

Dizayn paternlar (design patterns) nima?

Dizayn paternlar dasturlashda paydo bo'ladigan umumiy muammolarga o'ziga xos yechimlardir. Boshqacha aytganda dizayn paternlar kodingizda doim takrorlanib turadigan muammoni yechish uchun o'zingizga moslab olib ishlatishingiz mumkin bo'lgan, oldindan tayyorlab qo'yilgan yechim bo'lib hisoblanadi.

Siz hech qachon aynan bitta dizayn patternni olib o'zingizning kodingizda to'g'ridan to'g'ri ishlata olmaysiz. Ya'ni u birorta funksiya yoki kutubxona bo'lib hisoblanmaydi. Dizayn patern biror kodning bo'lagi ham emas. U biror muammoning umumiy yechimi xolos. Siz, faqatgina, dizayn paternda berilgan yo'riqnomaga amal qilgan holda o'zingizning kodingizni davom ettirishga va ishlatishga qulay qilib yozishingiz mumkin.

Paternlar, ko'pincha, algoritmlar bilan chalkashtiriladi, chunki ikkalasi ham biror muammoga o'ziga xos yechim bo'lib hisoblanadi. Biroq, algoritmlar doimo biror natijaga erishishdagi aniq harakatlar to'plami bo'lsa, paternlar esa yechimning biroz yuqori darajadagi ko'rinishi hisoblanadi.

Paternlar tarixi

Patternlarni kim ixtiro qilgan? Bu juda qiziq, lekin, ayni vaqtda biroz noto'g'ri savol. Yuqorida aytilganidek, dizayn paternlar obyektga yo'naltirilgan dasturlashdagi umumiy muammolarga o'ziga xos yechimlardir. Biror yechim turli loyihalarda qayta va qayta takrorlanib kelaverganidan keyin, axiryri biror kishi unga nom qo'yadi va uni umumiy ko'rinishda tushuntiradi. Mana shunday qilib dizayn paternlar paydo bo'ladi.

Paternlar tushunchasi birinchi bo'lib Kristofer Aleksandr tomonidan Patern tili: shaharlar, binolar, qurilish (A Pattern Language: Towns, Buildings, Construction) kitobida tushuntirilgan. Kitob shahar muhitini loyihalash uchun kerak

bo'lgan "til"ni tavsiflagan. Bu tilning qismlari paternlar hisoblangan. Paternlar, masalan, baland derazalar qanday bo'lishi mumkin, ko'p qavatli binolar qanday qurilishi kerak, katta istirohat bog'lari qanday bo'ladi va shu kabilarni tushuntirgan.

Bu g'oyani, keyinchalik, 4 ta muallif: Erix Gamma, Jon Vlissidis, Ralf Jonson va Richard Helmlar ham qo'llashgan. 1994-yilda ular Dizayn paternlar: qayta ishlatiluvchi obyektga yo'naltirilgan dasturlash elementlari (Design Patterns: Elements of Reusable Object-Oriented Software) kitobini nashr qilishdi. Ular bu kitobda Kristofer Aleksandrning g'oyasini foydalanishdi. Kitobda obyektga yo'naltirilgan dasturlashdagi muammolarga yechim bo'luvchi 23 ta paternlar keltirilgan edi va bu kitob tezda eng ko'p sotiladigan kitoblar qatoriga kirdi. Nomi uzun bo'lgani uchun odamlar uning nomini "gangster to'rtlikning kitobi (the book by the gang of four)" deb nomlashdi va keyinchalik "GoF kitobi(the GoF book)" deb qisqartirildi.

O'shandan buyon, ko'plab boshqa dizayn paternlar yaratildi. Hozirga kelib "patern yondashuvi" boshqa dasturlash sohalarida ham ommalashgan. Shu sababli, obyektga yo'naltirilgan loyihalashdan tashqarida ham ko'plab boshqa paternlar mavjud.

Nega paternlarni o'rganish kerak?

Haqiqat shundaki, siz hech qanday paternni bilmay turib ham ko'p yillar davomida dasturchi sifatida ishlab yurishingiz mumkin. Ko'pchilik dasturchilar ham aslida shunday qiladi. Hattoki, siz biror paternni bilmay turib uni kodingizda ishlatishingiz ham mumkin. Xo'sh, nega siz vaqtingizni sarflab ularni o'rganishingiz kerak?

- Dizayn paternlar dasturlashdagi ko'p uchraydigan umumiy muammolarga sinalgan yechim bo'lib hisoblanadi. Hattoki, siz bu muammoga hech qachon duch kelmagan bo'lsangiz ham, paternlarni bilish foydalidir, chunki paternlar obyektga yo'naltirilgan dasturlash tamoyillaridan foydalanib barcha turdagi muammolarni yechishni o'rgatadi.

- Dizayn paternlar aynan bitta til uchun mo'ljallanmagan. Uni xohlagan tilingizda ishlatishingiz mumkin. Har qanday tilda ishlovchi dasturchidan patern bo'yicha maslahat olishingiz mumkin.

Paternlar tasnifi.

Paternlar murakkabligi, batafsillik darajasi va qo'llanilish ko'lami bo'yicha farqlanadi.

Eng asosiy va past darajadagi paternlar ko'pincha **idiomlar** deb nomlanadi. Ular odatda bitta dasturlash tilidagina foydalaniladi.

Eng universal va yuqori darajali paternlar **arxitekturali paternlar** deb nomlanadi. Dasturchilar bu patternlarni virtual holatda har qanday dasturlash tilida ishlata oladilar. Boshqa paternlardan farqli ravishda, ular butun ilovaning arxitekturasini loyihalashda qo'llanilishi mumkin.

Shu bilan birga, barcha paternlar maqsadiga ko'ra kategoriyalanadi. Ushbu qo'llanma paternlarning uchta asosiy guruhidan tashkil topgan:

- Yaratuvchi paternlar (**Creational Patterns**) mavjud kodni qayta ishlatishga qulay va moslashuvchanligini oshirishda xizmat qiladigan obyektни yaratish mexanizmini ta'minlaydi.
- Tuzilmali paternlar (**Structural Patterns**) kodning tuzilishini samarali va moslashuvchan saqlagan holda obyekt va klasslarni katta tuzilmalarda qanday qilib birlashtirishni tushuntiradi.
- Xarakterli paternlar (**Behavioral Patterns**) obyektlar orasidagi javobgarlik vazifasi va aloqaning samaradorligi bilan shug'ullanadi.