

KLASS DIAGRAMMALARI VA HOLATLAR DIAGRAMMALARI

Muallif: Ibrohimov Laziz

Samarqand davlat universiteti Urgut filiali,

Biznesni boshqarish va tabiiy fanlar

fakulteti, 3-bosqich talabasi

✉ lazizibrohimov04@gmail.com

Ilmiy rahbar: Suyarov Akram

Samarqand iqtisodiyot va servis instituti, dotsent

✉ akramsuyarov@mail.ru

Annotatsiya

Ushbu maqolada obyektga yo'naltirilgan dasturlash va tizim modellashtirish jarayonida keng qo'llaniladigan UML klass diagrammalari va holatlar diagrammalarining mazmuni, amaliy ahamiyati, qurilish qoidalari va ular o'rtasidagi farqlar yoritiladi. Har bir diagramma qanday ma'lumotlarni aks ettirishi, dasturiy ta'minot arxitekturasini loyihalashdagi roli va foydalanuvchi ehtiyojlariga mos modellar yaratishdagi hissasi misollar orqali izohlanadi.

Kalit so'zlar: *UML, klass diagrammasi, holatlar diagrammasi, obyektga yo'naltirilgan modellashtirish, tizim dizayni, dasturiy injiniring.*

1. Kirish

Zamonaviy dasturiy ta'minot tizimlari murakkab va ko'p qirrali bo'lib, ularni to'g'ri loyihalash va tushunarli tarzda modellashtirish dasturchilar va tizim muhandislari uchun katta ahamiyatga ega. UML (Unified Modeling Language) — bu dasturiy tizimlarning struktura va xatti-harakatlarini vizual tarzda ifodalashga xizmat qiluvchi universal til hisoblanadi. UML diagrammalari orasida eng muhimlaridan ikkitasi bu klass diagrammalari va holatlar diagrammalaridir. Ushbu maqolada aynan shu diagrammalar haqida batafsil ma'lumot beriladi.

2. Asosiy qism

2.1. Klass diagrammalari haqida tushuncha

Klass diagrammasi — bu obyektga yo'naltirilgan tizimning statik tuzilmasini aks ettiruvchi diagramma bo'lib, unda klasslar, ularning atributlari, metodlari va o'zaro aloqalari ko'rsatiladi. Bu diagramma tizimning arxitekturasini va obyektlar orasidagi bog'liqlikni tahlil qilishda muhim vositadir.

Klass diagrammasining asosiy elementlari:

- **Klass:** tizimdagi obyektlar toifasini ifodalaydi.
- **Atributlar:** klassga tegishli xususiyatlar (masalan, ism, ID, narx).
- **Metodlar:** klassga tegishli funksiyalar (`hisobla()`, `qidir()`).

- **Aloqalar (Relationship):**
 - **Assotsiyatsiya** – oddiy aloqa;
 - **Agregatsiya** – "butun va qism" munosabati;
 - **Kompozitsiya** – kuchli "butun va qism";
- **Merosi olish (Inheritance)** – vorislik munosabati.

Misol: Agar biz kutubxona tizimini loyihalashtirsak, **Kitob**, **Foydalanuvchi**, **Admin** kabi klasslar bo'lishi mumkin. **Kitob** klassi **nomi**, **muallifi**, **sana** kabi atributlarga va **qidir()** yoki **band qilish()** kabi metodlarga ega bo'ladi.

2.2. Holatlar diagrammasi haqida tushuncha

Holatlar diagrammasi (State diagram) — bu tizim yoki uning komponentlari vaqt o'tishi bilan qanday holatdan holatga o'tishini tasvirlovchi diagrammadir. Bu diagramma foydalanuvchi harakati, tizim ichki logikasi yoki tashqi omillar ta'sirida sodir bo'ladigan holat o'zgarishlarini ifodalaydi.

Holatlar diagrammasining asosiy elementlari:

- **Boshlang'ich holat (Initial State):** tizim ish boshlagan vaqtdagi holat.
- **Holatlar (States):** tizimning mavjud bo'lishi mumkin bo'lgan holatlari.
- **O'tishlar (Transitions):** bir holatdan boshqasiga o'tish.
- **Trigger (voqealar):** holat o'zgarishini keltirib chiqaruvchi sabab.

Misol: **Buyurtma** klassining holatlari bo'lishi mumkin: **Yaratildi**, **To'landi**, **Yetkazilmoqda**, **Yopildi**. Har bir foydalanuvchi harakati bu holatni o'zgartiradi.

2.3. Diagrammalar o'rtasidagi farq va o'zaro bog'liqlik

Klass diagrammalari tizimning tuzilmasini, holatlar diagrammalari esa uning xatti-harakatini ifodalaydi. Klass diagrammasi orqali tizimdagi ob'ektlar va ularning o'zaro aloqalari aniqlansa, holatlar diagrammasi orqali bu ob'ektlarning vaqt davomida qanday o'zgarishini ko'rish mumkin.

Diagramma turi	Maqsadi	Yondashuv
Klass diagrammasi	Tizim strukturasi modellashtirish	Statik
Holatlar diagrammasi	Xatti-harakatni modellashtirish	Dinamik

2.4. Amaliy ahamiyati va tavsiyalar

Tizimni samarali loyihalash uchun klass va holatlar diagrammasini birgalikda qo'llash tavsiya etiladi. Quyidagi metodologiyalar foydalidir:

- **Model-Driven Development (MDD):** tizimni diagrammalar asosida yaratish.
- **Iterativ yondashuv:** diagrammalarni dastlabki versiyalaridan boshlab asta-sekin to'ldirib borish.
- **CASE vositalari:** Visual Paradigm, StarUML, Enterprise Architect kabi dasturlar orqali diagrammalarni yaratish va boshqarish.

Xulosa

Klass va holatlar diagrammalari dasturiy ta'minot loyihalashda eng muhim modellashtirish vositalaridan biridir. Klass diagrammasi orqali tizimning tuzilmasi

aniqlanadi, holatlar diagrammasi esa tizimning dinamik o'zgarishlarini tasvirlaydi. Har ikki diagrammani birgalikda tahlil qilish orqali dasturiy mahsulotni aniq, tushunarli va mukammal loyihalash mumkin bo'ladi. Diagrammalar orqali tizim dizaynini soddalashtirish, xatolarni kamaytirish va dasturchilar o'rtasida samarali hamkorlikni yo'lga qo'yish mumkin.

Foydalanilgan adabiyotlar

1. Jalilov A. (2021). Dasturiy injiniring asoslari. Toshkent: TATU nashriyoti.
2. Qodirov M., Xayitov A. (2020). UML diagrammalari va ularni qo'llash. Toshkent: Innovatsiya nashriyoti.
3. Karimov R. (2022). Kompyuter tizimlari va dasturiy modellashtirish. Samarqand: SamDU nashriyoti.
4. Booch G., Rumbaugh J., Jacobson I. (2005). The Unified Modeling Language User Guide. Addison-Wesley.
5. Fowler M. (2004). UML Distilled: A Brief Guide to the Standard Object Modeling Language. Addison-Wesley.
6. Sommerville I. (2016). Software Engineering. 10th ed. Pearson.
7. StarUML Rasmiy qo'llanmasi. <https://staruml.io>