Notes part 13

Clean C++20. Ch. 6. Modularization

• information hiding (p. 222)
• strong cohesion (p. 222)
• loose coupling (p. 222)
• **SRP** – Single Responsibility Principle – similar to Separation of Concerns in database design (pp. 225-226)
• **SLA** – Single Level of Abstraction (p. 226)
• **OCP** – Open-Closed Principle (pp. 232-233)
• type erasure (pp. 233-240)
– duck-typing (p. 237)
• **LSP** – Liskov Substitution Principle (p. 240)
• `final` specifier (p. 243)
• **RTTI** – run-time type information/identification (p. 250-253)
• **ISP** – Interface Segregation Principle (pp. 253-258)
• **DIP** – Dependency Inversion Principle (pp. 262-268)
• *Law of Demeter* – don't talk to strangers (pp. 269-275)
– aspect-oriented software development
– separate interface from hidden implementatiton
• avoid anemic classes (pp. 275-276)
• Tell, Don't'Ask (pp. 276-279)
• avoid static class members (pp. 279-280)
• avoid `include` if possible (pp. 281-282)
• **ODR** violation (p. 283)
• **BMI** – Built Module Interface file (pp. 284286)

Modern C++. Ch. 12. Modules

• Nice for reference

Advanced C++. Ch. 12. Modules

• Nice for reference

Exploring C++20. Ch. 43. Old-Fashioned "Modules"

• Nice for reference