3.5.1. Communication protection. 3.5.2. Security layers of the node. 3.5.3. Resource control. 3.5.4. Agent security. 3.6. The home application. 3.6.1. Implementation overview. 3.6.2. The transmission of a request to the
node. 3.6.3. The callback displayer. 3.6.4. Generic views of the agent results. 3.7. CSM internetworking support. 3.7.1. Name and topology information. 3.7.2. Routing. 3.8. Organization of the source code. 4. Applications of
service monitoring agents. 4.1. Monitoring a virtual private network service. 4.1.1. Functionality of a VPN control
agent. 4.1.2. Statistical tests on cryptographic algorithms. 4.2. Service level agreement monitoring. 4.3. Agents for
measuring QoS parameters. 4.3.1. Throughput measurements. 4.3.2. Coordination of distributed measurements. 4.3.3. One-way delay measurements. 4.3.4. The ping measurements. 4.4. Agent security. 4.4.1. Classification of
attacks. 4.4.2. The semantics of the agent. 4.4.3. Attacks on the input of the agent. 4.4.4. Evaluation of the
threat situation. 4.5. Extended application scenarios. 4.5.1. Further applications independent of new node
services. 4.5.2. Future CSM extensions. 5. Performance evaluation. 5.1. Performance evaluation of the node environment.
5.1.1. Throughput of the execution environment. 5.1.2. Node throughput including the TCP receiver. 5.2. Agent
performance. 5.3. Communication performance of the CSM system. 5.4. The T component. 5.5. Discussion and
improvements. 6. Comparison and related work. 6.1. The Internet2 initiative and the QBone. 6.1.1. The QBone.
6.1.2. QBone measurements. 6.1.3. Comparison to our approach. 6.2. Network measurements and monitoring.
6.2.1. IP measurements methodology. 6.2.2. The simple network management architecture. 6.2.3. Measurement
testbeds. 6.3. Mobile agents for management and monitoring. 6.3.1. Network management with mobile agents.
6.3.2. The script MIB. 6.3.3. Network management with active networks. 6.4. Open issues. 6.4.1. Collaboration of
monitoring agents. 6.4.2. Routing. 6.4.3. Artificial intelligence. 7. Summary and conclusion. 8. List of figures. List


Contents:
guide for Part II: Graphs, trees, and recursion. Part I: Linear order. 1. Basic concepts for linear order. 2. Topic I:
Sorting. 3. Topic II: Basic combinatorial lists. 4. Topic III: Symmetry—orbit enumeration and orderly algorithms.
concepts of graphs, trees, and recursion. 7. Topic I: Depth first search and planarity. 8. Topic II: Depth first
search and nonplanarity. 9. Topic III: Triconnectivity. 10. Topic IV: Matroids. References for graphs, trees, and
recurrences. Index.


Contents:
Preface. Notational conventions. 1. Running Java: Compilation, loading, and execution. 2. Names and reserved
names. 3. Java naming conventions. 4. Comments and program layout. 5. Types. 5.1. Primitive types. 5.2. Refer-
ence types. 5.3. Array types. 5.4. Subtypes and compatibility. 5.5. Signatures and subsumption. 6. Variables,
parameters, fields, and scope. 6.1. Values bound to variables, parameters, or fields. 6.2. Variable declarations.
6.3. Scope of variables, parameters, and fields. 7. Strings. 8. Arrays. 8.1. Array creation and access. 8.2. Array
initializers. 8.3. Multidimensional arrays. 8.4. The utility class arrays. 9. Classes. 9.1. Class declarations and class
bodies. 9.2. Top-level classes, nested classes, member classes, and local classes. 9.3. Class modifiers. 9.4. The class
modifiers public, final, abstract. 9.5. Subclasses, superclasses, class hierarchy, inheritance, and overriding.
9.6. Field declarations in classes. 9.7. The member access modifiers private, protected, public. 9.8. Method
declarations. 9.9. Constructor declarations. 9.10. Initializer blocks, field, initializers, and initializers. 9.11. Nested
classes, member classes, local classes, and inner classes. 9.12. Anonymous classes. 10. Classes and objects in the
computer. 10.1. What is a class? 10.2. What is an object? 10.3. Inner objects. 11. Expressions. 11.1. Table of
expression forms. 11.2. Arithmetic operators. 11.3. Logical operators. 11.4. Bitwise operators and shift operators.
11.5. Assignment expressions. 11.6. Conditional expressions. 11.7. Object creation expressions. 11.8. Instance
test expressions. 11.9. Field access expressions. 11.10. The current object reference this. 11.11. Method call
12.2. Block statements. 12.3. The empty statement. 12.4. Choice statements. 12.5. Loop statements. 12.6. Re-
turns, labeled statements, exits, and exceptions. 12.7. The assert statement. 13. Interfaces. 13.1. Interface
declarations. 13.2. Classes implementing interfaces. 14. Exceptions, checked and unchecked. 15. Threads, concurre-
cnt execution, and synchronization. 15.1. Threads and concurrent execution. 15.2. Locks and the synchronized
statement. 16. Compilation, source files, class names, and class files. 17. Packages and jars. 18. Mathematical
functions. 19. String buffers. 20. Collections and maps. 20.1. The collection interface. 20.2. The list interface
and the LinkedList and ArrayList implementations. 20.3. The set interface and the HashSet and LinkedHashSet
implementations. 20.4. The SortedSet interface and the TreeSet implementation. 20.5. The map interface and
the HashMap implementation. 20.6. The SortedMap interface and the TreeMap implementation. 20.7. Going
20.10. Choosing the right collection class or map class. 21. Input and output. 21.1. Creating streams and other
streams. 21.2. Kinds of input and output methods. 21.3. Imports, exceptions, thread safety. 21.4. Sequential