Design Patterns

Örnek: “Singleton Pattern”

DESIGN FOR CHANGE ----> Ana mottomuz.

Kitaplar:

* C++ Software Design – Klaus Iglberg (2022)
* Design Patterns: Elements of Reusable Object Oriented Software – Gamme Erich, Helm Richard, Johnson Ralph, Vlıssides John (1994)
  + GoF (Gang of Four)
  + Inheritance – Polymorphism
  + 20’den fazla tasarım kalıbı içerir
  + 2011’den sonra CONCURRENCY ve TEMPLATES+LAMBDA C++’a eklendi ve 2022 kitabı ortaya çıktı.

Timeline:

* 16 Ekim
  + Visitor Design Pattern (DP)
  + Chapter 4
* 23 Ekim
  + Strategy DP
  + Command DP
  + Chapter 5
* 30 Ekim
  + Adapter DP
  + Observer DP
  + CRTP DP
  + Chapter 6
* 6 Kasım
  + Bridge DP
  + Prototype DP
  + Chapter 7
* 13 Kasım
  + Decorater DP
  + Singleton DP
  + Chapter 9, 10
* 20 Kasım
  + Type Ensure DP
  + Chapter 8

Sınıf oluştururken sınıfın manager isminde olmasın, küçük parçalara böl.

SOLID DESIGN PRINCIPLES

* Single Responsibility
  + Sınıfın tek bir işi olsun, bir şey değişecekse sadece orası değişsin.
* Open/Closed
  + Classes should be
    - open to extension
    - close to modification
* Liskou Substitution Principle
  + Tasarım yaptım, base class var.
  + Base classtan türetilen class’ı base class yerine koyabiliyorsam tasarım iyidir.
  + Subtypes should be replaceable by their base types.
* Interface Segregation Principle
  + Base’in değiştirilme ihtiyacının engellenmesiyle ilgili.
  + Alakasız interfaceleri ayırmalısın.
  + Many specific interfaces are better than a general interface.
* Dependency Inversion
  + Sınıflar soyutlamalarla konuşmalı.
  + We must depend on abstractions and not concrete classes.
  + High-level modules should not depend on low-level modules, both should depend on abstractions.

Factory Method: loosens the coupling of a given code by separating the product’s construction code from the code that uses this product.

* Use it if you have no idea of the exact types of the objects

A diagram of a factory method

Description automatically generated