**What is the goal of using the design structural patterns?**

Dr Kruczkiewicz definition: Grouping classes and objects into larger structures

Normal definition: Structural design patterns ease the design by identifying a simple way to realize relationships between entities.

## Builder - change only the way of creation of the complex objects

Abstract Factory - change the families of created objects

Factory Method - change only the subclass of created objects

Prototype - change only the class of created objects

Singleton - need only the one instance of created object

Adapter(wrapper) - if you change the interfaces of the used objects

Decorator - change only the duties of the used objects without necessity definition of new subclasses, or changes of their interfaces

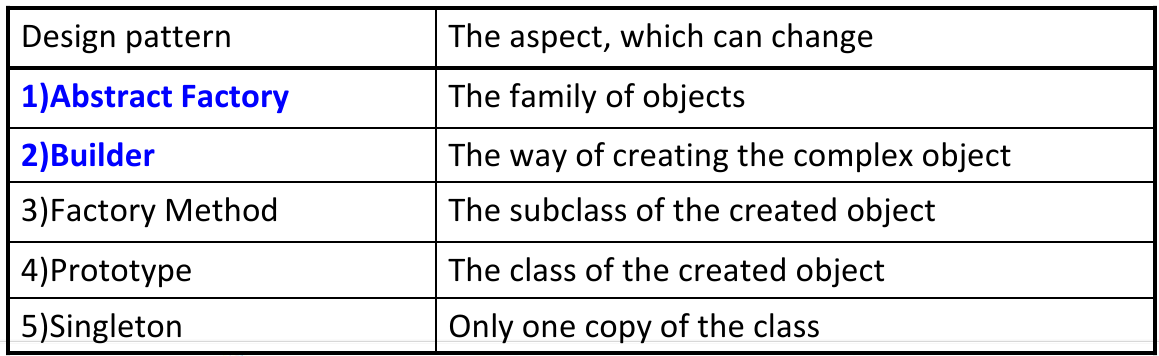
Facade - use if you need encapsulation of the complex logic of the used objects, which don't know this pattern

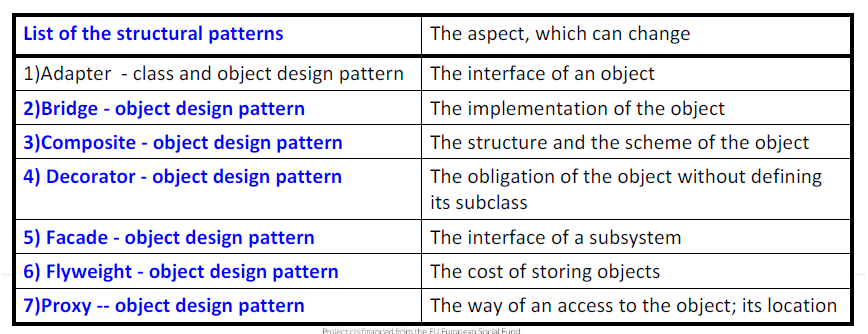
Composite - use if you change the structure of the used objects

Bridge - if you change implementation of some used objects separately from changes the duties of the other objects which use changed objects

Proxy - if you need to control the access to the used objects

Flyweight - if you need the access to the same instance of an object from other objects, included in a complex structure of objects





**3.1. What is the goal of using the design behavioral patterns?**

behavioral design patterns

identify common communication patterns between objects and realize these patterns. By doing so, these patterns increase flexibility in carrying out this communication.

Interpreter -if you change grammar and interpretation of language

Chain of Responsibility - if you select the object for performing a request

Mediator - if you change the used objects and the way of an affect on each other

Template method - if you define the new subclass to handle changes in the steps of an algorithm

## Observer - if you change the number of dependent objects upon other objects and their way of maintenance of the own state

Visitor - if you change the methods performed on the objects without change of their classes

Memento - if you change the  time and kind of stored information about the object outside it

Comand - if you change conditions and the way of realization of the request of the used object

State - if you change states of the used objects

Strategy - if you change the algorithms of the used projects

## https://lh3.googleusercontent.com/bMUKeiN7SiL4P9jcGOX7qqKr3ynI32pP7JB_r5o3wjtPYRXy-6m62GGN6lIRi9LYbQ-TZumPdvpXfyT0sujsY5Rut0nyvwTqh90-3ZO6Og7-BrJ_s86qRPgAKea8_w 4.1. Which are basic issues of the Business tier?

Business logic, transactions, data and services(EJB components, and other business objects)

Business Delegate/ Data transfer Object - should you use for increasing performance

transfer Object - should you use for remote calls

Business Delegate/

1. use for encapsulation of services and data

2. use for transforming exceptions from other tiers

3. use to increase an availability of services

Session facade -

1. use to centralize the transaction management

2. you use to centralize the safety management

3. you use to centralize of the security management

Applicaition service -

1. you use to decrease complexity of business logic

2. you use to centralize the business logic

3. you use to avoid the duplication of code

Business Object -

1. you use to reuse the business logic

2. you use to promote an object-oriented approach in implementation of the business model

3. you use for the separation of the persistence logic from the business logic

Composite Entity - use for improvement of software maintenance

5.1. Which are basic issues of the Presentation tier?

Login, session management, content creation, formatting, validation and content delivery(JSP pages, servlets, any other UI elements)

Application controller-

1.use to centralization of control

2. you use to promote an object-oriented approach

Intercepting Filter -

1. you use to reuse its implementation

2. for supporting the declarative configuration

Composite View - use for encapsulation of the structure and data of a functional fragment of the tier

Context object / Composite View -

1.decreases the performance

2. use for improvement of maintenance

Context object - improve the testability

**6.1. Which are basic issues of the Integration tier?**

Resource adapters, external systems, mechanisms for resource, control flow(JMS, JDBC, connectors, connections with external systems)

Domain store -

1. for encapsulating the structure and data of the persistence module

2. use for the separation of persistence logic from business logic?

Data access object -

1. decreased complexity of data and structure of the persistence module

2. to promote an object-oriented approach in implementation of the business model

3. use for increasing the performance (based on caching and centralization of searching services)