] FNG Silicon Carbide \(integral\)](#)
[\[Silicon Carbide \] FNG/TUD Silicon Carbide \(spectra\)](#)
[\[Sodium \] Cadarache Sodium \(HARMONIE\)](#)
[\[Sodium \] Ispra Sodium Benchmark \(EURACOS\)](#)
[\[Sodium \] ORNL TSF Sodium Broomstick](#)
[\[Sodium \] ORNL Neutron Transport in Thick Sodium](#)
[\[Sodium, Steel \] JANUS Phase VIII \(Neutron Transport Through Sodium and Mild Steel\)](#)
[\[Steel \] JANUS Phase I \(Neutron Transport Through Mild and Stainless Steel\)](#)
[\[Stainless Steel \] ORNL TSF Stainless Steel Broomstick](#)
[\[Stainless Steel \] FNG-SS Shield \(integral meas.\)](#)
[\[Iron and Stainless Steel \] ORNL Neutron Transport Through Iron and SS - Part I](#)

Fig. A.11. Previous search tool 11.

[\[Stainless Steel and Poly \] ORNL 14-MeV Neutron SS/Borated Poly Slab](#)
[\[Stainless Steel, Poly, Copper \] FNG-ITER Blanket Bulk Shield \(integral\)](#)
[\[Stainless Steel, Poly, Copper \] FNG/TUD ITER Blanket Bulk Shield \(spectra\)](#)
[\[Stainless Steel, Perspex \] FNG-ITER Dose Rate Experiment](#)
[\[Thorium \] IPPE Th shell with 14 MeV and Cf-252 source neutrons](#)
[\[Tungsten \] Tungsten Sphere \(OKTAVIAN\)](#)
[\[Tungsten \] FNS Tungsten](#)
[\[Tungsten \] FNG Tungsten \(integral\)](#)
[\[Tungsten \] FNG/TUD Tungsten \(spectra\)](#)
[\[Vanadium \] FNS Vanadium Cube](#)
[\[Vanadium \] IPPE Vanadium Shells](#)
[\[Water \] Winfrith Water Benchmark](#)
[\[Water \] Neutron Leakage from Water Spheres \(NIST\)](#)
[\[Water \] NAI/DE 1 Light Water Benchmark \(60cm\)](#)
[\[Water, Graphite, Iron \] Transmission Through Shielding Materials of Neutrons and Photons Generated by 52 MeV Protons](#)
[\[Water, Graphite, Lead \] Transmission Through Shielding Materials of Neutrons and Photons Generated by 65 MeV Protons](#)
[\[Water, Iron \] Pool Critical Assembly-Pressure Vessel Facility Benchmark](#)
[\[Water, Iron \] Winfrith Water/Iron Benchmark \(ASPIS-PCA REPLICIA\)](#)
[\[Water, Steel \] Winfrith Neutron-Gamma Ray Transport through Water/Steel Arrays \(ASPIS\)](#)
[\[Water, Steel, Aluminium \] NESDIP-3 Benchmark \(ASPIS\)](#)
[\[Water, Steel, Aluminium \] NESDIP-2 Benchmark \(ASPIS\)](#)
[\[Water, Silicon-Dioxide, Sodium-Chloride \] Photon Spectra from H₂O, SiO₂ and NaCl](#)
MULTIPLE (MORE THAN 3 Materials, either INTEGRAL or Separate)

[Radiation field parameters for pressure vessel monitoring in NRI LR-0 VVER-440 reactor](#)
[Radiation field parameters for pressure vessel monitoring in NRI LR-0 VVER-1000 reactor](#)
[Balakovo-3 VVER-1000 Ex-vessel Neutron Dosimetry Benchmark](#)
[CERF Residual Dose Rates](#)
[CERF Radionuclide Production](#)
[H.B. Robinson-2 Pressure Vessel](#)
[JASPER Advanced Reactor Axial Shield Measurements](#)
[JASPER Advanced Reactor Intermediate Heat Exchanger Measurements](#)
[JASPER Advanced Reactor Radial Shield Measurements](#)
[Gamma-ray Production Cross Sections from Thermal Neutron Capture in 14 elements and SS](#)
[Averaged Gamma-ray Production Cross Sections from Fast Neutron Capture in 14 ele. & SS](#)
[HIMAC experiments with He, C, Ne, Ar, Fe, Xe and Si ions on C, Al, Cu & Pb targets](#)
[VENUS-3 LWR-PVS Benchmark](#)
[Photon Leakage Spectra from Al, Ti, Fe, Cu, Zr, Pb, U238 Spheres](#)
[Neutron Production from Thick Targets of Carbon, Iron, Copper, and Lead by 30- and 52-MeV Protons](#)
[Radioactivity induced by GeV-Protons and Spallation Neutrons using AGS accelerator](#)
[FNS Experimental data for fusion neutronics benchmark](#)
[Intermediate and High-Energy Accelerator Shielding Benchmarks](#)

Fig. A.12. Previous search tool 12.

APPENDIX B:
LIST OF THE SINBAD SEARCH TOOL CODE, WRITTEN IN JAVA LANGUAGE
ON THE NETBEANS IDE 7.1 PLATFORM

SinbadInterface.java (The Main Window)

```
import java.io.FileNotFoundException;
import java.io.IOException;
import java.util.ArrayList;
import java.util.Arrays;
import java.util.Iterator;
import java.util.List;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.swing.JOptionPane;

public class SinbadInterface extends javax.swing.JFrame {
    public ArrayList keyList= new ArrayList();
    String typ="All";

    /**
     * Creates new form SinbadeInterface
     */
    public SinbadInterface() {
        initComponents();
        setLocationRelativeTo(null);
    }

    /**
     * This method is called from within the constructor to initialize the form.
     * WARNING: Do NOT modify this code. The content of this method is always
     * regenerated by the Form Editor.
     */
    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code">
    private void initComponents() {

        jLabel1 = new javax.swing.JLabel();
        jLabel2 = new javax.swing.JLabel();
        jLabel3 = new javax.swing.JLabel();
        jLabel4 = new javax.swing.JLabel();
        jLabel5 = new javax.swing.JLabel();
        jScrollPane1 = new javax.swing.JScrollPane();
        jList1 = new javax.swing.JList();
        jScrollPane2 = new javax.swing.JScrollPane();
        jList2 = new javax.swing.JList();
        jComboBox1 = new javax.swing.JComboBox();
        jComboBox2 = new javax.swing.JComboBox();
        jButton1 = new javax.swing.JButton();
        jButton2 = new javax.swing.JButton();

        setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
    }
}
```

```

setTitle("SINBAD SEARCH");
setResizable(false);

jLabel1.setFont(new java.awt.Font("Tahoma", 1, 14)); // NOI18N
jLabel1.setText("SINBAD SEARCH");

jLabel2.setFont(new java.awt.Font("Arial", 1, 12)); // NOI18N
jLabel2.setText("Type");

jLabel3.setFont(new java.awt.Font("Arial", 1, 12)); // NOI18N
jLabel3.setText("Material");

jLabel4.setFont(new java.awt.Font("Arial", 1, 12)); // NOI18N
jLabel4.setText("Laboratory");

jLabel5.setFont(new java.awt.Font("Arial", 1, 12)); // NOI18N
jLabel5.setText("Geometry");

jList1.setModel(new javax.swing.AbstractListModel() {
    String[] strings = { "All", "Reactor Shielding", "Fusion Neutronics Shielding", "Accelerator
Shielding" };
    public int getSize() { return strings.length; }
    public Object getElementAt(int i) { return strings[i]; }
});
jList1.setSelectionMode(javax.swing.ListSelectionModel.SINGLE_SELECTION);
jList1.setToolTipText("");
jList1.setSelectedIndex(0);
jList1.addListSelectionListener(new javax.swing.event.ListSelectionListener() {
    public void valueChanged(javax.swing.event.ListSelectionEvent evt) {
        type(evt);
    }
});
jScrollPane1.setViewportView(jList1);

jList2.setModel(new javax.swing.AbstractListModel() {
    String[] strings = { "Any", "aluminium ", "Beryllium ", "Concrete", "copper ", "Graphite ", "Iron
", "Lead ", "Lithium ", "Nickel ", "Niobium ", "Nitrogen ", "Oxygen", "polyethylene ", "Silicon ",
"silicon-dioxide ", "sodium-Chloride ", "Sodium ", "Steel ", "Perspex ", "Tungsten ", "Vanadium ",
"Water " };
    public int getSize() { return strings.length; }
    public Object getElementAt(int i) { return strings[i]; }
});
jList2.setLayoutOrientation(javax.swing.JList.HORIZONTAL_WRAP);
jList2.setSelectedIndex(0);
jScrollPane2.setViewportView(jList2);

jComboBox1.setModel(new javax.swing.DefaultComboBoxModel(new String[] { "All", "AGH-
UST/Poland", "ARCS/Austria", "Cadarache/France-Harmonie", "CEA/France", "CERN/Switzerland",
"FNS/JAEA-Japan", "FNG/ENEA-Italy", "FNG/Italy-TUD/Germany", "TUD/Germany",
"FzK/Germany", "IPPE/Russia", "IRI-TUB", "ISPRA Univ. of Pavia/Italy-EURACOS II",
"JAEA/Japan", "KEK/KENS-Japan", "LBNL/USA", "MEPhI/Russian Fed.", "MSU/USA",

```

```
"NIRS/Japan", "NIST/USA", "NRC/USA", "NRI/Rez", "ORNL/USA", "PSI/Swiss", "RAL/England",
"RDIPE/Russia", "RFNC/Russia", "RIKEN/Japan", "SNL/USA", "SCK/CEN-Belgium", "SEC
NRS/FZR-Russia/Germany", "University of Illinois/USA", "Univ. of Osaka/Japan-OKTAVIAN", "Univ.
of Osaka/Japan-AVF Cyclotron", "Univ. of Osaka and Univ. of Tokyo/Japan", "Univ. of Tokyo/Japan-
INS", "Univ. of Tokyo/Japan-YAYOI", "Winfrith/England-ASPIS", "Wuerenlingen/Switzerland-
PROTEUS" }));
```

```
jComboBox2.setModel(new javax.swing.DefaultComboBoxModel(new String[] { "Any", "Sphere",
"Cylindrical" }));
```

```
jButton1.setFont(new java.awt.Font("Arial", 1, 12)); // NOI18N
jButton1.setText("EXECUTE SEARCH");
jButton1.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        search(evt);
    }
});
```

```
jButton2.setText("jButton2");
```

```
javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
getContentPane().setLayout(layout);
layout.setHorizontalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()
            .addGap(53, 53, 53)
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addComponent(jLabel4)
                .addComponent(jLabel3)
                .addComponent(jLabel2)
                .addComponent(jLabel5))
            .addGap(74, 74, 74)
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.CENTER)
                .addComponent(jLabel1)
                .addComponent(jScrollPane1)
                .addComponent(jScrollPane2)
                .addComponent(jComboBox1, 0, javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE))
                .addComponent(jComboBox2, 0, javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE))
            .addContainerGap(258, Short.MAX_VALUE))
        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING, layout.createSequentialGroup()
            .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
            .addComponent(jButton1)
            .addGap(56, 56, 56))
    );
layout.setVerticalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()
            .addGap(28, 28, 28)
            .addComponent(jLabel1)
```

```

        .addGap(50, 50, 50)
        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addComponent(jLabel2)
            .addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED_SIZE, 96,
javax.swing.GroupLayout.PREFERRED_SIZE))
        .addGap(18, 18, 18)
        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addComponent(jLabel3)
            .addComponent(jScrollPane2, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(layout.createSequentialGroup())
                .addComponent(jLabel4)
                .addGap(53, 53, 53)
                .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                    .addComponent(jLabel5)
                    .addComponent(jComboBox2, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)))
            .addComponent(jComboBox1, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addComponent(jButton1, javax.swing.GroupLayout.PREFERRED_SIZE, 58,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addContainerGap(32, Short.MAX_VALUE)
    );

```

```

    pack();
} // </editor-fold>

```

```

private void type(javax.swing.event.ListSelectionEvent evt) {
    if (evt.getValueIsAdjusting() == false) {
        if (!jList1.isSelectionEmpty()) {
            typ= (String) jList1.getSelectedValue();
        }
    }
}

```

```

private void search(java.awt.event.ActionEvent evt) {

```

```

    if (!jList2.isSelectionEmpty()) {
        List aux1 = Arrays.asList(jList2.getSelectedValues());
        ArrayList aux = new ArrayList(aux1);
        for (Iterator<String> it2 = aux.iterator(); it2.hasNext();){
            String aux2= it2.next();

            if (!keylist.contains(aux2) && !"Any".equals(aux2) ) {
                if ("sodium-Chloride ".equals(aux2)&& !keylist.contains("NaCl") ) keylist.add("NaCl");
                if ("silicon-dioxide ".equals(aux2)&& !keylist.contains("SiO2")) keylist.add("SiO2");
            }
        }
    }
}

```

```

        if (!"sodium-Chloride ".equals(aux2) && !"silicon-dioxide ".equals(aux2) )
keylist.add(aux2);

    }

}

}

//Get the selected Lab
String aux1 = (String) jComboBox1.getSelectedItemAt();
if (!"All".equals(aux1)){
    if ("AGH-UST/Poland".equals(aux1)) {
        keylist.add("AGH University");
        keylist.add("of Science and Techniques");
    }
    if ("ARCS/Austria".equals(aux1)) keylist.add("Austrian Research Center Seibersdorf");
    if ("Cadarache/France-Harmonie".equals(aux1)) keylist.add("HARMONIE");
    if ("CEA/France".equals(aux1)){
        keylist.add("CEA");
    }
    if ("CERN/Switzerland".equals(aux1)) keylist.add("CERN");
    if ("FNS/JAEA-Japan".equals(aux1)) {
        keylist.add("FNS");
        keylist.add("Japan");
    }
    if ("FNG/ENEA-Italy".equals(aux1)){
        keylist.add("FNG");
        keylist.add("ENEA");
    }
    if ("FNG/Italy-TUD/Germany".equals(aux1)) keylist.add("FNG/TUD");
    if ("TUD/Germany".equals(aux1)) {
        keylist.add("TUD");
        keylist.add("Germany");
    }
    if ("FzK/Germany".equals(aux1)) keylist.add("Forschungszentrum Karlsruhe");
    if ("IPPE/Russia".equals(aux1)) {
        keylist.add("ippe");
        keylist.add("fzk");
    }
    if ("IRI-TUB".equals(aux1)) keylist.add(aux1);
    if ("ISPRA Univ. of Pavia/Italy-EURACOS II".equals(aux1)){
        keylist.add("EURACOS");
        keylist.add("Pavia");
    }
    if ("JAEA/Japan".equals(aux1)) keylist.add("Japan Atomic Energy Research Institute");
    if ("KEK/KENS-Japan".equals(aux1)) keylist.add("KENS");
    if ("LBNL/USA".equals(aux1)) keylist.add("Lawrence Berkeley Laboratory");
    if ("MEPhI/Russian Fed.".equals(aux1)) keylist.add("MEPhI");
    if ("MSU/USA".equals(aux1)) keylist.add("MSU");
    if ("NIRS/Japan".equals(aux1)) keylist.add("NIRS");
}

```

```

if ("NIST/USA".equals(aux1)) keylist.add("NIST");
if ("NRC/USA".equals(aux1)) keylist.add("NRC");
if ("NRI/Rez".equals(aux1)){
    keylist.add("NRI");
    keylist.add("Rez");
}
if ("ORNL/USA".equals(aux1)){
    keylist.add("ORNL");
    keylist.add("Oak Ridge");
}
if ("PSI/Swiss".equals(aux1)) {
    keylist.add("Swiss Institute for Nuclear");
    keylist.add("Research");
}
if ("RAL/England".equals(aux1)) keylist.add("Rutherford Appleton Laboratory");
if ("RDIPE/Russia".equals(aux1)) keylist.add("RDIPE");
if ("RFNC/Russia".equals(aux1)) keylist.add("RFNC-VNIITF");
if ("RIKEN/Japan".equals(aux1)) keylist.add("The Institute of Physical and Chemical
Research");
//if ("SNL/USA".equals(aux1)) keylist.add(""); VERRIFICAR O QUE FAZER
if ("SCK/CEN-Belgium".equals(aux1)) keylist.add("CEN/SCK");
if ("SEC NRS/FZR-Russia/Germany".equals(aux1)) {
    keylist.add("SEC NRS");
    keylist.add("Germany");
    keylist.add("FZR");
    keylist.add("Russia");
}
if ("University of Illinois/USA".equals(aux1)) keylist.add("University of Illinois");
if ("Univ. of Osaka/Japan-OKTAVIAN".equals(aux1)) {
    keylist.add("OKTAVIAN");
    keylist.add("Osaka");
}
if ("Univ. of Osaka/Japan-AVF Cyclotron".equals(aux1)) {
    keylist.add("AVF cyclotron");
    keylist.add("Osaka");
}
if ("Univ. of Tokyo/Japan-YAYOI".equals(aux1)) keylist.add("YAYOI");
if ("Univ. of Osaka and Univ. of Tokyo/Japan".equals(aux1)) keylist.add("University of Tokyo
and the University of Osaka");
if ("Winfrith/England-ASPIS".equals(aux1)) keylist.add("Winfrith");
if ("Wuerenlingen/Switzerland-PROTEUS".equals(aux1)) keylist.add("Wuerenlingen");
if ("Univ. of Tokyo/Japan-INS".equals(aux1)) {
    keylist.add("Institute for Nuclear Study");
    keylist.add("of University of Tokyo");
}
}
// Get the select geometry
String aux2 = (String) jComboBox2.getSelectedItemAt();
if (!"Any".equals(aux2)){
    if ("Sphere".equals(aux2)) keylist.add(aux2);
}

```

```

        if ("Cylindrical".equals(aux2)) keylist.add(aux2);
    }

    // Call the crawler

    if ("Reactor Shielding".equals(typ)) {
        CrawlerFission SinCrawler = new CrawlerFission();
        try {
            SinCrawler.Crawler(keylist);
        } catch (FileNotFoundException ex) {
            Logger.getLogger(SinbadInterface.class.getName()).log(Level.SEVERE, null, ex);
        } catch (IOException ex) {
            Logger.getLogger(SinbadInterface.class.getName()).log(Level.SEVERE, null, ex);
        }
    }

    if ("Fusion Neutronics Shielding".equals(typ)) {
        CrawlerFusion SinCrawler = new CrawlerFusion();
        try {
            SinCrawler.Crawler(keylist);
        } catch (FileNotFoundException ex) {
            Logger.getLogger(SinbadInterface.class.getName()).log(Level.SEVERE, null, ex);
        } catch (IOException ex) {
            Logger.getLogger(SinbadInterface.class.getName()).log(Level.SEVERE, null, ex);
        }
    }

    if ("Accelerator Shielding".equals(typ))
    {
        CrawlerAccelerator SinCrawler = new CrawlerAccelerator();
        try {
            SinCrawler.Crawler(keylist);
        } catch (FileNotFoundException ex) {
            Logger.getLogger(SinbadInterface.class.getName()).log(Level.SEVERE, null, ex);
        } catch (IOException ex) {
            Logger.getLogger(SinbadInterface.class.getName()).log(Level.SEVERE, null, ex);
        }
    }

    if ("All".equals(typ))
    {
        if (!keylist.isEmpty()){

            CrawlerAll SinCrawler = new CrawlerAll();
            try {
                SinCrawler.Crawler(keylist);
            } catch (FileNotFoundException ex) {
                Logger.getLogger(SinbadInterface.class.getName()).log(Level.SEVERE, null, ex);
            } catch (IOException ex) {
                Logger.getLogger(SinbadInterface.class.getName()).log(Level.SEVERE, null, ex);
            }
        }
    }

```

```

    }

    }else JOptionPane.showMessageDialog(null,"You need to set parameters for the search");

    }
    keylist.clear();
    jList1.setSelectedIndex(0);
    jList2.setSelectedIndex(0);
    jComboBox1.setSelectedIndex(0);
    jComboBox2.setSelectedIndex(0);

}

/**
 * @param args the command line arguments
 */
public static void main(String args[]) {

    /*
     * Set the Nimbus look and feel
     */
    //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">
    /*
     * If Nimbus (introduced in Java SE 6) is not available, stay with the
     * default look and feel. For details see
     * http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
     */
    try {
        for (javax.swing.UIManager.LookAndFeelInfo info :
javax.swing.UIManager.getInstalledLookAndFeels()) {
            if ("Nimbus".equals(info.getName())) {
                javax.swing.UIManager.setLookAndFeel(info.getClassName());
                break;
            }
        }
    } catch (ClassNotFoundException ex) {

java.util.logging.Logger.getLogger(SinbadInterface.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    } catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(SinbadInterface.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    } catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(SinbadInterface.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    } catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(SinbadInterface.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

```



```

    }
    //</editor-fold>
    /*
    * Create and display the form
    */
    java.awt.EventQueue.invokeLater(new Runnable() {

        @Override
        public void run() {
            new SinbadInterface().setVisible(true);
        }
    });
}

// Variables declaration - do not modify
private javax.swing.JButton jButton1;
private javax.swing.JButton jButton2;
private javax.swing.JComboBox jComboBox1;
private javax.swing.JComboBox jComboBox2;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel2;
private javax.swing.JLabel jLabel3;
private javax.swing.JLabel jLabel4;
private javax.swing.JLabel jLabel5;
private javax.swing.JList jList1;
private javax.swing.JList jList2;
private javax.swing.JScrollPane jScrollPane1;
private javax.swing.JScrollPane jScrollPane2;
// End of variables declaration
}

```

CrawlerFission.java (The search for the Reactor Shielding benchmarks type)

```

import java.io.*;
import java.util.ArrayList;
import java.util.Iterator;
import java.util.regex.Matcher;
import java.util.regex.Pattern;
import javax.swing.JOptionPane;

public class CrawlerFission {

    void Crawler (ArrayList key) throws FileNotFoundException, IOException {
        File arquivo;
        String sourceLine;
        String content = "";
        String ender = "";
        ArrayList keylist = key;
    }
}

```

```

String str;
ArrayList found= new ArrayList();
boolean bool=true;

try {
    File dir = null;
    String home = System.getProperty("user.home");
    File ff = new File(home);
    String dire = ff.getAbsolutePath()+File.separator+".Sinbad";

    if (!(new File(dire+File.separator+"fission.txt")).exists()) {

        dir = new File(dire);
        dir.mkdir();
        arquivo = new File(dir,"fission.txt");
        FileOutputStream fos = new FileOutputStream(arquivo);
        String ad = new File("").getAbsolutePath();
        ad=ad.replace("\\", "/");

        String gg = ad+File.separator+"Docs"+File.separator+"indexfission.htm";

        InputStreamReader pageInput = new InputStreamReader(new FileInputStream(gg));
        BufferedReader source = new BufferedReader(pageInput);

        // Append each new HTML line into one string. Add a tab character.
        while ((sourceLine = source.readLine()) != null)
            content += sourceLine + "\t";

        // Remove style tags & inclusive content
        Pattern path = Pattern.compile("<a href=.*?>.??</a>");
        Matcher mpath = path.matcher(content);
        while (mpath.find ()) ender += mpath + "\n";

        Pattern path2 = Pattern.compile("<a href=");
        Matcher mpath2 = path2.matcher(ender);
        while (mpath2.find()) ender = mpath2.replaceAll("");

        Pattern path3 = Pattern.compile("</a>");
        Matcher mpath3 = path3.matcher(ender);
        while (mpath3.find()) ender = mpath3.replaceAll("");

        Pattern path4 = Pattern.compile("<span.*?>");
        Matcher mpath4 = path4.matcher(ender);
        while (mpath4.find()) ender = mpath4.replaceAll("");

        Pattern path7 = Pattern.compile("</span>");
        Matcher mpath7 = path7.matcher(ender);
        while (mpath7.find()) ender = mpath7.replaceAll("");
    }
}

```

```

        fos.write(ender.getBytes());
        fos.close();
    }

    BufferedReader in = new BufferedReader(new
FileReader(home+File.separator+".Sinbad"+File.separator+"fission.txt"));

    if (!keylist.isEmpty())
    {
        for (Object keyword : keylist){
            if ((found==null || found.isEmpty())&& bool){

                while (in.ready()) {
                    String content2="";
                    str = in.readLine();
                    int ind2 = str.lastIndexOf("]");
                    int ind= str.indexOf(">[");
                    str=str.replace(str.subSequence(ind, ind2+1),"");
                    str=str.replace("java.util.regex.Matcher[pattern=.*?>.*? region=0,29264 lastmatch=",
"");

                    str=str.replaceAll("\\", "");
                    String ad = new File("").getAbsolutePath();
                    ad=ad.replace("\\", "/");

                    String gg = ad+File.separator+"d237mnycp01"+File.separator+str;

                    // Open the address and create a BufferedReader with the source code.
                    InputStreamReader pageInput = new InputStreamReader(new FileInputStream(gg));
                    BufferedReader source = new BufferedReader(pageInput);

                    // Append each new HTML line into one string. Add a tab character.

                    while ((sourceLine = source.readLine()) != null){
                        content2 += sourceLine + "\t";

                    }
                    Pattern path5 = Pattern.compile((String)keyword);
                    Matcher mpath5 = path5.matcher(content2);
                    while (mpath5.find()) {
                        if (!found.contains(str)){
                            found.add(str);
                        }
                    }
                }
            }
        }
    }
    else{// if ArrayList found is not empty
        bool=false;// if the list became empty after the search the cod will not go back
        for (Iterator<String> it = found.iterator(); it.hasNext();){

            String content2="";
            String aux = it.next();

```

```

String ad = new File("").getAbsolutePath();
ad=ad.replace("\\", "/");

String gg = ad+File.separator+"d237mnycp01"+File.separator+aux;

// Open the address and create a BufferedReader with the source code.
InputStreamReader pageInput = new InputStreamReader(new FileInputStream(gg));
BufferedReader source = new BufferedReader(pageInput);

// Append each new HTML line into one string. Add a tab character.

while ((sourceLine = source.readLine()) != null)
    content2 += sourceLine + "\t";

Pattern path6 = Pattern.compile((String)keyword);
Matcher mpath6 = path6.matcher(content2);
if (!mpath6.find()) it.remove();

    } //end intern for
    } //end else

} //end for
} else {
    while (in.ready()) {
        str = in.readLine();
        int ind2 = str.lastIndexOf("]");
        int ind= str.indexOf(">[");
        str=str.replace(str.subSequence(ind, ind2+1), "");
        str=str.replace("java.util.regex.Matcher[pattern=.*?>.*? region=0,29264 lastmatch=", "");
        str=str.replaceAll("\\", "");
        found.add(str);
    }
}

} // end try

catch (Exception ee) {
    ee.printStackTrace();
}
if (found.isEmpty()) {
    JOptionPane.showMessageDialog(null, "Your search did not match any document");

} else {
    Output Output = new Output();
    Output.results("fission", found);
}
}
}

```

CrawlerFusion.java (The search for the Fusion Neutronics Shielding benchmarks type)

```
import java.io.*;
import java.util.ArrayList;
import java.util.Iterator;
import java.util.regex.Matcher;
import java.util.regex.Pattern;
import javax.swing.JOptionPane;
```

```
public class CrawlerFusion {
```

```
void Crawler (ArrayList key) throws FileNotFoundException, IOException {
```

```
    File arquivo;
    String sourceLine;
    String content = "";
    String ender = "";
    ArrayList keylist = key;
    String str;
    ArrayList found= new ArrayList();
    boolean bool=true;
```

```
try {
```

```
    File dir = null;
    String home = System.getProperty("user.home");
    File ff = new File(home);
    String dire = ff.getAbsolutePath()+File.separator+".Sinbad";
```

```
if (!(new File(dire+File.separator+"fusion.txt")).exists()) {
```

```
    dir = new File(dire);
    dir.mkdir();
    arquivo = new File(dir,"fusion.txt");
    FileOutputStream fos = new FileOutputStream(arquivo);
    String ad = new File("").getAbsolutePath();
    ad=ad.replace("\\", "/");
```

```
//URL address = new URL("file:///"+ad+"/Docs/index.htm");
String gg = ad+File.separator+"Docs"+File.separator+"indexfusion.htm";
```

```
// Open the address and create a BufferedReader with the source code.
//InputStreamReader pageInput = new InputStreamReader(address.openStream());
InputStreamReader pageInput = new InputStreamReader(new FileInputStream(gg));
BufferedReader source = new BufferedReader(pageInput);
```

```
// Append each new HTML line into one string. Add a tab character.
while ((sourceLine = source.readLine()) != null)
    content += sourceLine + "\t";
```

```
// Remove style tags & inclusive content
```

```

Pattern path = Pattern.compile("<a href=.*?>.??</a>");
Matcher mpath = path.matcher(content);
while (mpath.find ()) ender += mpath + "\n";

Pattern path2 = Pattern.compile("<a href=");
Matcher mpath2 = path2.matcher(ender);
while (mpath2.find()) ender = mpath2.replaceAll("");

Pattern path3 = Pattern.compile("</a>");
Matcher mpath3 = path3.matcher(ender);
while (mpath3.find()) ender = mpath3.replaceAll("");

Pattern path4 = Pattern.compile("<span.*?>");
Matcher mpath4 = path4.matcher(ender);
while (mpath4.find()) ender = mpath4.replaceAll("");

Pattern path7 = Pattern.compile("</span>");
Matcher mpath7 = path7.matcher(ender);
while (mpath7.find()) ender = mpath7.replaceAll("");

fos.write(ender.getBytes());
fos.close();
}

BufferedReader in = new BufferedReader(new
FileReader(home+File.separator+".Sinbad"+File.separator+"fusion.txt"));

if (!keylist.isEmpty())
{
for (Object keyword : keylist){
if ((found==null || found.isEmpty())&& bool){

while (in.ready()) {
String content2="";
str = in.readLine();
int ind2 = str.lastIndexOf("]");
int ind= str.indexOf(">[");
str=str.replace(str.subSequence(ind, ind2+1),"");
str=str.replace("java.util.regex.Matcher[pattern=.*?>.?? region=0,26920 lastmatch=",
");

str=str.replaceAll("\\"", "");
String ad = new File("").getAbsolutePath();
ad=ad.replace("\\", "/");

String gg = ad+File.separator+"d237mnycp01"+File.separator+str;

// Open the address and create a BufferedReader with the source code.
InputStreamReader pageInput = new InputStreamReader(new FileInputStream(gg));
BufferedReader source = new BufferedReader(pageInput);

```

```

// Append each new HTML line into one string. Add a tab character.

while ((sourceLine = source.readLine()) != null){
    content2 += sourceLine + "\t";
}
Pattern path5 = Pattern.compile((String)keyword);
Matcher mpath5 = path5.matcher(content2);
while (mpath5.find()) {
    if (!found.contains(str)){
        found.add(str);
    }
}
}

}else{// if ArrayList found is not empty
bool=false;// if the list became empty after the search the cod will not go back
for (Iterator<String> it = found.iterator(); it.hasNext();){

String content2="";
String aux = it.next();
String ad = new File("").getAbsolutePath();
ad=ad.replace("\\", "/");

String gg = ad+File.separator+"d237mnycp01"+File.separator+aux;

// Open the address and create a BufferedReader with the source code.
InputStreamReader pageInput = new InputStreamReader(new FileInputStream(gg));
BufferedReader source = new BufferedReader(pageInput);

// Append each new HTML line into one string. Add a tab character.

while ((sourceLine = source.readLine()) != null)
    content2 += sourceLine + "\t";

Pattern path6 = Pattern.compile((String)keyword);
Matcher mpath6 = path6.matcher(content2);
if (!mpath6.find()) it.remove();

}end intern for
}end else

}end for
}else{
while (in.ready()) {
    str = in.readLine();
    int ind2 = str.lastIndexOf("]");
    int ind= str.indexOf(">[");
    str=str.replace(str.subSequence(ind, ind2+1), "");
    str=str.replace("java.util.regex.Matcher[pattern=.*?>.*? region=0,26920 lastmatch=", "");
}
}

```

```

        str=str.replaceAll("\\", "");
        found.add(str);
    }
}

} // end try

catch (Exception ee) {
    ee.printStackTrace();
}
if (found.isEmpty()) {
    JOptionPane.showMessageDialog(null, "Your search did not match any document");

}
else {

    Output Output = new Output();
    Output.results("fusion",found);

}
}
}

```

CrawlerAccelerator.java (The search for the Accelerator Shielding benchmarks type)

```

import java.io.*;
import java.util.ArrayList;
import java.util.Iterator;
import java.util.regex.Matcher;
import java.util.regex.Pattern;
import javax.swing.JOptionPane;

public class CrawlerAccelerator {

    void Crawler (ArrayList key) throws FileNotFoundException, IOException {
        File arquivo;
        String sourceLine;
        String content = "";
        String ender = "";
        ArrayList keylist = key;
        String str;
        ArrayList found= new ArrayList();
        boolean bool=true;

        try {
            File dir = null;
            String home = System.getProperty("user.home");
            File ff = new File(home);

```



```

String dire = ff.getAbsolutePath()+File.separator+".Sinbad";

if (!(new File(dire+File.separator+"accelerator.txt")).exists()) {

    dir = new File(dire);
    dir.mkdir();
    arquivo = new File(dir,"accelerator.txt");
    FileOutputStream fos = new FileOutputStream(arquivo);
    String ad = new File("").getAbsolutePath();
    ad=ad.replace("\\", "/");

    String gg = ad+File.separator+"Docs"+File.separator+"indexaccelerator.htm";

    // Open the address and create a BufferedReader with the source code.
    //InputStreamReader pageInput = new InputStreamReader(address.openStream());
    InputStreamReader pageInput = new InputStreamReader(new FileInputStream(gg));
    BufferedReader source = new BufferedReader(pageInput);

    // Append each new HTML line into one string. Add a tab character.
    while ((sourceLine = source.readLine()) != null)
        content += sourceLine + "\t";

    // Remove style tags & inclusive content
    Pattern path = Pattern.compile("<a href=.*?>.??</a>");
    Matcher mpath = path.matcher(content);
    while (mpath.find ()) ender += mpath + "\n";

    Pattern path2 = Pattern.compile("<a href=");
    Matcher mpath2 = path2.matcher(ender);
    while (mpath2.find()) ender = mpath2.replaceAll("");

    Pattern path3 = Pattern.compile("</a>");
    Matcher mpath3 = path3.matcher(ender);
    while (mpath3.find()) ender = mpath3.replaceAll("");

    Pattern path4 = Pattern.compile("<span.*?>");
    Matcher mpath4 = path4.matcher(ender);
    while (mpath4.find()) ender = mpath4.replaceAll("");

    Pattern path7 = Pattern.compile("</span>");
    Matcher mpath7 = path7.matcher(ender);
    while (mpath7.find()) ender = mpath7.replaceAll("");

    fos.write(ender.getBytes());
    fos.close();
}

BufferedReader in = new BufferedReader(new
FileReader(home+File.separator+".Sinbad"+File.separator+"accelerator.txt"));

```

```

if (!keylist.isEmpty())
{
    for (Object keyword : keylist){
        if ((found==null || found.isEmpty())&& bool){

            while (in.ready() {
                String content2="";
                str = in.readLine();
                int ind2 = str.lastIndexOf("]");
                int ind= str.indexOf(">[");
                str=str.replace(str.subSequence(ind, ind2+1),"");
                str=str.replace("java.util.regex.Matcher[pattern=.*?>.*? region=0,27437 lastmatch=",
                "");

                str=str.replaceAll("\\\"", "");
                String ad = new File("").getAbsolutePath();
                ad=ad.replace("\\", "/");

                String gg = ad+File.separator+"d237mnycp01"+File.separator+str;

                // Open the address and create a BufferedReader with the source code.
                InputStreamReader pageInput = new InputStreamReader(new FileInputStream(gg));
                BufferedReader source = new BufferedReader(pageInput);

                // Append each new HTML line into one string. Add a tab character.

                while ((sourceLine = source.readLine()) != null){
                    content2 += sourceLine + "\t";
                }
                Pattern path5 = Pattern.compile((String)keyword);
                Matcher mpath5 = path5.matcher(content2);
                while (mpath5.find()) {
                    if (!found.contains(str)){
                        found.add(str);
                    }
                }
            }
        }
    }
}
}

}else{// if ArrayList found is not empty
    bool=false;// if the list became empty after the search the cod will not go back
    for (Iterator<String> it = found.iterator(); it.hasNext();){

        String content2="";
        String aux = it.next();
        String ad = new File("").getAbsolutePath();
        ad=ad.replace("\\", "/");

        String gg = ad+File.separator+"d237mnycp01"+File.separator+aux;

        // Open the address and create a BufferedReader with the source code.
        InputStreamReader pageInput = new InputStreamReader(new FileInputStream(gg));
        BufferedReader source = new BufferedReader(pageInput);
    }
}

```

```

// Append each new HTML line into one string. Add a tab character.

while ((sourceLine = source.readLine()) != null)
    content2 += sourceLine + "\t";

Pattern path6 = Pattern.compile((String)keyword);
Matcher mpath6 = path6.matcher(content2);
if (!mpath6.find()) it.remove();

    }//end intern for
    }//end else

} //end for
} else {
    while (in.ready()) {
        str = in.readLine();
        int ind2 = str.lastIndexOf("]");
        int ind= str.indexOf(">[");
        str=str.replace(str.subSequence(ind, ind2+1), "");
        str=str.replace("java.util.regex.Matcher[pattern=.*?>.*? region=0,27437 lastmatch=", "");
        str=str.replaceAll("\\"", "");
        found.add(str);
    }
}

} // end try

catch (Exception ee) {
    ee.printStackTrace();
}
if (found.isEmpty()) {
    JOptionPane.showMessageDialog(null, "Your search did not match any document");
}
else {

    Output Output = new Output();
    Output.results("accelerator",found);

}
}
}

```

CrawlerAll.java (The search All types of benchmarks)

```

import java.io.*;
import java.util.ArrayList;
import java.util.Iterator;
import java.util.regex.Matcher;
import java.util.regex.Pattern;

```

```

import javax.swing.JOptionPane;

public class CrawlerAll {

    void Crawler (ArrayList key) throws FileNotFoundException, IOException {
        File arquivo;
        String sourceLine;
        String content = "";
        String ender = "";
        ArrayList keylist = key;
        String str;
        ArrayList found= new ArrayList();
        boolean bool=true;

        try {
            File dir = null;
            String home = System.getProperty("user.home");
            File ff = new File(home);
            String dire = ff.getAbsolutePath()+File.separator+".Sinbad";

            if (!(new File(dire+File.separator+"file.txt")).exists()) {

                dir = new File(dire);
                dir.mkdir();
                arquivo = new File(dir, "file.txt");
                FileOutputStream fos = new FileOutputStream(arquivo);
                String ad = new File("").getAbsolutePath();
                ad=ad.replace("\\", "/");

                String gg = ad+File.separator+"Docs"+File.separator+"index.htm";

                InputStreamReader pageInput = new InputStreamReader(new FileInputStream(gg));
                BufferedReader source = new BufferedReader(pageInput);

                // Append each new HTML line into one string. Add a tab character.
                while ((sourceLine = source.readLine()) != null)
                    content += sourceLine + "\t";

                // Remove style tags & inclusive content
                Pattern path = Pattern.compile("<a href=.*?>.*?</a>");
                Matcher mpath = path.matcher(content);
                while (mpath.find ()) ender += mpath + "\n";

                Pattern path2 = Pattern.compile("<a href=");
                Matcher mpath2 = path2.matcher(ender);
                while (mpath2.find()) ender = mpath2.replaceAll("");
            }
        }
    }
}

```

```

Pattern path3 = Pattern.compile("</a>");
Matcher mpath3 = path3.matcher(ender);
while (mpath3.find()) ender = mpath3.replaceAll("");

Pattern path4 = Pattern.compile("<span.*?>");
Matcher mpath4 = path4.matcher(ender);
while (mpath4.find()) ender = mpath4.replaceAll("");

Pattern path7 = Pattern.compile("</span>");
Matcher mpath7 = path7.matcher(ender);
while (mpath7.find()) ender = mpath7.replaceAll("");

fos.write(ender.getBytes());
fos.close();
}

BufferedReader in = new BufferedReader(new
FileReader(home+File.separator+".Sinbad"+File.separator+"file.txt"));

if (!keylist.isEmpty())
{
for (Object keyword : keylist){
if ((found==null || found.isEmpty())&& bool){

while (in.ready()) {
String content2="";
str = in.readLine();
int ind2 = str.lastIndexOf("]");
int ind= str.indexOf(">[");
str=str.replace(str.subSequence(ind, ind2+1),"");
str=str.replace("java.util.regex.Matcher[pattern=.*?>.*? region=0,35883 lastmatch=",
");

str=str.replaceAll("\\\"", "");
String ad = new File("").getAbsolutePath();
ad=ad.replace("\\", "/");

String gg = ad+File.separator+"d237mnycp01"+File.separator+str;

// Open the address and create a BufferedReader with the source code.
InputStreamReader pageInput = new InputStreamReader(new FileInputStream(gg));
BufferedReader source = new BufferedReader(pageInput);

// Append each new HTML line into one string. Add a tab character.

while ((sourceLine = source.readLine()) != null){
content2 += sourceLine + "\t";
}

Pattern path5 = Pattern.compile((String)keyword);
Matcher mpath5 = path5.matcher(content2);
while (mpath5.find()) {

```

```

        if (!found.contains(str)){
            found.add(str);
        }
    }
}

}else{// if ArrayList found is not empty
    bool=false;// if the list became empty after the search the cod will not go back
    for (Iterator<String> it = found.iterator(); it.hasNext();){

        String content2="";
        String aux = it.next();
        String ad = new File("").getAbsolutePath();
        ad=ad.replace("\\", "/");

        String gg = ad+File.separator+"d237mnycp01"+File.separator+aux;

        // Open the address and create a BufferedReader with the source code.
        InputStreamReader pageInput = new InputStreamReader(new FileInputStream(gg));
        BufferedReader source = new BufferedReader(pageInput);

        // Append each new HTML line into one string. Add a tab character.

        while ((sourceLine = source.readLine()) != null)
            content2 += sourceLine + "\t";

        Pattern path6 = Pattern.compile((String)keyword);
        Matcher mpath6 = path6.matcher(content2);
        if (!mpath6.find()) it.remove();

    }//end intern for
    }//end else

} //end for
}

} // end try

catch (Exception ee) {
    ee.printStackTrace();
}
if (found.isEmpty()) {
    JOptionPane.showMessageDialog(null,"Your search did not match any document");
}
else {
    Output Output = new Output();
    Output.results("all",found);
}
}

```

```
}  
}
```

Output.java (Organize the results)

```
import java.awt.Dimension;  
import java.io.*;  
import java.util.ArrayList;  
import java.util.Iterator;  
  
public class Output {  
    String line;  
    ArrayList names = new ArrayList();  
    ArrayList path2 = new ArrayList();  
    public boolean lock;  
  
    void results (String nam,ArrayList path) throws FileNotFoundException, IOException {  
        String home = System.getProperty("user.home");  
        String name=nam;  
        path2=path;  
        for (Iterator<String> it = path2.iterator(); it.hasNext();){  
            String aux = it.next();  
  
            if ("all".equals(name)){  
                BufferedReader in = new BufferedReader(new  
FileReader(home+File.separator+".Sinbad"+File.separator+"file.txt"));  
  
                while (in.ready()) {  
                    line= in.readLine();  
                    int first = line.indexOf(aux);  
                    if (first!= -1){  
                        String aux2 = line.substring(line.indexOf("]")+1, line.lastIndexOf("]"));  
                        CrawlerInputs crawlerinputs = new CrawlerInputs();  
                        aux2+=crawlerinputs.Inputs(aux);  
                        names.add(aux2);  
                        break;  
                    }  
                }  
            }  
            if ("accelerator".equals(name)){  
                String os= (String) System.getProperties().get("os.name");  
                BufferedReader in = new BufferedReader(new  
FileReader(home+File.separator+".Sinbad"+File.separator+"accelerator.txt"));  
  
                while (in.ready()) {  
                    line= in.readLine();
```

```

        int first = line.indexOf(aux);
        if (first!= -1){
            String aux2 = line.substring(line.indexOf("]")+1, line.lastIndexOf("]"));
            CrawlerInputs crawlerinputs = new CrawlerInputs();
            aux2+=crawlerinputs.Inputs(aux);
            names.add(aux2);
            break;
        }
    }
}
if ("fusion".equals(name)){
    BufferedReader in = new BufferedReader(new
FileReader(home+File.separator+".Sinbad"+File.separator+"fusion.txt"));

    while (in.ready()) {
        line= in.readLine();
        int first = line.indexOf(aux);
        if (first!= -1){
            String aux2 = line.substring(line.indexOf("]")+1, line.lastIndexOf("]"));
            CrawlerInputs crawlerinputs = new CrawlerInputs();
            aux2+=crawlerinputs.Inputs(aux);

            names.add(aux2);
            break;
        }
    }
}
if ("fission".equals(name)){
    BufferedReader in = new BufferedReader(new
FileReader(home+File.separator+".Sinbad"+File.separator+"fission.txt"));

    while (in.ready()) {
        line= in.readLine();
        int first = line.indexOf(aux);
        if (first!= -1){
            String aux2 = line.substring(line.indexOf("]")+1, line.lastIndexOf("]"));
            CrawlerInputs crawlerinputs = new CrawlerInputs();
            aux2+=crawlerinputs.Inputs(aux);
            names.add(aux2);
            break;
        }
    }
}
}
Results Resultado = new Results(names,path2);

Dimension d = new Dimension();
d.setSize(800, 500);
Resultado.setMinimumSize(d);
Resultado.setVisible(true);

```



```
}  
}
```

CrawlerInputs.java (Returns the kind of input the benchmarks has)

```
import java.io.BufferedReader;  
import java.io.File;  
import java.io.FileReader;  
import java.io.IOException;  
import java.util.regex.Matcher;  
import java.util.regex.Pattern;  
  
public class CrawlerInputs {  
    String sourceLine;  
    String content="";  
  
    public String Inputs(String adress1) throws IOException{  
        String nm="";  
        String ad = new File("").getAbsolutePath();  
        ad=ad.replace("\\", "/");  
  
        BufferedReader in = new BufferedReader(new FileReader(ad+"/d237mnycp01/"+adress1));  
  
        while (in.ready()) {  
            String line = in.readLine();  
            Pattern path5 = Pattern.compile("\\.inp");  
            Matcher mpath5 = path5.matcher(line);  
            if (mpath5.find()) {  
                CrawlerInputs inp= new CrawlerInputs();  
                if (nm.indexOf(inp.nameInputs(line))==-1) nm += inp.nameInputs(line)+",";  
  
            }  
        }  
        if ("".equals(nm)) return "";  
        else {  
            nm=nm+""; ;  
            nm=nm.replace(",","");  
            return "          (" +nm;  
  
        }  
    }  
  
    public String nameInputs(String line) throws IOException{  
  
        String method="";  
        Pattern path = Pattern.compile("Input data for");  
        Matcher mpath = path.matcher(line);  
        if (mpath.find()) {  
            if (line.indexOf("Input data for")!= -1){  
                int begin=line.indexOf("Input data for")+14;
```

```

        int end = line.indexOf(" ", begin+1);
        method = line.substring(begin, end);
    }else{
        int begin=line.indexOf("Input Data for")+14;
        int end = line.indexOf(" ", begin+1);
        method = line.substring(begin, end);
    }
}

return method;
}
}

```

Results.java (The results interface)

```

import java.awt.BorderLayout;
import java.awt.Desktop;
import java.awt.Font;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.io.File;
import java.io.IOException;
import java.net.URI;
import java.net.URISyntaxException;
import java.net.URLDecoder;
import java.util.ArrayList;
import java.util.Iterator;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.swing.*.*;

public class Results extends JDialog implements ActionListener {

    JLabel label;
    JList list;
    JButton open;
    ArrayList bench = new ArrayList();
    ArrayList ad = new ArrayList();
    JScrollPane scroll;

    public Results(ArrayList nm, ArrayList path){
        this.setModal(true);
        this.setTitle("SINBAD RESULTS");

        bench = nm;
        ad= path;
    }
}

```

```

setLayout(new BorderLayout());

open = new JButton("Open");
label = new JLabel("BENCHMARKS");
label.setFont(new Font("Serife",Font.BOLD,16));
DefaultListModel model = new DefaultListModel();
list = new JList(model);
for (Iterator<String> it = bench.iterator(); it.hasNext();){
    model.addElement(it.next());

}
scroll= new JScrollPane(list);
add(scroll,BorderLayout.CENTER);
add(label,BorderLayout.PAGE_START);
add(open,BorderLayout.PAGE_END);

open.addActionListener(this);

}

@Override
public void actionPerformed(ActionEvent ae) {

    String end = (String) ad.get(list.getSelectedIndex());

    try {
        try {
            if (System.getProperty("os.name").contains("Windows")){
                String ad2 = new File("").getAbsolutePath();
                ad2=ad2.replace("\\", "/");
                ad2=ad2+"/d237mnycp01/"+end;
                Runtime r = Runtime.getRuntime();
                Process p = r.exec("rundll32 url.dll,FileProtocolHandler "+ad2);
            }else{
                String ad2 = new File("").getAbsolutePath();
                ad2=ad2.replace("\\", "/");
                ad2=ad2+"/d237mnycp01/"+end;
                ad2="file:///"+ad2;
                String uri = URLDecoder.decode(ad2);
                System.out.println("Have: "+uri);
                Desktop.getDesktop().browse(new URI(uri));
            }
        } catch (URISyntaxException ex) {
            Logger.getLogger(Results.class.getName()).log(Level.SEVERE, null, ex);
        }
        } catch (IOException ex) {
            Logger.getLogger(Results.class.getName()).log(Level.SEVERE, null, ex);
        }
    }
}
}
}

```

