UNIVERSIDADE FEDERAL DO PAMPA

Training Basic Material

Marcelo Schmitt Laser

April 25, 2017

Contents

1 Programming Paradigms 6

1.1 Programming Paradigm vs Programming Language 6

1.2 Programming Languages 6

1.2.1 Basic Elements 6

1.3 Paradigms 6

1.3.1 Imperative 6

1.3.2 Object-Oriented 6

1.3.3 Aspect-Oriented 6

1.3.4 Declarative 6

1.3.5 Functional 6

1.3.6 Logic 6

1.3.7 Event-Driven 6

1.3.8 Reflective 6

2 Programming Logic 7

2.1 Propositional Logic 7

2.2 First-Order Logic 7

2.3 Other Types 7

3 Modeling 8

3.1 Fundaments 9

3.1.1 Model vs Diagram 9

3.1.2 View vs Visualization 9

3.1.3 Structure vs Behaviour 9

3.1.4 System vs Software 9

3.1.5 Analysis vs Architecture 9

3.2 Unified Modelling Language 9

3.2.1 Fundamentals 9

3.2.2 Use-Case Diagram 9

3.2.3 Activity Diagram 9

3.2.4 Class Diagram 9

3.2.5 Component Diagram 9

3.2.6 Sequence Diagram 9

3.2.7 Communications Diagram 9

3.2.8 State Machine Diagram 9
3.2.9 Other UML Diagrams 9
3.3 Other Models and Languages 9
3.3.1 Entity-Relationship 9
3.3.2 Flow Chart 9
3.3.3 User Story 9
3.3.4 Clafer 9
3.3.5 XML 9
3.3.6 xADL 9

4 Data Structures **10**
4.1 Fundamentals 11
4.2 Array 11
4.2.1 Matrix 11
4.3 List 11
4.3.1 Array List 11
4.3.2 Linked List 11
4.4 Stack 11
4.5 Queue 11
4.6 Tree 11
4.6.1 Binary Tree 11
4.6.2 Heap 11
4.7 Graph 11
4.7.1 Directed Graph 11
4.7.2 Weighted Graph 11
4.8 Class 11
4.8.1 Object 11
4.8.2 Struct 11
4.9 Table 11
4.9.1 Hash Table 11
4.9.2 Dictionary 11

5 Algorithms **12**
5.1 Sorting Algorithms 12
5.1.1 Insertion Sort 12
5.1.2 Bubble Sort 12
5.1.3 Quicksort 12
5.1.4 Merge Sort 12
5.2 Search Algorithms 12
5.2.1 Breadth-First Search 12
5.2.2 Depth-First Search 12
5.2.3 A* 12
5.2.4 Dijkstra's Algorithm 12
5.2.5 Floyd-Warshall Algorithm 12

6 Patterns **13**
6.1 Fundamentals . . . 14
6.1.1 Design Pattern vs Architectural Pattern . . . 14
6.1.2 Architectural Pattern vs Architectural Style . . . 14
6.1.3 Architectural Style vs Programming Paradigm . . . 14
6.2 Architectural Styles . . . 14
6.2.1 Pipe-and-Filter . . . 14
6.2.2 Batch-Sequential . . . 14
6.2.3 Implicit Invocation . . . 14
6.2.4 Object-Oriented . . . 14
6.2.5 Distributed Objects (CORBA) . . . 14
6.2.6 Client-Server . . . 14
6.2.7 Mobile Code . . . 14
6.2.8 Peer-to-Peer . . . 14
6.2.9 Event-Based . . . 14
6.2.10 Publish-Subscribe . . . 14
6.2.11 Blackboard . . . 14
6.2.12 C2 . . . 14
6.3 Architectural Patterns . . . 14
6.3.1 Model-View-Controller . . . 14
6.3.2 Broker . . . 14
6.3.3 Layers . . . 14
6.3.4 Sensor-Controller-Actuator . . . 14
6.3.5 Interpreter . . . 14
6.3.6 Application Controller . . . 14
6.4 Design Patterns . . . 14
6.4.1 Factory Method . . . 14
6.4.2 Abstract Factory . . . 14
6.4.3 Adapter . . . 14
6.4.4 Decorator . . . 14
6.4.5 Facade . . . 14
6.4.6 Observer . . . 14
6.4.7 Template Method . . . 14
6.4.8 Builder . . . 14
6.4.9 Prototype . . . 14
6.4.10 Singleton . . . 14
6.4.11 Iterator . . . 14
6.4.12 Mediator . . . 14
6.4.13 Memento . . . 14
6.4.14 State . . . 14
6.4.15 Strategy . . . 14
6.4.16 Visitor . . . 14
6.4.17 Data Transfer Object . . . 14
6.4.18 Plugin . . . 14
6.4.19 Separated Interface . . . 14
6.4.20 Component Configurator . . . 14

6.4.21 Monitor 14
6.5 Connectors 14
6.5.1 First-Class vs Second-Class 14
6.5.2 Types of Connectors 14

7 Formal Languages **15**
7.1 Regular Expressions 15
7.2 Automata 15
7.2.1 Finite vs Infinite 15
7.2.2 Deterministic vs Non-deterministic 15
7.2.3 Probabilistic 15
7.3 Lexical Analysis 15
7.4 Backus-Naur Form 15
7.5 Syntactic Analysis 15
7.6 Semantic Analysis 15

8 Software Testing **16**
8.1 Fundamentals 16
8.1.1 Verification vs Validation 16
8.1.2 Formal Verification vs Testing 16
8.1.3 System Testing vs Software Testing 16
8.2 Functional Testing 16
8.3 Performance Testing 16
8.4 Unit Testing 16
8.5 Integration Testing 16
8.6 Usability Testing 16
8.7 Acceptance Testing 16

Appendices **17**

Chapter 1

Programming Paradigms

1.1 Programming Paradigm vs Programming Language

1.2 Programming Languages

1.2.1 Basic Elements

1.3 Paradigms

1.3.1 Imperative

Structured

Procedural

1.3.2 Object-Oriented

1.3.3 Aspect-Oriented

1.3.4 Declarative

1.3.5 Functional

1.3.6 Logic

1.3.7 Event-Driven

1.3.8 Reflective

Chapter 2

Programming Logic

2.1 Propositional Logic

2.2 First-Order Logic

2.3 Other Types

Chapter 3

Modeling

3.1 Fundaments

3.1.1 Model vs Diagram

3.1.2 View vs Visualization

3.1.3 Structure vs Behaviour

3.1.4 System vs Software

3.1.5 Analysis vs Architecture

3.2 Unified Modelling Language

3.2.1 Fundamentals

3.2.2 Use-Case Diagram

3.2.3 Activity Diagram

3.2.4 Class Diagram

3.2.5 Component Diagram

3.2.6 Sequence Diagram

3.2.7 Communications Diagram

3.2.8 State Machine Diagram

3.2.9 Other UML Diagrams

3.3 Other Models and Languages

3.3.1 Entity-Relationship

3.3.2 Flow Chart

3.3.3 User Story

3.3.4 Clafer

3.3.5 XML

3.3.6 xADL

Chapter 4

Data Structures

4.1 Fundamentals

4.2 Array

4.2.1 Matrix

4.3 List

4.3.1 Array List

4.3.2 Linked List

4.4 Stack

4.5 Queue

4.6 Tree

4.6.1 Binary Tree

4.6.2 Heap

4.7 Graph

4.7.1 Directed Graph

4.7.2 Weighted Graph

4.8 Class

4.8.1 Object

4.8.2 Struct

4.9 Table

4.9.1 Hash Table

4.9.2 Dictionary

Tuple

Chapter 5

Algorithms

5.1 Sorting Algorithms

5.1.1 Insertion Sort

5.1.2 Bubble Sort

5.1.3 Quicksort

5.1.4 Merge Sort

5.2 Search Algorithms

5.2.1 Breadth-First Search

5.2.2 Depth-First Search

5.2.3 A*

5.2.4 Dijkstra's Algorithm

5.2.5 Floyd-Warshall Algorithm

Chapter 6

Patterns

6.1 Fundamentals

6.1.1 Design Pattern vs Architectural Pattern

6.1.2 Architectural Pattern vs Architectural Style

6.1.3 Architectural Style vs Programming Paradigm

6.2 Architectural Styles

6.2.1 Pipe-and-Filter

6.2.2 Batch-Sequential

6.2.3 Implicit Invocation

6.2.4 Object-Oriented

6.2.5 Distributed Objects (CORBA)

6.2.6 Client-Server

6.2.7 Mobile Code

6.2.8 Peer-to-Peer

6.2.9 Event-Based

6.2.10 Publish-Subscribe

6.2.11 Blackboard

6.2.12 C2

6.3 Architectural Patterns

6.3.1 Model-View-Controller

6.3.2 Broker

6.3.3 Layers

6.3.4 Sensor-Controller-Actuator

6.3.5 Interpreter

Chapter 7

Formal Languages

7.1 Regular Expressions

7.2 Automata

7.2.1 Finite vs Infinite

7.2.2 Deterministic vs Non-deterministic

7.2.3 Probabilistic

7.3 Lexical Analysis

7.4 Backus-Naur Form

7.5 Syntactic Analysis

7.6 Semantic Analysis

Chapter 8

Software Testing

8.1 Fundamentals

8.1.1 Verification vs Validation

8.1.2 Formal Verification vs Testing

8.1.3 System Testing vs Software Testing

8.2 Functional Testing

8.3 Performance Testing

8.4 Unit Testing

8.5 Integration Testing

8.6 Usability Testing

8.7 Acceptance Testing

Appendices

Bibliography