

UTILIZATION OF JAVA REFLECTION  
IN DETECTING OBJECT CONCEPT SIMILARITIES

A master project submitted to the Graduate School of in partial  
fulfillment of the requirement for the degree of  
Master of Science (Information Technology) (MSc.IT)  
Universiti Utara Malaysia.

by

TAN CHOO JUN



**Sekolah Siswazah  
(Graduate School)  
Universiti Utara Malaysia**

**PERAKUAN KERJA KERTAS PROJEK  
(Certification of Project Paper)**

Saya, yang bertandatangan, memperakukan bahawa  
*(I, the undersigned, certify that)*

**TAN CHOO JUN**

calon untuk Ijazah \_\_\_\_\_  
*(candidate for the degree of)* Master of Science (Information Technology)

telah mengemukakan kertas projek yang bertajuk  
*(has presented his/her project paper of the following title)*

**UTILIZATION OF JAVA REFLECTION**


**IN DETECTING OBJECT CONCEPT SIMILARITIES**

seperti yang tercatat di muka surat tajuk dan kulit kertas projek  
*(as it appears on the title page and front cover of project paper)*

bahawa kertas projek tersebut boleh diterima dari segi bentuk serta kandungan,  
dan meliputi bidang ilmu dengan memuaskan.  
*(that the project paper acceptable in form and content, and that a satisfactory  
knowledge of the field is covered by the project paper).*

Nama Penyelia \_\_\_\_\_  
*(Name of Supervisor):* Mohd. Zamberi Saad

Tandatangan  
*(Signature)*

  
\_\_\_\_\_

Tarikh  
*(Date)*

8<sup>th</sup> March 2000  
\_\_\_\_\_

## **PERMISSION TO USE**

In presenting this master project in partial fulfillment of the requirements for a degree of Master of Information Technology from Universiti Utara Malaysia, I agree that the University Library may make it freely available for inspection. I also grant permission for copying of this project in any manner, in completely or in part, for scholarly purposes. In my absence, this may be granted by the lecturers who supervised my project or by the Dean of Graduate School. It is understood that any copying or publication or use of this project or parts thereof for financial gain not be allowed without my written permission. It is also understood that due recognition shall be given to me and to Universiti Utara Malaysia for any scholarly use which may be made of any material from my project.

Requests for permission to copy or to make other use of materials in this project, in completely or in part, should be addressed to:

**Dean of Graduate School  
Universiti Utara Malaysia  
06010 UUM Sintok  
Kedah Darul Aman**

## ABSTRAK

Projek ini membangunkan aplikasi “Java Reflection”. Ia menggunakan package system java.lang.reflect yang dibangunkan oleh Sun Microsystem dalam JDK versi 1.2 dan ke atas. “Reflection”, juga dikenali sebagai “Introspection”, mempunyai keupayaan “melihat kandungan” sesuatu kelas atau objek. Ia digunakan untuk mengekplotasi kandungan fail kelas. Dengan bantuan enjin penganalisa, aplikasi yang dibangunkan berkeupayaan menghasilkan maklumat kesamaan antara objek tanpa merujuk kepada kod asal. Metodologi Object-Oriented, khususnya Teknik Permodelan Objek, digunakan untuk membangunkan aplikasi ini. Terdapat empat tahap untuk dirujuk dalam methodologi ini, iaitu analisis, rekabentuk sistem, rekabentuk objek, dan implementasi. Inputnya adalah fail objek Java, dan output aplikasi mengandungi maklumat persamaan pada fail-fail input. Maklumat objek dibahagikan kepada lima kategori, iaitu Modifier, Interface, Field, Method, dan Constructor. Sistem akan menghasilkan maklumat kesamaan di antara dua fail objek untuk setiap kategori, termasuk kategori yang digunakan dan dikemukakan. Ke kerapian maklumat kesamaan juga merupakan sub komponen output terperinci sistem. Kesimpulannya, aplikasi ini adalah alat alternatif untuk membandingkan sekumpulan fail objek dengan pantas dan menghasilkan output yang mudah difahami. Contoh kegunaan aplikasi ini adalah sebagai satu alat membantu pensyarah menilai tugas pelajar dengan satu model ideal dan kriteria penilaian yang malar. Selain itu, ia juga sesuai digunakan untuk membandingkan tahap kesamaan antara dua jawapan pelajar bagi mengesan plagiarisme.

## ABSTRACT

This project is about developing a Java reflection application. It utilizes the reflection features in the system package `java.lang.reflect` of Sun Microsystems' JDK version 1.2 and above. Reflection, also named as Introspection, has the ability to "look inside" a class or an object (Lemay, 1996). It uses to explore the content of the class files. With the help of the analyzer engine, the developed application is capable to produce similarity object's information between the inputs without referring to the source code. The Object-Oriented Methodology, specifically the Object Modeling Technique, is used to develop this Reflection Application. There are four stages involving analysis, system design, object design, and implementation that are followed in this methodology. The input is the Java object files, and the output contains of similarity information of those object files. The object's information is divided into five categories including Modifier, Interface, Field, Method, and Constructor. The system address the similar information for each category between two object files to the user, which including the similar used and declared category. The similar items' frequency will also be an element of the system's detail output. As a conclusion, this application is an alternative tool to compare a group of object files in fast mode with readable result in application's output. The example of the application usage is as a contributing tool to help lecturers to evaluate student assignments with an ideal model answer with constant evaluating criteria requirements. It is also suitable to be used in determining student plagiarism.

## ACKNOWLEDGEMENTS

Many people contributed to the successful completion of my project at Universiti Utara Malaysia. First, I would like to express deep gratitude to my supervisor Mohd. Zamberi Saad for his valuable guidance and advice, which contributed substantially towards the completion of this study. My gratitude is also extended to my course coordinator, Associate Professor Dr. Wan Rozaini Sheik Osman, for all the help extended and meticulously undergoing the project, and giving numerous suggestions in overall presentation.

Special thanks give to Associate Professor Haji Abdul Razak Ismail, deputy Dean of Language and Scientific Thinking School, who guide me in completing this academic writing in Information Technology's field. Also, not forgetting the former Dean of Information Technology School, Associate Professor Shahrum Hashim, Dr. Mubarak Rahamathulla Ali, and William Koh Siew Yan who inspired me to explore various fields in research, and finally undertake this programming area of the project. I am grateful to Mr. Wong Chee Onn, lecturer of Information Technology School, for his kindness to correct this report's grammars and spelling mistakes.

Finally, I express a deep sense of gratitude to my family and my dearest Joyce Ong for their valuable untiring and moral support.

## CONTENT PAGE

TITLE PAGE	
CERTIFICATION OF PROJECT PAPER	
PERMISSION TO USE	
ABSTRAK	i
ABSTRACT	ii
ACKNOWLEDGEMENTS	iii
CONTENT PAGE	iv
	v
CHAPTER ONE : INTRODUCTION	1
CHAPTER TWO : THE PURPOSE AND PROBLEM STATEMENT	2
2.1 THE BACKGROUND OF THE SOLUTION	3
2.2 THE REFLECTION	5
2.2.1 THE USE OF REFLECTION	6
2.2.2 THE SUPPORT VERSION OF JDK FOR REFLECTION	6
2.2.3 THE CONTENT OF <code>java.lang.reflect</code>	6
CHAPTER THREE : THE PROJECT DEFINITION	8
3.1 THE NATURE OF APPLICATION	8
3.2 THE PROJECT ASSUMPTION	11
3.3 THE SCOPE AND OBJECTIVE	12
CHAPTER FOUR : THE LITERATURE REVIEW	13
CHAPTER FIVE : THE METHODOLOGY	18
5.1 THE CONCEPTUAL DEVELOPMENT	18
A) ANALISIS	19
B) SYSTEM DESIGN	28
C) OBJECT DESIGN	32
D) IMPLEMENTATION	35
5.2 THE LOGICAL DEVELOPMENT	37
5.3 THE EXPECTED INPUT AND OUTPUT	48
CHAPTER SIX : CONCLUSION	49
6.1 THE CONTRIBUTION OF PROJECT	49
6.2 THE CONTRIBUTION FOR FUTURE STUDY	51
6.3 THE LIMITATION	52

BIBLIOGRAPHY	54
APPENDIX A: SAMPLE OUTPUT WITH DISCRPTION	56
APPENDIX B: USER MANUAL	63
APPENDIX C: FIGURE / DIAGRAM	66
FIGURE 1: OBJECT MODEL DIAGRAM	67
FIGURE 2: EVENT TRACE FOR REFLECTION APPLICATION	68
FIGURE 3: EVENT FLOW DIAGRAM	69
FIGURE 4: STATE DIAGRAM FOR CLASS REFLECTAPP	70
FIGURE 5: INPUT AND OUTPUT VALUES FOR REFLECTAPP SYSTEM	71
FIGURE 6: DATA FLOW DIAGRAM (LEVEL 0)	72
FIGURE 7: DATA FLOW DIAGRAM (LEVEL 1) PROCESS 1.0 (DETECTING OBJECT FILES)	73
FIGURE 8: DATA FLOW DIAGRAM (LEVEL 1) PROCESS 2.0 (ANALYSING OBJECT FILES)	74
FIGURE 9: DATA FLOW DIAGRAM (LEVEL 1) PROCESS 3.0 (SEARCHING DETAIL INFORMATION)	75
FIGURE 10: REFLECTAPP CONTROL	76
APPENDIX D: GANTT CHART	77
APPENDIX E: GLOSSARY	79



## **Chapter One**

### **Introduction**

---

This project uses Java technology, a platform-independent<sup>1</sup> application, developed by Sun Microsystems. The Java programming offers “reflection” features, which is capable to examine or interpret Java compiled codes, and turns them into actual uses of classes, method and so on. The main technology applied in this project is the Java’s reflection feature that uses it to reflect the similarities of object concepts.

---

<sup>1</sup> Refer to Appendix E: Glossary, page 79.

The contents of  
the thesis is for  
internal user  
only

## **Bibliography**

Arnold, Ken & Gosling, James. (1997). The Java™ Programming language (2<sup>nd</sup> Edition). England: Addison-Wesley Longman Inc.

Awad, Elias M. (1996). Building Expert System: Principles, Procedures, and Applications. New York: West Publishing Company.

Horstmann, Cay. (1998). Computing Concepts with Java Essentials. New York: John Willy & Sons, Inc.

Jaworski, Jamie. (1996). JAVA Developer's Guide. Indianapolis: Sams.net Publishing.

Jaworski, Jamie. (1998). Java 1.2 Unleashed. Indianapolis: Macmillan Computer Publishing

Kirby, G. N. C. (1992). Persistent Programming with Strongly Typed Linguistic Reflection. <http://www-ppg.dcs.st-and.ac.uk/Publications/1992.html#persistent.programming> (Accessed on 18/08/1998).

Kirby, Graham & Morrison, Ron. (1998). Linguistic Reflection in Java. <http://www-ppg.dcs.st-and.ac.uk/Publications/1998.html#java.reflection> (Accessed on 18/08/1998).

Lemay, Laura & Perkins, Charles L. & Morrison, Michael. (1996). Teach Yourself Java in 21 Days (Professional Reference Edition). Indianapolis: Sams.net Publishing.

Rambaugh, James. & Blaha, Michael. & Premerlani, William. & Eddy, Frederick. & Lorensen, William. (1991). Object-Oriented Modeling and Design. New York: Prentice-Hall International, Inc.

Sjoberg, Dag I.K. & Welland, Ray & Atkinson, Malcolm. (1997). Software Constraints for Large Application Systems. The Computer Journal. Vol.40 No.10 1997. Pg 598-616.

Stemple, D. & Morrison, R. & Kirby, G.N.C. & Connor, R.C.II. (1993). Integrating Reflection, Strong Typing and Static Checking. <http://www-ppg.dcs.st-and.ac.uk/Publications/1993.html#integrating.reflection> (Accessed on 18/08/1998).

Verco, Kristina L. & Wise, Michael J. (1996). Software for Detecting Suspected Plagiarism: Comparing Structure and Attribute-Counting System. Presented paper of First Australian Conference on Computer Science Education, Sydney, Australia, July 3-5 1996, John Rosenberg (Ed), ACM.

Whale, Geoff. (1990). Software Metrics and Plagiarism Detection. Journal of System Software. Vol.13 1990. Pg 131-138.

Whale, G. (2). (1990). Identification of Program Similarity in Large Populations. The computer Journal. Vol.33 No.2 1990. Pg 141-146.