Macmillan Computer Science Series

- A. Abdellatif, J. Le Bihan and M. Limame, *Oracle – A User's Guide*
Ian O. Angell, *High-resolution Computer Graphics Using C*
Ian O. Angell and Gareth Griffith, *High-resolution Computer Graphics Using FORTRAN 77*
Ian O. Angell and Gareth Griffith, *High-resolution Computer Graphics Using Pascal*
C. Bamford and P. Curran, *Data Structures, Files and Databases, second edition*
P. Beynon-Davies, *Database Systems*
P. Beynon-Davies, *Information Systems Development, second edition*
Linda E.M. Brackenbury, *Design of VLSI Systems – A Practical Introduction*
Alan Bradley, *Peripherals for Computer Systems*
P.C. Capon and P.J. Jinks, *Compiler Engineering Using Pascal*
B.S. Chalk, *Computer Organisation and Architecture*
E. Davalo and P. Naïm, *Neural Networks*
S.M. Deen, *Principles and Practice of Database Systems*
Joyce Duncan, Lesley Rackley and Alexandria Walker, *SSADM in Practice*
D. England et al., *A Sun User's Guide, second edition*
Jean Ettinger, *Programming in C++*
J.S. Florentin, *Microprogrammed Systems Design*
Michel Gauthier, *Ada – A Professional Course*
M.G. Hartley, M. Healey and P.G. Depledge, *Mini and Microcomputer Systems*
M.J. King and J.P. Pardoe, *Program Design Using JSP – A Practical Introduction, second edition*
Bernard Leguy, *Ada – A Programmer's Introduction*
M. Léonard, *Database Design Theory*
David Lightfoot, *Formal Specification Using Z*
A.M. Lister and R.D. Eager, *Fundamentals of Operating Systems, fifth edition*
Elizabeth Lynch, *Understanding SQL*
G.P. McKeown and V.J. Rayward-Smith, *Mathematical Foundations for Computing*
Tom Manns and Michael Coleman, *Software Quality Assurance, second edition*
B.A.E. Meekings, T.P. Kudrycki and M.D. Soren, *A Book on C, third edition*
R.J. Mitchell, *C++ Object-Oriented Programming*
R.J. Mitchell, *Microcomputer Systems Using the STE Bus*
R.J. Mitchell, *Modula-2 Applied*
J.P. Pardoe and M.J. King, *Object-Oriented Programming Using C++ – An Introduction*
Pham Thu Quang and C. Chartier-Kastler, *MERISE in Practice*
Ian Pratt, *Artificial Intelligence*
F.D. Rolland, *Programming with VDM*
S. Skidmore, *Introducing Systems Analysis, second edition*

Continued overleaf

Macmillan Computer Science Series (*continued*)

S. Skidmore, *Introducing Systems Design*, second edition
A.G. Sutcliffe, *Human-Computer Interface Design*, second edition
D.J. Tudor and I.J. Tudor, *Systems Analysis and Design – A Comparison of Structured Methods*
Colin J. Theaker and Graham R. Brookes, *Concepts of Operating Systems*
M. Thorin, *Real-time Transaction Processing*
A.J. Tyrell, *Eiffel Object-Oriented Programming*

Non-series

I.O. Angell and D. Tsoubelis, *Advanced Graphics on VGA and XGA Cards using Borland C++*
N. Frude, *A Guide to SPSS/PC+, second edition*
Peter Grossman, *Discrete Mathematics for Computing*
H. Harper and A. Meadows, *GNVQ Advanced Information Technology*
Percy Mett, *Introduction to Computing*
P. Picton, *Neural Networks*
Tony Royce, *COBOL – An Introduction*
Tony Royce, *Structured COBOL – An Introduction*
Tony Royce, *C Programming*

Macmillan Computer Science Series

Series Standing Order

ISBN 978-0-333-71697-7 hardcover

ISBN 978-0-333-69339-1 paperback

(*outside North America only*)

You can receive future titles in this series as they are published by placing a standing order. Please contact your bookseller or, in case of difficulty, write to us at the address below with your name and address, the title of the series and the ISBN quoted above.

Customer Services Department, Macmillan Distribution Ltd
Houndmills, Basingstoke, Hampshire RG21 6XS, England

Program Design Using JSP

A Practical Introduction

M. J. King and J. P. Pardoe

*School of Computing and Mathematical Sciences
Liverpool John Moore's University*

Second Edition

© M. J. King and J. P. Pardoe 1985, 1992

All rights reserved. No reproduction, copy or transmission of this publication may be made without written permission.

No paragraph of this publication may be reproduced, copied or transmitted save with written permission or in accordance with the provisions of the Copyright, Designs and Patents Act 1988, or under the terms of any licence permitting limited copying issued by the Copyright Licensing Agency, 90 Tottenham Court Road, London W1P 9HE.

Any person who does any unauthorised act in relation to this publication may be liable to criminal prosecution and civil claims for damages.

First edition 1985

Second edition 1992

Published by
MACMILLAN PRESS LTD
Houndmills, Basingstoke, Hampshire RG21 6XS
and London
Companies and representatives
throughout the world

ISBN 978-0-333-57674-8

ISBN 978-1-349-22081-6 (eBook)

DOI 10.1007/978-1-349-22081-6

A catalogue record for this book is available
from the British Library.

12 11 10 9 8
03 02 01 00

Contents

<i>Preface</i>	viii
<i>Acknowledgements</i>	x
1 The Importance of Program Design	1
1.1 Introduction	1
1.2 Requirements of a software design method	1
1.3 The characteristics of JSP	2
1.4 The stages of JSP	2
2 Data Structures	4
2.1 The JSP notation	4
2.2 Data and procedures	6
2.3 Examples of how data structures are produced	7
2.4 Exercises	13
3 Data Structures for Particular Problems	16
3.1 Physical data structures and logical data structures	16
3.2 A logical data structure for a simple problem	18
3.3 Exercises	20
4 Program Structures	23
4.1 Identifying correspondences	23
4.2 Combining data structures to produce program structures	24
4.3 Exercises	33
5 Elementary Program Operations and Conditions	36
5.1 Listing the conditions	36
5.2 Listing the elementary program operations	38
5.3 Allocating the conditions and operations	40
5.4 Exercises	44

6	Schematic Logic	45
6.1	Production of schematic logic (pseudo-code)	45
6.2	Schematic logic for a complete program	49
6.3	Exercises	51
7	Implementation in a High Level Programming Language	52
7.1	From schematic logic to target language	52
7.2	Pascal as target language	53
7.3	COBOL as target language	58
7.4	Exercises	64
8	More Than One Input File	65
8.1	Merging data (at most, one record per key per file)	65
8.2	More complex merges (more than one record per key)	72
8.3	The sequential file update	76
8.4	Exercises	81
9	Structure Clashes and Inversion	83
9.1	Types of structure clash	83
9.2	Ordering clash	83
9.3	Boundary clash	84
9.3.1	Why boundary clashes occur	84
9.3.2	A typical example	86
9.4	Solving the boundary clash	87
9.5	Program inversion	94
9.6	Interleaving clash	99
9.7	Exercises	101
10	Recognition Problems and Backtracking	103
10.1	Introduction	103
10.2	The technique of backtracking	105
10.3	The accepted and rejected batch problem	106
10.4	Dealing with intolerable side effects	110
10.5	Implementing quits	112
10.6	Quit from iteration	114
10.7	Exercise	115
11	Procedurisation	116
11.1	Introduction	116
11.2	Bottom-up procedures	116
11.3	Top-down procedures	117

12 Interactive Systems	120
12.1 Interactive dialogue	120
12.2 Menu selection	120
12.3 Form filling	124
12.4 Question and answer	126
12.5 Interrogate or command	131
12.6 Exercises	133
13 Testing, Documentation and Program Amendment	136
13.1 Testing	136
13.1.1 What to test	136
13.1.2 Testing the design	137
13.1.3 Testing the coding	138
13.2 Producing test data using STM	139
13.2.1 What is STM?	139
13.2.2 The STM algebra	140
13.2.3 Deriving conditional expressions	142
13.2.4 Applying the method	145
13.3 Documentation	149
13.4 Program amendment	150
13.5 Exercises	151
<i>Appendix A: Solutions to Exercises</i>	<i>153</i>
<i>Appendix B: An Invoice Printing Case Study</i>	<i>244</i>
<i>Appendix C: A Label Production Case Study</i>	<i>247</i>
<i>Appendix D: A Course File Completeness Case Study</i>	<i>249</i>
<i>Appendix E: An Interactive System Case Study</i>	<i>251</i>
<i>Appendix F: The Invoice Printing Case Study - a Solution</i>	<i>255</i>
<i>Appendix G: The Label Production Case Study - a Solution</i>	<i>261</i>
<i>Appendix H: The Course File Completeness Case Study - a Solution</i>	<i>272</i>
<i>Appendix I: The Interactive System Case Study - a Solution</i>	<i>279</i>
<i>Appendix J: Complete Programs for the Inversion Example (Chapter 9)</i>	<i>286</i>

Preface

Aims of the book

This text aims to provide a practical course for those who want to both understand and apply the Jackson Structured Programming (JSP) approach to program design. It has been designed for trainee and experienced programmers, and students on BTEC courses (such as the Higher National Diploma in Computing) and degree courses that incorporate the study of program design.

The aim is to present the principles and techniques of JSP in a way that will enable the reader to apply the method with confidence.

Structure of the book

The book first presents, in chapters 1-6, the basic method for relatively straightforward problems while stressing the important aspects through numerous examples and exercises. The production of logical data structures, being the basis of the method and the area where students may initially encounter difficulties, is dealt with at length. Chapter 4 then explains how the identification of corresponding components in the input and output structures enables them to be combined to form a program structure. The allocation of conditions and operations to this program structure is covered in chapter 5; conversion of the program structure with allocated conditions and operations into schematic logic (that is, a pseudo-code representation) is then explained in chapter 6.

Since JSP does not presume upon any particular programming language, neither does this book. However, to illustrate the ease with which programs can be coded after they have been designed in a language-free manner, the appropriate coding rules for COBOL and Pascal are outlined in chapter 7.

After the basic techniques have been mastered, the student may then proceed to the more difficult aspects. The concepts of processing more than one input file, including different types of merge problems, are examined in detail. The concepts of structure clashes and recognition problems, together with their solutions (that is, inversion and backtracking respectively) are examined in a practical manner. Interactive programs appear to be a special

class of problem, so a separate chapter deals with the application of the method to such problems.

The book also illustrates the usefulness of the design method in respect of procedurisation, testing, documentation and amendment of existing programs. Special attention is given to a structural testing method.

Use of the book

To ensure that each technique is mastered as it is introduced, we strongly urge the reader to attempt all the exercises given at the end of most chapters, and to arrive at his/her own solutions before turning to the solutions provided in appendix A.

Since this is primarily a book on program design, these solutions are normally given as logical data structures, program structure diagrams with allocated conditions and operations, or as schematic logic. However, having completed the design, the reader may wish to experience the satisfaction of running the program by translating certain solutions into a target language (see chapter 7).

Four case studies and their solutions are also provided in the appendixes. A note is included at the end of chapters 6, 9, 10 and 12, indicating that the appropriate case study may then be attempted. These extended problems can therefore be used for reinforcement and additional practice immediately after the appropriate chapter, or for revision purposes at a later stage.

Acknowledgements

It would be inappropriate to write a book on JSP without acknowledging the work of Michael Jackson, particularly his book *Principles of Program Design*. Our interest in Jackson's method was also stimulated by Brian Ratcliff of the University of Aston in Birmingham.

We would like to thank our colleagues at Liverpool Polytechnic for their advice, assistance and encouragement. Our thanks also to all those students who attended the lecture courses on which this book is based - their questions and comments have helped enormously and contributed to the changes made in this second edition.