

## Software Design Pyramide

- Architektur Patterns
- Design Patterns
- Prinzipien des OOD
- Grundlegende Konzepte des OOD

## Grundlegende Konzepte des OOD

### Abstraktion

- Ignorieren irrelevanter Details

### Kapselung

- Geheimnisprinzip (information hiding)
- Trennung von Implementation und Schnittstelle

### Vererbung

- Erweiterung und Spezialisierung

### Polymorphismus

- Austauschbarkeit
- Gleiche Schnittstelle, anderes Verhalten

### Kohäsion

- Maß für die Zusammengehörigkeit der Bestandteile einer Komponente
- Hohe Kohäsion einer Komponente erleichtert Verständnis, Wartung und Anpassung

### Kopplung

- Maß für die Abhängigkeiten zwischen Komponenten
- Geringe Kopplung erleichtert die Wartbarkeit und macht das System stabiler

### Allgemeines

- Immer gegen Interfaces anstatt Classes programmieren (zb. list -> arraylist)
- Wenn die equals method implementiert wird dann auch hashCode implementieren

## Test

**Whitebox Tests (WBT)** Testet eher Struktur als Funktion, Entwicklertest

### Vorteile

- Testen von Teilkomponenten und der internen Funktionsweise
- Geringerer organisatorischer Aufwand
- Automatisierung durch gute Tool-Unterstützung

### Nachteile

- Erfüllung der Spezifikation nicht überprüfbar
- Eventuelles Testen um Fehler herum

**Black-Box-Test (BBT)** Entwicklung ohne Kenntnis des Codes, Orientierung an Spezifikation bzw. Anforderungsdefinition, funktionsorientiert, kein Entwicklertest

### Vorteile

- bessere Verifikation des Gesamtsystems
- Testen von semantischen Eigenschaften bei geeigneter Spezifikation
- Portabilität von systematisch erstellten Testsequenzen auf plattformunabhängige Implementierungen

### Nachteile

- größerer organisatorischer Aufwand
- zusätzlich eingefügte Funktionen bei der Implementierung werden nur durch Zufall getestet
- Testsequenzen einer unzureichenden Spezifikation sind Unbrauchbar

### Grey-Box-Test (GBT)

- Vereint die Vorteile aus WBT und BBT
- Unterstützt eine testgetriebene Entwicklung
- Ohne Kenntnis der Implementierung entwickelt

**Fazit Unittests** Anzahl der Testabdeckung sagt nur aus, dass die Codezeilen durchlaufen wurden, nicht wie gut die Tests sind!

## Design Patterns GOF

### Gang of Four Gruppen

#### Erzeugungsmuster

- Singleton
- Builder
- Factory Method
- Abstract Factory
- Prototype

#### Strukturmuster

- Facade
- Decorator
- Adapter
- Composite
- Bridge
- Flyweight
- Proxy
- Abstract Factory
- Prototype

#### Verhaltensmuster

- State
- Template Method
- Strategy
- Observer
- Chain of Responsibility
- Command
- Interpreter
- Iterator
- ...

**Packaging Designprinzipien**    Zyklen von Paketen durch ein Interface eines Pakets auflösen, danach Implementierung des Interfaces in ein Paket einbinden.

## SOLID

### SOLID

S	O	L
SRP	OCP	LSP
Single Responsibility Principle	Open Closed Principle	Liskov Substitution Principle