

1. Record Nr.	UNINA9910523783903321
Autore	Nesteruk Dmitri
Titolo	Design patterns in modern C++20 : reusable approaches for object-oriented software design // Dmitri Nesteruk
Pubbl/distr/stampa	[California] : , : Apress, , [2022] ©2022
ISBN	1-4842-7295-1
Edizione	[Second edition.]
Descrizione fisica	1 online resource (391 pages)
Disciplina	005.133
Soggetti	C++ (Computer program language) Software patterns Computer software - Reusability
Lingua di pubblicazione	Inglese
Formato	Materiale a stampa
Livello bibliografico	Monografia
Note generali	Includes index.
Nota di contenuto	1. Introduction -- Pt I Creational Patterns -- 2. Builder -- 3. Factories -- 4. Prototype -- 5. Singleton -- Pt II Structural Patterns -- 6. Adapter -- 7. Bridge -- 8. Composite -- 9. Decorator -- 10. Facade -- 11. Flyweight -- 12. Proxy -- Pt III Behavioral Patterns -- 13. Chain of Responsibility -- 14. Command -- 15. Interpreter -- 16. Iterator -- 17. Mediator -- 18. Memento -- 19. Null Object -- 20. Observer -- 21. State -- 22. Strategy -- 23. Template Method -- 24. Visitor.
Sommario/riassunto	Apply the latest editions of the C++ standard to the implementation of design patterns. As well as covering traditional design patterns, this book fleshes out new design patterns and approaches that will be useful to modern C++ developers. Author Dmitri Nesteruk presents concepts as a fun investigation of how problems can be solved in different ways, along the way using varying degrees of technical sophistication and explaining different sorts of trade-offs. Design Patterns in Modern C++20, Second Edition also provides a technology demo for modern C++, showcasing how some of its latest features (e. g., coroutines, modules and more) make difficult problems a lot easier to solve. The examples in this book are all suitable for putting into production, with only a few simplifications made in order to aid readability. What You Will Learn Use creational patterns such as builder,

factories, prototype and singleton Implement structural patterns such as adapter, bridge, decorator, facade and more Work with the behavioral patterns such as chain of responsibility, command, iterator, mediator and more Apply functional design patterns such as the Maybe Monad Who This Book Is For This book is for both beginner and experienced C++ developers.
