**Design Pattern:** A pattern is a recurring solution to a standard problem.

**GRASP:** General Responsibility Assignment Software Patterns

**Five Grasp Patterns:**

1. **Creator**
2. **Information Expert**
3. **Low Coupling**
4. **Controller**
5. **High Cohesion**

**Creator Pattern**

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

**Information Expert Pattern**

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

**Low Coupling**

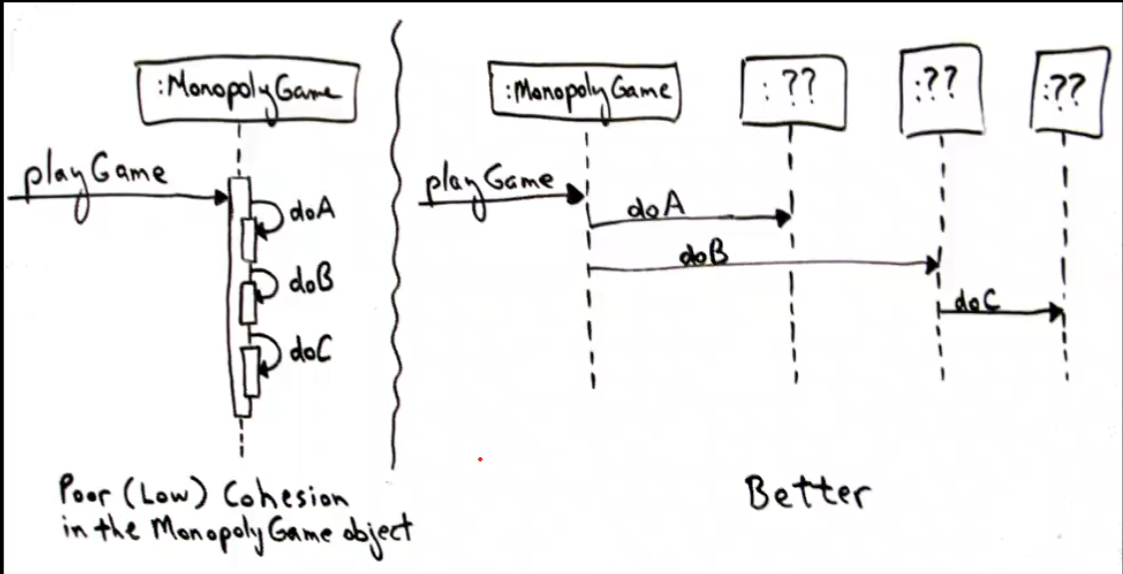
metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

**High Cohesion**

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu



**Controller Pattern**

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

**More Grasp Patterns:**

1. **Polymorphism**
2. **Indirection**
3. **Pure Fabrication**
4. **Protected Variations**

**Testing**

* Testing begins by evaluating the OOA and OOD models
* **Extreme Programming (XP)** advocates writing tests for units before writing actual code for methods.
* **Testing OO code:**
  + **Class tests (Unit):** Test each method within a class.
  + **System tests:** Finger pointing defence. Types of system testing:
    - **Recovery Testing:** Recover from faults.
    - **Security Testing:** Protection.
    - **Stress Testing:** Abnormal load.
    - **Performance Testing:** Run-time performance.
  + **Integration tests**
    - **Thread-based testing:** Integrates classes required to respont to one input or event.
    - **Use-based testing:** Integreates classes required by one use case
    - **Cluster testing:** Integrates classes required to demonstrate one collaboration
  + **Validation tests:** User-visible actions and user-recognizable outputs. Details of class connections disappear at this level. Alpha: Developer level. Beta: Customer level.