DIZAYN PATTERNLAR

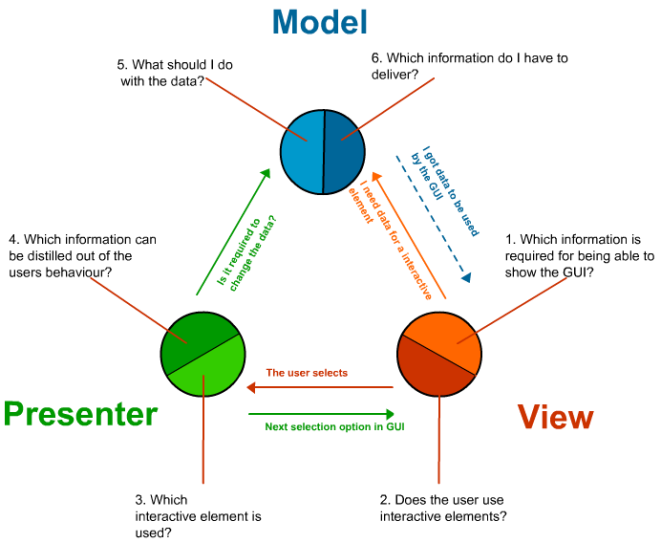
Saidova Feruza Komilovna .

Buxoro viloyati, Vobkent tumani 1-son kasb-hunar maktabi

Annotatsiya: Bugun, 20-dekabr kuni Axborot texnologiyalari va kommunikatsiyalarini rivojlantirish vaziri Sherzod Shermatov «Toshkent Yoshlar forumi-2021» tadbirida «Yoshlar kelajagi uchun IT-sohadagi imkoniyatlar» mavzusida chiqish qildi. Aytish joiz, yoshlarning vazir bilan uchrashuvi ochiq muloqot shaklida qiziqarli o‘tdi. *Iqtisodiyotimiz yaqin yillar ichida yanada barqaror, o‘ziga baquvvat, jahon va mintaqaviy bozorlarda raqobatdosh bo‘lmogi uchun iqtisodiyotimizni tarkibiy o‘zgartirish va diversifikatsiya qilish bo‘yicha hali ko‘p ish qilish lozimligini ham biz o‘zimizga yaxshi tasavvur etamiz. «O‘z-o‘zidan ravshanki, bularning barchasi iqtisodiyotimizning yuqori suratlar bilan barqaror o‘sib borayotgani va mamlakatimizda ro‘y berayotgan ulkan ijobiy o‘zgarishlardan dalolat beradi.*

Bozor sharoitida milliy axborot kommunikatsiyalarini kirib kelishi, muhandis-pedagoglar oldiga iqtisodiy, pedagogik, psixologik bilimlarga ega, korxonalarni boshqarish, modellashtirish, avtomatlashtirish, shuningdek, kam sarf-harajatli arzon, mustahkam uzoqqa chidamli ishlab chiqaruvchi texnologiyalarni matematika, informatika, matematik dasturlash, statistika, axborot texnologiyalari, iqtisodiyot nazariyasi, sotsialogiya, pedagogika va boshqa fanlardan o‘zlashtirgan bilimlari asosida o‘rgangan yetuk mutaxassis kadrlar bo‘lishlikni taqozo etmoqda.

Kalit so`zlar: **Dizayn Patternlar**



Universitetdagi o'qishim mobaynida olgan kurslarim ichida eng qiziqarlisi va muhimlaridan biri shu kunlarda olayotgan kursim **ASD(Advanced software design)**dir. Ushbu kursda asosiy e'tiborimizni OOP ning asosi bo'lgan Design Patternlarga qaratyapmiz. Poliformizm, Enkapsulatsiya, Inheritance kabi muhim prinsiplarni to'gri ishlatish, biror bir kontekstdagi muammo uchun aniq bir Design Patterni qo'llash kabi muammolarni ko'rib chiqyapmiz.

Design Patternlarni algoritmlar va ma'lumotlar strukturasiga aloqasi yo'qdek tuyuladi. Lekin komputer olamida har ikkala yirik bo'limning vazifasi aniq bir muammoni yechimini topishdir. Shu sabab ushbu blogda Design Patternlarga ham imkon qadar to'htalib o'tib ba'zi bir misollarni berib borishga qaror qildim.

Design patternning o'zi nima?

Design Pattern bu qayta-qayta qo'llash, ma'lum bir kontekstda aniq muammoni hal qilish uchun foydalanish mumkin bo'lgan yechim, yillar davomida dasturchilar tomonidan yig'ilib borgan eng yaxshi tajribalar to'plamidir. Design patternning o'zi muammo hal qilmaydi, aksincha u abstrakt fikr bo'lib class, interface, metodlarning o'zaro bog'lanishlari ko'rinishida beriladi. Misol tariqasida, bino qurilishi davomida quruvchilarning avval poydevor, keyin devor keyin esa tom qurish ketma-ketligini keltirishimiz mumkin. Undan tashqari OOP sifat talablarining mukammal tarzda

ishlatilgan holati. Maqolalar seriyasi davomida ushbu prinsiplarning qanday qilib yo'lga qo'yilishi haqida ma'lumot berib borishga xarakat qilamiz. Design Patternlar haqida yozilgan shu kunga qadar eng mashhur kitob bu Design Patterns: Elements of Reusable Object-Oriented Software bo'lib, bu shu darajada mashhurki kitob muallfilariga **GOF(Gang of Four)**laqabini olib bergan. Ushbu to'rtlik kompyuter olamida eng ko'p eslanadigan insonlardan hisoblanishadi.

Design Patternlar oilasi.

Design pattern tushunchasi dastlab GOF tomonidan qo'llanilgan, aynan GOF ko'p yillik tajribalarga ko'ra design patternlarning ishlatilish holatiga ko'ra 3 hil yirik oilaga bo'lgan. Quyida ushbu ro'yxat berilgan.

Creational. Obyekt yaratilishi jarayonida qo'llanadigan	**Structural. Class yoki obyektlarning tuzilishi, tarkibining ifodalashida qo'llaniladigan**	**Behavioral. Class yoki obyektlarning o'zaro muomalasining turli ko'rinishlarida ishlatiladigan**
Factory	Adapter	Interpreter
Abstract Factory	Bridge	Template Method
Prototype	Composite	Chain of responsibility
Singleton	Decorator	Command
Facade	Iterator	

Creational. Obyekt yaratilishi jarayonida qo'llanadigan	**Structural. Class yoki obyektlarning tuzilishi, tarkibining ifodalashida qo'llaniladigan**	**Behavioral. Class yoki obyektlarning o'zaro muomalasining turli ko'rinishlarida ishlatiladigan**
Flyweight	Mediator	
Proxy	Memento	
Observer		
State		
Strategy		
Visitor		

OOP(Obkyektga yo'naltirilgan dasturlash)ning asosiy tushuncha va prinsiplari.

Har qanday binoni qurishda ishlatiladigan umumiy qabul qilingan qoidalar mavjud bo'lgani kabi OOPning ham o'z qoidalari mavjud. Quyida ularga qisqacha to'xtalib o'tamiz. OOPning 3 ustuni deganda odata poliymorfizm, enkapsulatsiya va meros olish tushuniladi.

Poliymorfizm

Obyektga abstrakt darajada qarash hususiyati. Masalan, futbolda turli futbol jamoalari mavjud: ayollar futbol jamoasi, yoshlar fubol jamoasi, mini futbol

jamoasi. Shu obyektlarga alohida obyekt emas umumiy qilib futbol jamoasi sifatida qarash imkonini beradi. Yana boshqacha qilib aytganda turli hil obyektlar bilan bir hil uniformada ishlash imoniyati desak bo'ladi. Design patternlarning P2I prinsipining asosini tashkil qiladi.

Enkapsulatsiya

Ma'lumotlar va funksiyalarni bir komponent ichiga yig'ishga atyiladi. Buning uchun classlardan foydalaniladi. Enkapsulatsiya tanlov asosida classning ba'zi hususiyatlarini foydalanuvchidan yashirish imkonini beradi. Ushbu jarayonga misol sifatida avtomobil minayotgan haydovchini olishimiz mumkin. Xaydovchi tormoz pedalni bosganda, pedalning ishlash jihatlaridan bexabar bo'ladi. Sababi unga bu bilimning keragi yo'q. Pedal ichidagi murakkab mexanizm foydalanuvchidan yashirilgan bo'ladi.

Meros olish(Inheritance)

OOPning uchinchi ustuni. Yuqorida berilgan ustunlar bilan doim birga yuradi. Ma'lum obyekt asosida boshqa obyektni yaratish jarayoniga aytiladi. Bir classning boshqa classdan meros olishi yordamida amalga oshiriladi. Meros olingan obyekt ota obyektdagi xususiyatlarni tanlovga ko'ra meros oladi. Masalan, avtoulov bu ota obyekt. Bu obyekt yordamida yengil mashina, yuk mashinasi, poyga mashinasi kabi boshqa obyektlarni yaratib olishimiz mumkin. Ota classda bo'lgan 4 g'ildirak farzand classlarda ham mavjud bo'ladi. Ya'ni poyga mashinasi, avtoulovdan g'ildiraklarni meros oladi.

Design patternlar uchun bir nechta sifat ko'rsatkichlari(prinsiplar) mavjud. Quyida ushbularni sanab o'tamiz.

Obyektlarning juftligi(Coupling)

Ikki classning bir-biriga o'zaro bog'lanishi, ya'ni o'zaro mustaqillik darajasiga aytiladi. Obyektlar bir-biridan qancha taqozo qilsa shuncha yomon. Design Patternlarning asosiy maqsadlaridan biri couplingni kamaytirish hisoblanadi. Yana tight coupling ham deyiladi bu holatda, class boshqa classning ichidagi ma'lumotlardan bohabar bo'ladi. Natijada esa u yoki bu classni o'zgartirish qiyin kechadi. Loose Coupling tushunchasi bu o'zaro bog'lanishlar sonini minimum darajada kamaytirish jarayoniga aytiladi.

Obyektlarning birligi(Cohesion)

Birligi deganda o'lchov birligi nazarda tutilmadi. Obyektlarning o'zaro kamchiliklarini to'ldirishiga aytiladi. Ya'ni ular birgalikda huddi bir jismdek mavjud bo'ladi. Yana boshqacha qilib aytaganda obyektlarning axilligi, yoki professorimning sevimli so'zi bo'lgan – obyektlarning garmoniyasi tushuniladi. Misol sifati odamning tana a'zolarini keltirishimiz mumkin. Qo'l, oyoq yoki shu kabu boshqa a'zolarga alohida obyekt sifatida qarashimiz mumkin, lekin ular birgalikda mukammal tizimni tashkil qiladi. Demak, Design Patternlarning vazifalaridan biri yuqori darajadagi Cohesionga erishish.

SRP.(Single Responsibility Prinsiple – Yagona ma'suliyat prinsipi)

Bir class yoki metodga faqat bir turdagi vazifani yuklashga aytiladi. Agar g'isht teruvchiga kranni boshqarishniyam topshirasangiz yoki devor qulaydi yoki tom.

O'zinggi o'zing qaytarma prinsipi(Don't Repeat Yourself-DRY)

Aynan bir hil vazifa bajaruvchi class yoki metodlar sonining kamligi darajasi. Masalan, avtorizatsiya mantiqi uchun faqat bir dona class yaratish qoidasi. Agar kodning turli qismlarida bir hil vazifali class yoki metodlar paydo bo'lsa uni darrov abstrakt darajaga olib chiqishga xarakat qilinadi.

Ochiq/Yopiq prinsipi(Open/Closed)

Misol sifatida USB asboblarni keltirishimiz mumkin. Aytaylik kompyuterimizda WI-FI qurilmasi yo'q. Kompyuterni ochib o'zgaritirish imkoniga ega emasmiz, USB qurilma sotib olib ushbu muammoni hal qilishning esa imkoni bor. Ya'ni kengaytmalarga ochiq, o'zgarishlarga yopiq.

LSP(Liskov Substitution Principle – Liskovning o'rin almashish prinsipi)

Ilk bor Liskov ismli ayol tamonida ilgari surilgan. Aytaylik F classimiz boshqa R classidan meros olinib yaratilgan. LSPning asosiy ma'nosi shuki, biz R classni ishlatish davomida F classidan foydalanishimiz mumkin. Boshqacha qilib aytganda ota borishi kerak bo'lgan nahorgi oshga o'g'il borishi mumkin degan ma'noni beradi.

Abstraksiya darajasi(Abstraction)

Obyekt bu real jism. Abstraksiya esa ushbu jismni reallikdan uzoq deb tassavur qilib u haqida fikr yuritishga aytiladi. Abstraksiyaga fikrlar to'plami sifatida ham qarasa bo'ladi. Misol sifatida choyxonada o'tirgan chollarning suhbatini olishimiz mumkin. Aytaylik ular AQSH va Rossiya o'rtasidagi munosabatni muhokama qilishmoqda. Putinning ohirgi yillarda qilgan ishlarini gapirayotib, qanday qilingan, qayerda qilingan, nima vositasida qilinganligini umuman e'tiborga olmasdan uning natijalari haqida fikr yuritish sifatida qarasak bo'ladi. Design Patternlarni abstraktlik darajasi yuqoriliga qarab baholanadi.

P2I. (Programming to Interface – Interfeysga tomon dasturlash)

Abstraksiyaning asosiy ustuni hisoblanadi. Biror muammoni hal qilinayotganimizda konkret classga emas, uning otasi interface ustida ishlashga urinishga aytiladi. Masalan, savdo markazlarida, odamlarga qulaylik sifatini oshirish ustida ishlanyapti deylik. Bu jarayonda erkak, ayol yoki bolalarga alohida qarab emas, umumiy – odamlar so'zi ishlatiladi. Ya'ni "Odamlarga qulay bo'lsin!!!".

VOD. (Variation Oriented Design – O'zgarishlarga yo'naltirilgan design)

Designning ko'p o'zgaradigan qismiga alohida, kam o'zgaradigan qismiga alohida qarab ishlashga aytiladi. Ko'p o'zgaradigan qism enkapsulatsiya qilinadi. Ochiq/Yopiq prinsipiga erishish uchun ushbu jarayon amalga oshiriladi.

DIP. (Dependency Inversion Principle)

Yuqorida darajadagi modul pastki darajadagi moduldan taqozo qilmasligiga aytiladi. Yoki, abstraksiya konkret obyektdan taqozo qilmasligi kerak aksincha obyekt asbtraksiyaning qoidalariga amal qilishi lozim.

Keyingi maqolamizda, Observer design patterni haqida so'z yuritamiz.

FOYDALANILGAN ADABIYOTLAR RO`YXATI:

1. O‘zbekiston Respublikasi Prezidentining 2020-yil 5-oktabrdagi PF-6079-son Farmoni

2. A.Axmedov.N.Toyloqov. “Informatika”, T.,2001 y.

3. A.Aripov, A.Xaydarov.”Informatika asoslari”. 2002 y.

4. A.Abduqodirov, A.Xayitov. “Axborot texnologiyasi”. 2002 y.

5. G.Yormatov, YO.Isamuhamedov.”Mexnatni muhofaza qilish”. 2002 y

6. Y.G.Sibarov «Охрана труда в вычислительных центрах»

7. K.Yo’ldoshev, Sh.Mamatkulov, K.Muftaydinov.”Iqtisodiyot asoslari” 2002