Explaining GRASP

Matteo Concas

Telecommunication Software

Week 9 Task

G.R.A.S.P for General Responsibility Assignment Software Patterns

- Used to assign responsibilities to collaborating objects.
- 9 GRASP patterns :
  - Creator
  - Information Expert
  - Low Coupling
  - Controller
  - High Cohesion
  - Indirection
  - Polymorphism
  - Protected Variations
  - Pure Fabrication

Responsability

- Responsability can be assigned to one or multiple objects
- GRASP only helps with assigning responsabilities not how objects should be developped / designed / implemented.

Pattern 1 : The Creator

- A minimum number of objects should be able to create new objects / instances.
- Choice made by looking at the objects' association and their interactions.

Pattern 2 : The Information Expert

- Expert principle : assign responsibility to an object if the object has the information to fulfill the responsibility.
- They might directly have all the information needed or they can get help from another object.

Pattern 3 : Low Coupling

- Evaluates how strongly the objects are coupled (read dependent) to each others.
- Low coupling -> reduce impacts of change and increase stability to the system in case an object attributes change.
- Low coupling makes the code more compartmentalized which leads to less technical debt.
- Two objects are coupled if :
  - One is a parent of another.
  - They share information to work.

Pattern 4 : The Controller

- Bridges the gap from GUI to domain layer objects.
- Delegates UI request to the other objects.
- Multiple controllers can exist simultaneously, usually to avoid having a class overloaded with too many responsibilities (a bloated controller).

Pattern 5 : High Cohesion

- Evaluates how the operations of the different objects are related.
- Define the purpose of the element.
- High Cohesion works in hand with Low Coupling and gives the same benefits.

Pattern 6 : Polymorphism

- Follows from High Cohesion : how to handle related but varying elements ?
- Polymorphism helps us decide which objects is responsible for these elements
- Implementation makes handling new variations easier.

Pattern 7 : Pure Fabrication

- Artificial class that is assigned responsibilities that can’t be assigned to any other objects.
- Benefits : high cohesion and low coupling since the class is “custom” made for the responsibilities needed.

Pattern 8 : Indirection

- Used to avoid direct coupling between objects.
- Create a “middleman” to interact with the objects thus not coupling them directly.
- Benefits : low coupling.

Pattern 9 : Protected Variation

- Used to avoid impacts of variations of one object on the other objects.
- Gives a defined interface between objects.
- Benefits : protection from variations.