SOLID (object-oriented design)

From Wikipedia, the free encyclopedia

In computer programming, **SOLID** (**single responsibility, open-closed, Liskov substitution, interface segregation and dependency inversion**) is a mnemonic acronym introduced by Michael Feathers for the "first five principles" named by Robert C. Martin[1][2] in the early 2000s[3] that stands for five basic principles of object-oriented programming and design. The intention is that these principles, when applied together, will make it more likely that a programmer will create a system that is easy to maintain and extend over time.[3] The principles of SOLID are guidelines that can be applied while working on software to remove code smells by providing a framework through which the programmer may refactor the software's source code until it is both legible and extensible. It is part of an overall strategy of agile and Adaptive Software Development.[3]

Contents

Overview

Initial	Stands for	Concept
S	SRP[4]	**Single responsibility principle** a class should have only a single responsibility (i.e. only one potential change in the software's specification should be able to affect the specification of the class)
O	OCP[5]	**Open/closed principle** “software entities … should be open for extension, but closed for modification.”
L	LSP[6]	**Liskov substitution principle** “objects in a program should be replaceable with instances of their subtypes without altering the correctness of that program.” See also design by contract.
I	ISP[7]	**Interface segregation principle** “many client-specific interfaces are better than one general-purpose interface.”[8]
D	DIP[9]	**Dependency inversion principle** one should “depend upon abstractions, [not] concretions.”[8]

See also

Basic concepts and related topics

- Adaptive Software Development
- Agile programming
- Code reuse
- Computer programming
- Object-oriented programming
  - Inheritance (object-oriented programming)

Design and development principles

- Package principles
- DRY
- GRASP (object-oriented design)
- KISS
- YAGNI

References

1. “Principles Of OOD” (http://butunclebob.com/ArticleS.UncleBob.PrinciplesOfOod), Robert C. Martin (“Uncle BOB”), butunclebob.com, Last verified 2014-07-17. (Note the reference to “the first five principles”, though the acronym is not used in this article.) Dates back to at least 2003.
2. “Getting a SOLID start.” (https://sites.google.com/site/unclebobconsultingllc/getting-a-solid-start), Robert C. Martin (“Uncle Bob”), objectmentor.com. Last verified 2013-08-19.
3. “SOLID Object-Oriented Design” (http://www.confreaks.com/videos/240-goruco2009-solid-object-oriented-design), Sandi Metz (Duke University), Talk given at the 2009 Gotham Ruby Conference in May, 2009. Last verified 2009-01-15.
4. "Single Responsibility Principle" (PDF). Archived from the original (PDF) on 1 June 2015.
5. "Open/Closed Principle" (PDF). Archived from the original (PDF) on 5 September 2015.
6. "Liskov Substitution Principle" (PDF). Archived from the original (PDF) on 5 September 2015.
7. "Interface Segregation Principle" (PDF). 1996. Archived from the original (PDF) on 5 September 2015.
8. Robert C. Martin (“Uncle Bob”) (2000), *Design Principles and Design Patterns* (PDF), objectmentor.com, archived from the original (PDF) on 6 September 2015, retrieved 2009-01-14
9. "Dependency Inversion Principle" (PDF). Archived from the original (PDF) on 5 September 2015.