Design patterns – christopher alexander – arhitect

Pattern – problem – solution – consequences

Problem – when to apply the pattern

Solution – abstract description

Consequences – Speed, extensibility…

Benefits – large-scale reuse of software architectures

Drawbacks – do not lead to a direct code reuse

Creational, Structural,Behavioral, Architectural

Creational :

Deal with object creation mechanisms

-Singleton- 1 instance – antipattern – console – zashto moje da e anti pattern – ti pazish state I toi ne se maha, pri pisane na unit testove trqbva da se pishat testove koito sa v opredelen red za da raboti

Pri mnogonishkovo programirane dvete nishki moje da zapochvat da instancirat singletona I taka stavat 2 obekta

-SimpleFactory- ne e pattern – vsichkite obekti se suzdavat na edno mqsto

-Factory Method – obekta se suzdava ot razlichna func, tq ne znae kak, no naslednicite znaqt

-Abstract Factory – obektite zavisqt edin ot drug – used In systems that are frequently changed

-Builder Pattern – logikata da se razdeli ot dannite / order dependent / too many too params /diff constr - subway sandwiches – give me this kind of bread, give me this kind of cheese, this kind of salad, this kind of salami

-Prototype pattern – factory for cloning new instance from a prototype, IClonable

-Object Pool

-Lazy initialization – kusno Inicializirane – dai mi kursove I togava se inicializirat I se davat

Structural:

Kak da si strukturirate klasovete za da moje v budeshte da ima nai malko promeni nujni

Kak da suzdavate vruzkata mejdu klasovete,  
-Facade – suzdava se izkustven klas koito podava publichnite raboti na klasovete izvun grupata

Kino sistema – primer e distancionno – kato natisnesh kopcheto to si pravi rabotata, a kak stava ne trqbva da znaesh za da go izpolzvash

-Composite Pattern – razlichni vidove obekti,da gi izpolzvame po edin I sushti nachin – da suzdavash std durvo -

Razlichni obekti s ednakva f(x)

-Proxy Pattern – dobavqne na oshte edno nivo na dostupvane, primerno zashtita bez realniq obekt da znae che go ima - web service

-Decorator – da dobavq novi statetove na klasovete dinamichno – reshava problem s nasledqvato, kogato ima mnogo za nasledqvane,

-Adapter Pattern – adapter s pristavka, middle man da natamani stariq da moje da se izpolzva

-Bridge Pattern – razdelq implementaciqta ot abstrakciqta, izpolzva se kogato trqbva da se dobavqt mnogo novi neshta, reshava se problema s nasledqvane

Proxy – to lazy-instantiate an obj, or hide the fact that you are calling a remote service or ctrl access

Lazy loading,

Decorator – to add func to an obj runtime, bez da extendvate

Adapter – to map an abstract interface to another obj which has similar f(x) role

Ili nqmate dustup do star kod, ili legacy sistemata e mn golqma

Bridge – you need to swap out diff implementations

-Flyweight Pattern – da namali resursite izpolzvani za edin obekt – bukvata a – da ne se suzdava nov obj, a da se preizpolzva

Behavioral:

Kak si komunikirat klasovete edin sus drug

-Chain of responsibility – izpulni tova, ako ne mojesh(nqmash pravomoshtiq) dai go na sledvashtiq po verigata

Dev > Team Leader > Manager > …  
-Iterator – obikalqte kolekciq, dori ako kolekciite sa razlichni – po edin I sushti nachin

Foreach

-Command – otdelen klas command da izpulnqva komandata – komandata znae na koi da kaje da q execute-va

Abstract command – calculator command – execute / unexecuted – user

-Template Method – za da se izpulni tozi algoritm trqbva da se izpulnqt tezi 5 stupki, az ne znam kak se izpulnqvat, bazoviq klas izvikva metoda s naslednicite(te znaqt kak se izpulnqva metoda)

HotDrink – boil water – brew – PourInCup – addSpices - override v naslednika na metoda

-Strategy Pattern – otdelen algo kato klas – nqkolko algo koito da se zamestvat

SortedList – sortStrategy : mergeSort / quicksort

-Observer – da komunikirate mejdu dva klasa bez te da znaqt mnogo edin za drug

Stock – Investor –

-Mediator – middle man v komunikaciq mejdu dva klasa –

Chat – 5 choveka – chata svurzva razlichnite hora, dokato 1 chovek znae samo za chata

-Memento Pattern – po vsqko edno vreme da se zapazva state I po kusno da se restorne pri jelanie

Ima otdelen klass koito da pazi state

-State Pattern – promenq se izpulnenieto na nqkakva komanda v zavisimost ot suotvetniq state na suotvetniq obekt

Account – state : silver, gold, red

-Interpreter – da se vkluchi drug ezik za programirane, da se interpretira nqkakuv input

Expression : oneExpression, tenExpression, hundredExpression, thousand

-Visitor Pattern – dava da posetite nqkakuv klas I da dobavite nqkakva operaciq bez da promenqte konkretno samiq klas –

Element : employee : President/Clerk/ - IncomeVisitor-Employees

-Null Object -/

Architectural:

Client – server architecture

-server/ client

server– a single machine/ app that provides services to multiple clients

client

- 3-tier architecture – front-end / client layer, middle-tier/ business logic, back-end/ data layer

-multi-tier architecture –

Data – tier / data – access tier / business tier / Presentation logic tier / GUI

-MVC - model/ data – view/display – controller/interaction

Mvc ne zamenq mnogosloinata arhitektura

-MVC presenter – kogato nqma debeli logiki, podgotvq danni I gi predstavq

-Model –view- ViewModel(MVVM) – sedi I slusha I ako neshto se promeni promenq I drugoto, sinhronizira dannite

-SOA – service-oriented arch – vrushta dannite, opravqi se