

1. Record Nr.	UNINA9910300747103321
Autore	Noback Matthias
Titolo	Principles of Package Design : Creating Reusable Software Components // by Matthias Noback
Pubbl/distr/stampa	Berkeley, CA : , : Apress : , : Imprint : Apress, , 2018
ISBN	9781484241196 1484241193
Edizione	[1st ed. 2018.]
Descrizione fisica	1 online resource (287 pages)
Disciplina	005.117
Soggetti	Computer programming Programming languages (Electronic computers) Software engineering Open source software Programming Techniques Programming Languages, Compilers, Interpreters Software Engineering Web Development Open Source
Lingua di pubblicazione	Inglese
Formato	Materiale a stampa
Livello bibliografico	Monografia
Nota di bibliografia	Includes bibliographical references.
Nota di contenuto	Part 1: Class Design -- Chapter 1: The Single Responsibility Principle -- Chapter 2: The Open/Closed Principle -- Chapter 3: The Liskov Substitution Principle -- Chapter 4: The Interface Segregation Principle -- Chapter 5: The Dependency Inversion Principle -- Part 2: Package Design -- Chapter 6: The Release/Reuse Equivalence Principle -- Chapter 7: The Common Reuse Principle -- Chapter 8: The Common Closure Principle -- Chapter 9: The Acyclic Dependencies Principle -- Chapter 10: The Stable Dependencies Principle -- Chapter 11: The Stable Abstractions Principle -- Chapter 12: Conclusion -- Appendix A: The Full Page Class.-
Sommario/riassunto	Apply design principles to your classes, preparing them for reuse. You will use package design principles to create packages that are just right in terms of cohesion and coupling, and are user- and maintainer-

friendly at the same time. The first part of this book walks you through the five SOLID principles that will help you improve the design of your classes. The second part introduces you to the best practices of package design, and covers both package cohesion principles and package coupling principles. Cohesion principles show you which classes should be put together in a package, when to split packages, and if a combination of classes may be considered a "package" in the first place. Package coupling principles help you choose the right dependencies and prevent wrong directions in the dependency graph of your packages. What You'll Learn: Apply the SOLID principles of class design Determine if classes belong in the same package Know whether it is safe for packages to depend on each other.
