

Tecnico di sviluppo software

EVOLUZIONE HARDWARE

2

- **Calcolo automatico** (regolo calcolatore)
- **Macchine elettromeccaniche** (selezionatrici)
- **Calcolatori a valvole e diodi**
- **Mainframe** (IBM 1130, 370, 3090, UNIVAC, ..)
- **Microcomputer** (Olivetti P101)
- **Personal computer** (IBM, Olivetti M100, ..)
- ...
- **Reti neurali**
- **Computer quantistici** (Qbit)

3

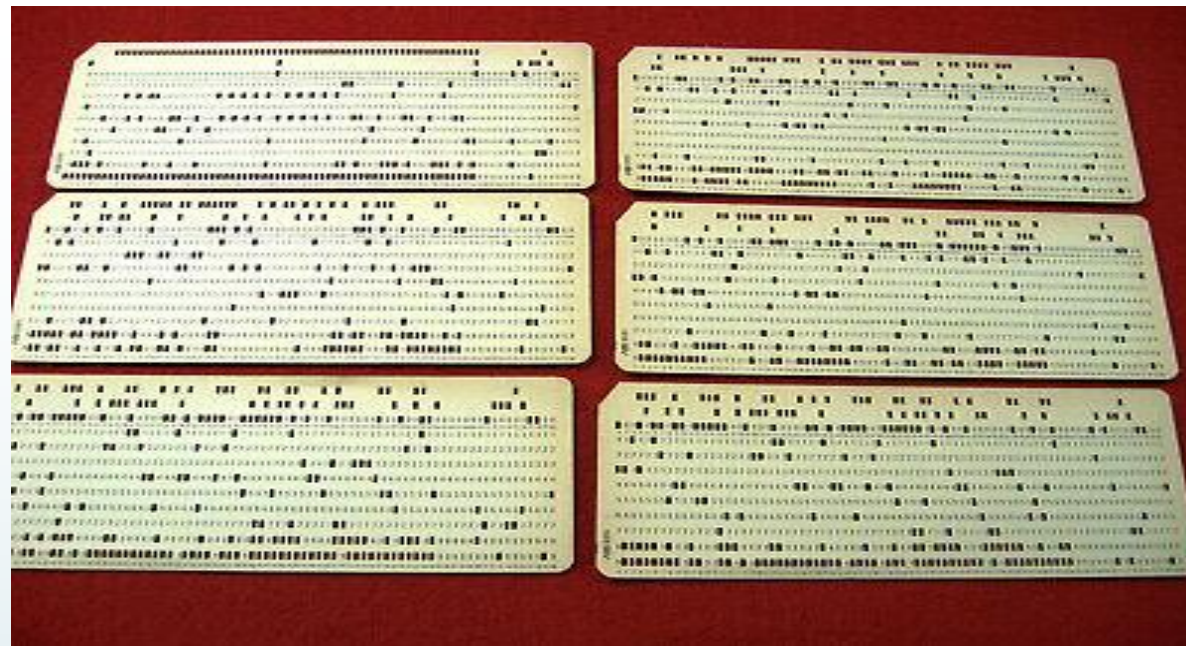


ing. Giampietro Zedda



24/04/2023

4



EVOLUZIONE SOFTWARE

**Linguaggio
macchina**
(0110100...)

Assembler (ST r1,r5;
LA area, *+5; ...)

Scripting (Ibm Rexx,
Unix Shell)

**Linguaggi III
generazione** (C,
Fortran, Algol,
Pascal, Cobol,
Rpg,...)

**Linguaggi IV
generazione
interpretati** (Mantis,
Ufo, Ideal,..)

Linguaggi OO (C++,
Java, Scala, ..)


```
1 public class HelloWorld{  
2  
3     public static void main(String []args){  
4         String mese = "02";  
5         String nome;  
6  
7         switch(mese) {  
8             case "01":  
9                 nome="Gennaio";  
10                break;  
11                case "02":  
12                    nome="Febbraio";  
13                    break;  
14                    case "03":  
15                        nome="Marzo";  
16                        break;  
17                    default:  
18                        nome="Altri mesi";  
19                }  
20                System.out.println("fascia = " + nome);  
21            }  
22        }  
23    }
```

ATTIVITA NELLO SVILUPPO SOFTWARE

- Pianificazione progetto
- Raccolta requisiti di Business
- Analisi dati flussi e processi
- Analisi funzionale
- Analisi tecnica
- Codifica
- Test
- System test
- Collaudo
- Deploy in produzione

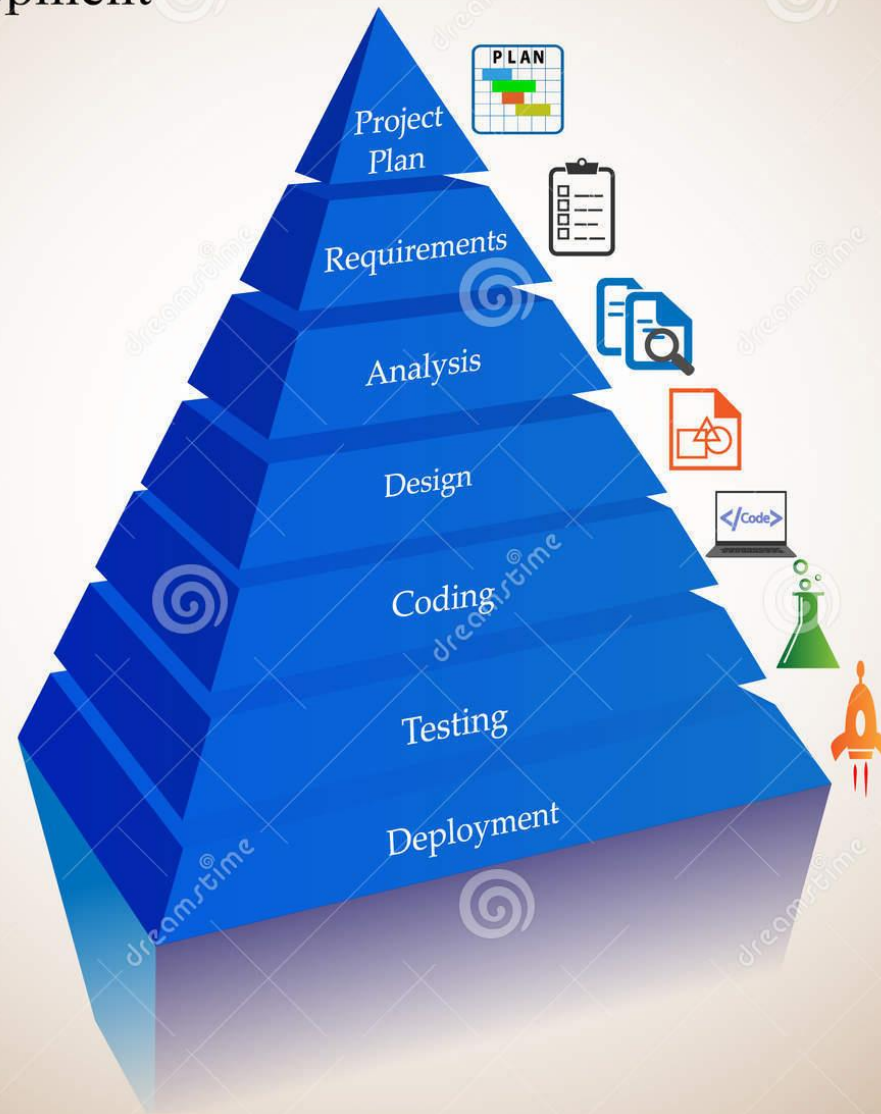
MODELLI SVILUPPO SOFTWARE

- Waterfall (Sviluppo rigido a a cascata)
- Per prototipi
- Agile (Sviluppo iterativo o incrementale poiché lo sviluppo del software si basa sull'apprendimento continuo dalle iterazioni)
 - SCRUM
 - ..
 - CI (continuous Integration, Jenkins, Kubernetes, ..)

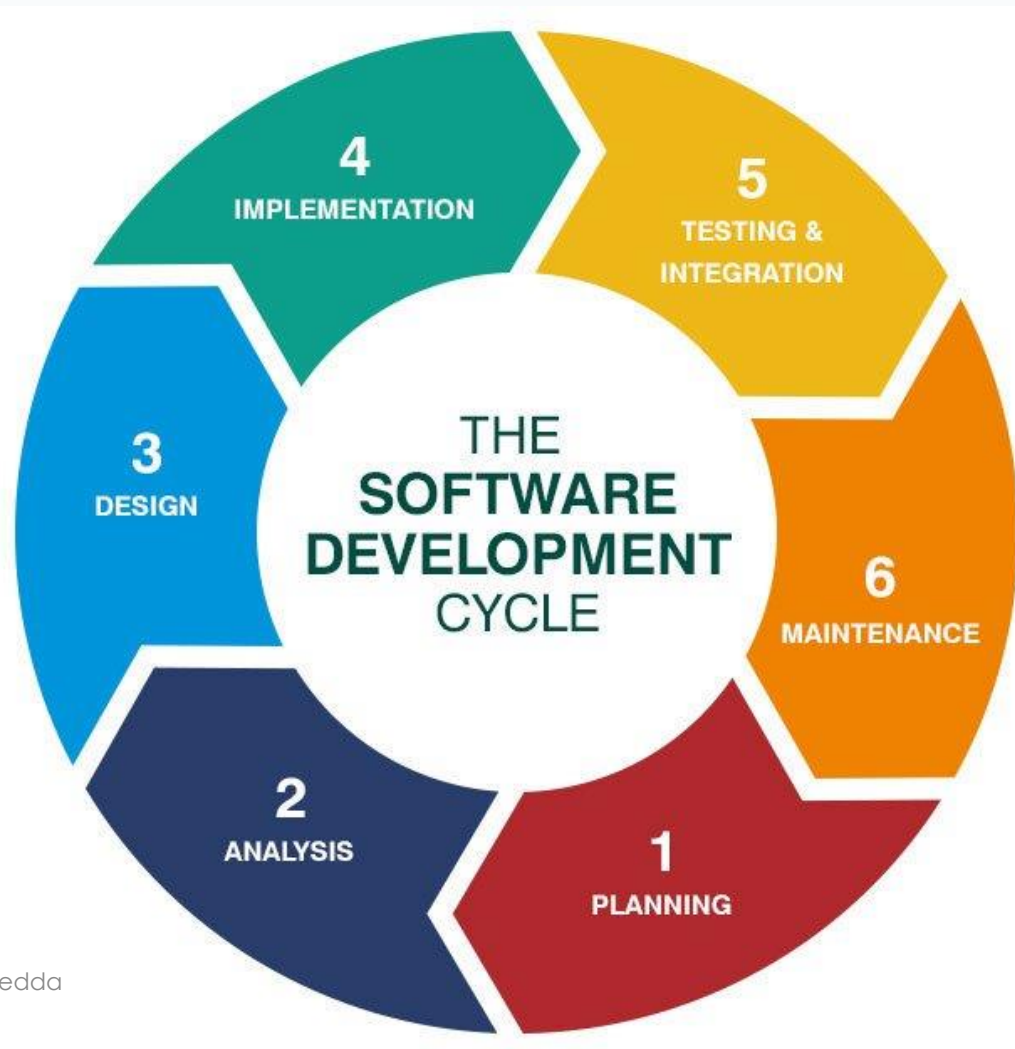
Waterfall-Model



Software Development Life Cycle



AGILE MODEL



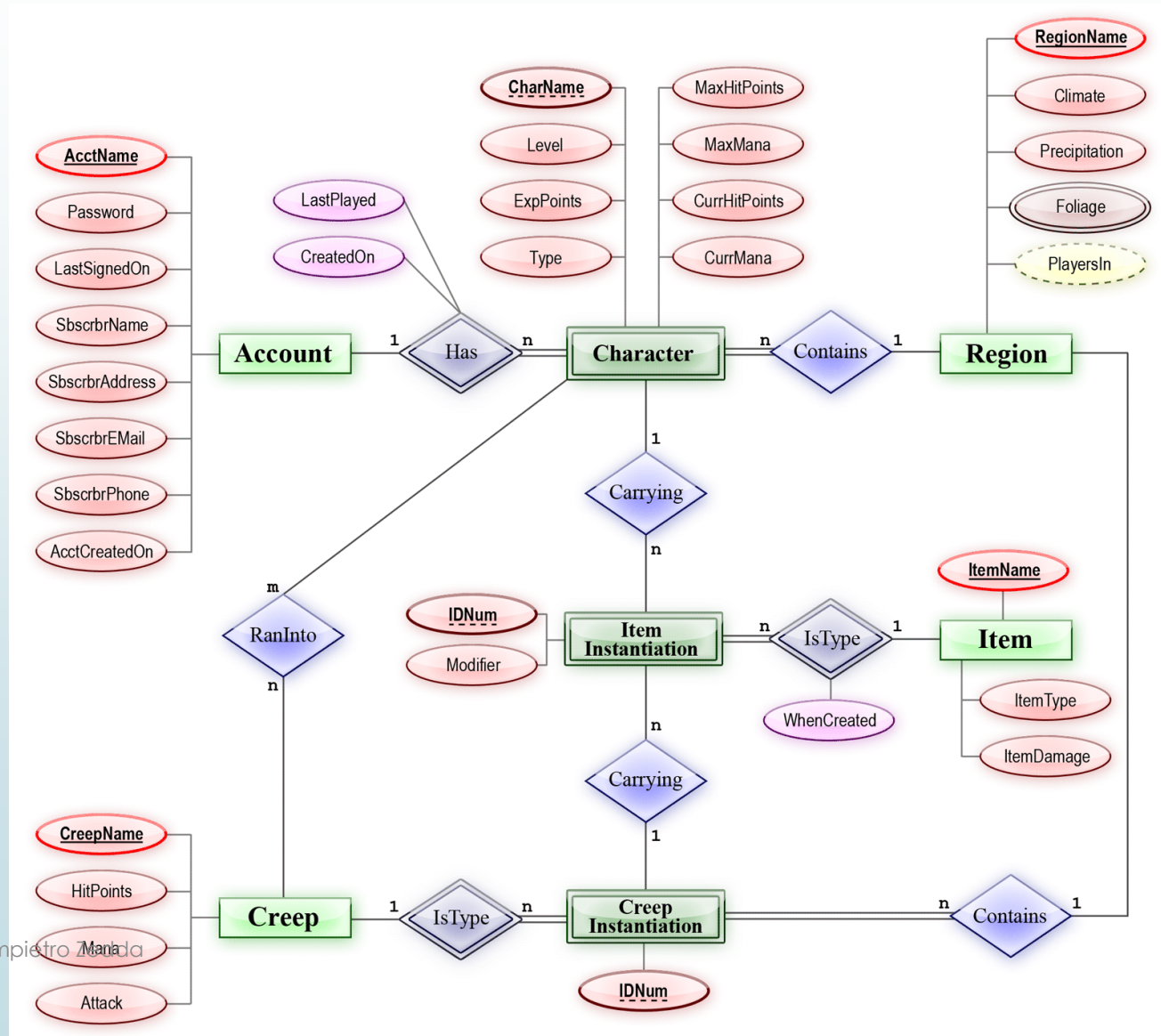
RUOLI TRADIZIONALI

- Business Analyst (business needs, market, competitors)
- Process Analyst (processi, flusso dati, eventi, tempi)
- System Analyst (architettura HW/SW,
- Functional Analyst
- Data Analyst
- Programmer
- System programmer
- Tester
- Data Tester

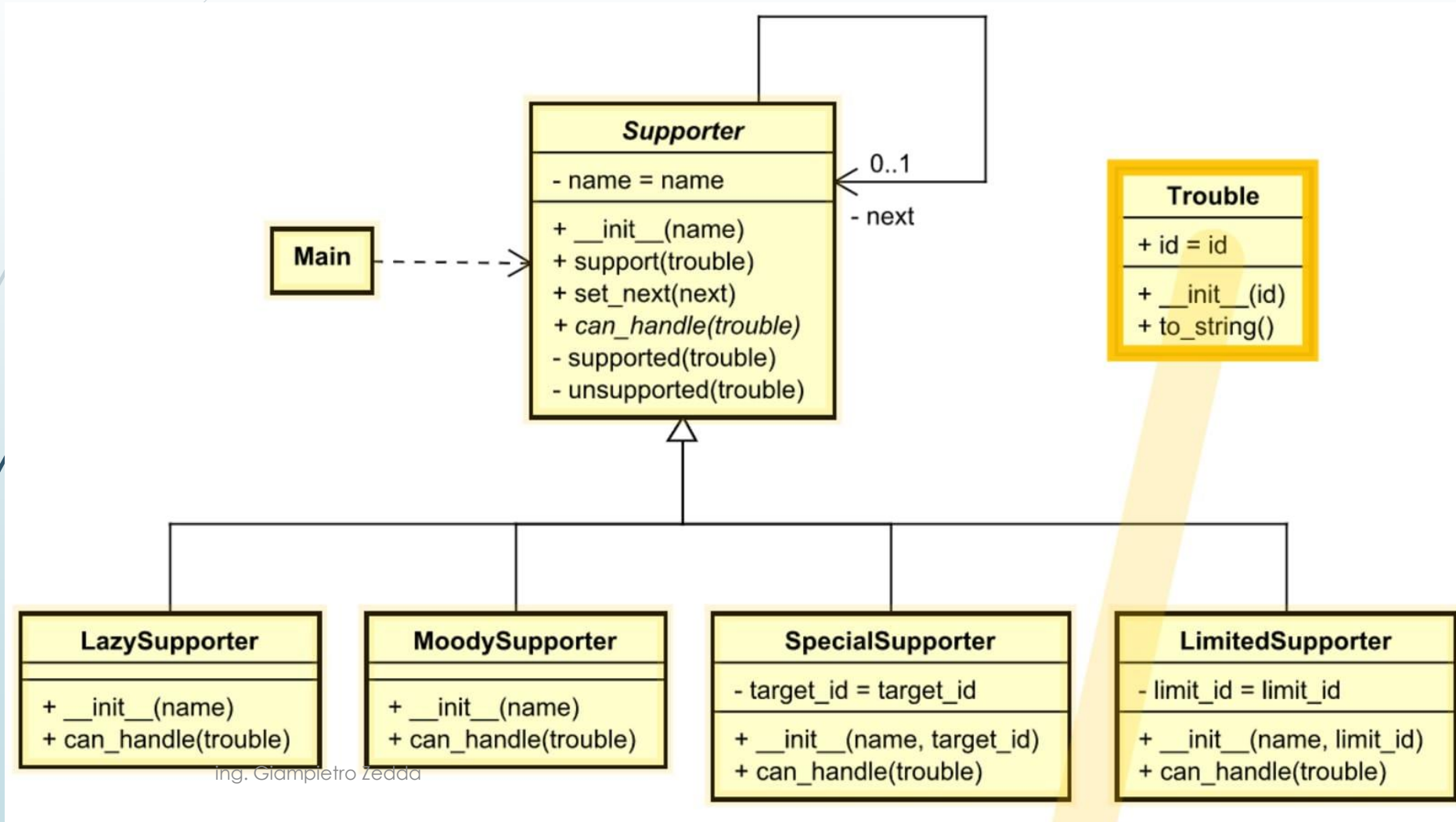
Modelli

- Dal caso specifico al modello generale e viceversa
- Dati (Entity Relationship, UML, ..)
- Funzioni/eventi (Data Flow Diagram, UML, ..)
- Flow Chart
- Diagramma a blocchi

ENTITY RELATIONSHIP

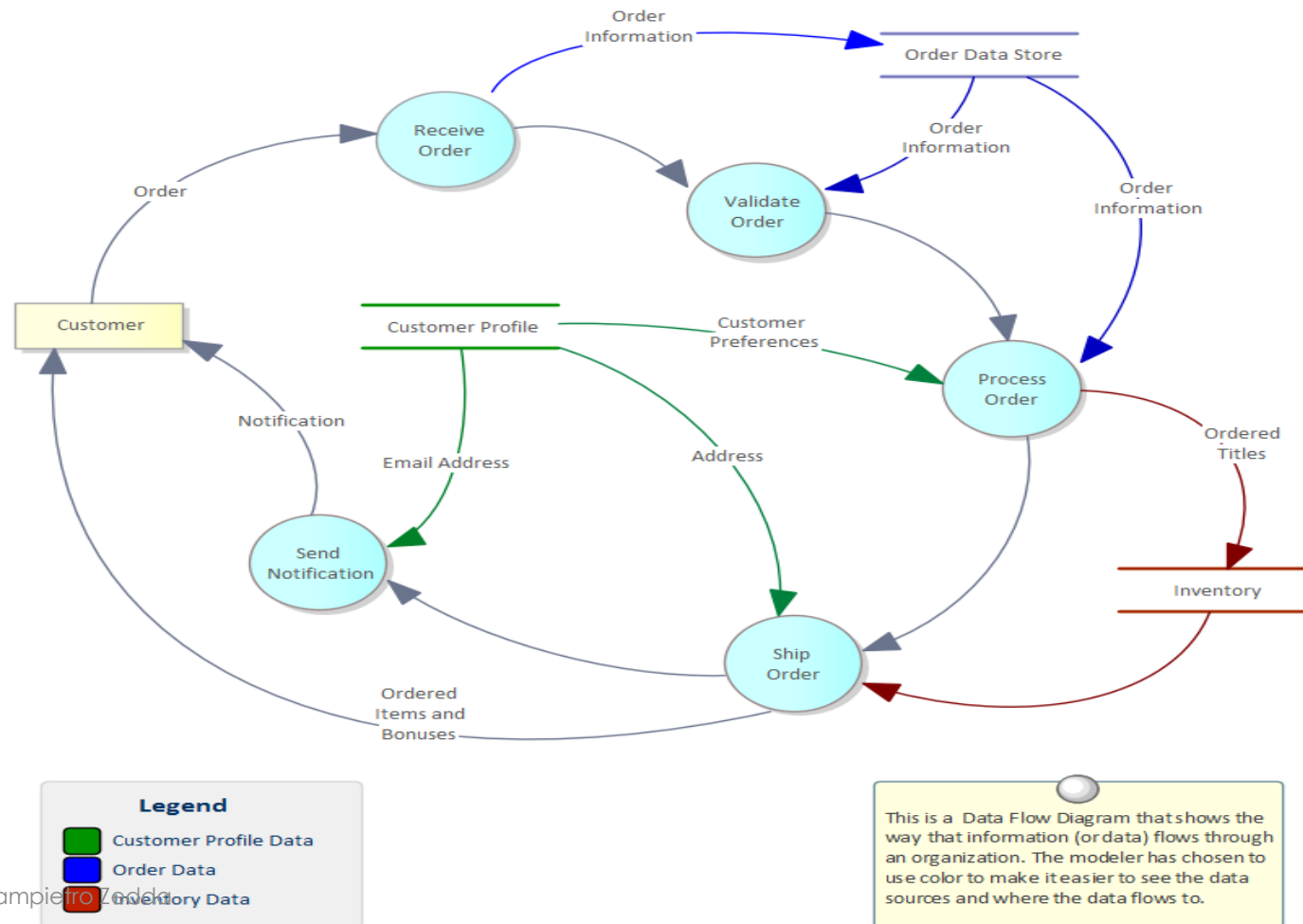


UML DIAGRAM



Data Flow Diagram

dfd Title Order to Delivery Level One



ARTE DEL PROGRAMMARE

- Aspetti artigianali del codice
- Codifica per Livelli omogenei
- Chiarezza e leggibilità (commenti, indentatura)
- Eliminazione istruzioni inutili
- Robustezza runtime
- Qualità intrinseca (Complessità ciclomatica, trasportabilità, ..)
- Qualità strutturale
- Disaccoppiamento
- Sviluppo parallelo di Dati & Funzioni
- Da programmazione strutturale/procedurale a OOP

Introduzione a Java

19

- **1995** Sun Microsystems e Netscape integrano Java nel Browser
- Piattaforma Java
- **OOP** facilitata (**WORA**, architecture independent, JVM, Garbage Collector,.)
- **Java**, C, C++, JavaScript
- Versioni di Java (**JRE, JDK, J2SE, J2EE, J2ME**)
- **1995 (Java1.0)**
- **1998 (Java 1.2)** JIT, Event Handling, J2EE, J2ME
- **2000 (Java 3)** JNDI, JDPA Java Platform Debugging Architecture
- **2002 (Java 4)** Logging, Regular Expression, XML processing,
- **2004 (Java 5)** Generics, Annotation/Metadata, Autoboxing/unboxing
- **2006 (Java 6)** Web Services, Security, Scripting
- **2011 (Java 7)** JVM linguaggi dinamici, libreria file I/O, API per grafica

Introduzione a Java

20

- **2014 (Java 8)** Lambda expression, programmazione funzionale, Stream, Crittografia
- **2017 (Java 9)** Static e methods in Interface, Stream enhancements, immutable collection, Jshell, modules
- **2018 (Java 10)**
- ...
- **2022 (Java 18)** Var declare, Stream Collectors,

JDK Java Developer Kit

- **Javac** È il **compilatore**, il cui compito è quello di trasformare il codice sorgente Java nel bytecode che sarà poi eseguito dalla macchina virtuale Java (JVM)
- **JRE** (Java Runtime Environment) È il compilatore, il cui compito è quello di trasformare il codice sorgente Java nel bytec
- **JAR (Java archiver)** Java Archiver per realizzare archivi di classi
- **Javadoc** Utility che serve per generare la documentazione (in HTML) di codice java a partire da commenti inseriti nei sorgenti stessi.
- **Javap** Java Class Disassembler
- **Javah** Esecuzione codice C nativo da Java
- **AppletViewer** Esecutore Applet fuori da un Web Browser
- **Jdb** Java Debugger

IDE (Integrated Development Environment)

- NetBeans
- Eclipse
- IntelliJ Idea
- Visual Studio Code
- Plugins
- Automazione processo di Build
- Refactoring
- Intellisense
- Autocomplete