

**INGEGNERIA DEL SOFTWARE 2016-2017**

**Documento di Object Design 0.1**

**Fiorazon**

**14/12/2016**

**Sommario**

[**1. Introduction 3**](#_Toc469503994)

[**I. Object design trade-offs 3**](#_Toc469503995)

[**II. Interface documentation guidelines 3**](#_Toc469503996)

[**III. Definitions, acronyms, and abbreviations 3**](#_Toc469503997)

[**IV. References 3**](#_Toc469503998)

[**V. Packages 3**](#_Toc469503999)

[**2. Class interfaces 4**](#_Toc469504000)

[**3. Glossary 4**](#_Toc469504001)

[**4. Team 4**](#_Toc469504002)

[**5. Storia delle versioni 4**](#_Toc469504003)

### **Introduction**

The first section of the ODD is an introduction to the document. It describes the general trade-offs made by developers (e.g., buy vs. build, memory space vs. response time), guidelines and conventions (e.g., naming conventions, boundary cases, exception handling mechanisms), and an overview of the document.

Interface documentation guidelines and coding conventions are the single most important factor that can improve communication between developers during object design. These include a list of rules that developers should use when designing and naming interfaces. These are examples of such conventions:

Classes are named with singular nouns.

Methods are named with verb phrases, fields, and parameters with noun phrases.

Error status is returned via an exception, not a return value.

Collections and containers have an elements() method returning an Enumeration.

Enumerations returned by elements() methods are robust to element removals.

Such conventions help developers design interfaces consistently, even if many developers contribute to the interface specification. Moreover, making these conventions explicit before object design makes it easier for developers to follow them. In general, these conventions should not evolve during the project.

### **Object design trade-offs**

### **Interface documentation guidelines**

### **Definitions, acronyms, and abbreviations**

### **References**

### **Packages**

### **Class interfaces**

The second section of the ODD, Packages, describes the decomposition of subsystems into packages and the file organization of the code. This includes an overview of each package, its dependencies with other packages, and its expected usage.

### **Glossary**

The third section, Class interfaces, describes the classes and their public interfaces. This includes an overview of each class, its dependencies with other classes and packages, its public attributes, operations, and the exceptions they can raise.

### **Team**

|  |  |
| --- | --- |
| **Nome** | **Matricola** |
| Giuseppe Siani | 0512102958 |
| Carmine D’ Alessandro | 0512103402 |
| Aleandro Giuseppe Libano | 0512102976 |

### **Storia delle versioni**

|  |  |  |  |
| --- | --- | --- | --- |
| DATA | VERSIONE | DESCRIZIONE | AUTORE |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |