VORISLIK VA CHATISHTIRISH

Tojimamatov Isroil Nurmamatovich
Farg'ona davlat unversiteti o'qituvchi, israiltojimamatov@gmail.com
Muxsinova Sevinchxon Ikromjon qizi
Farg'ona davlat unversiteti 3-kurs talabasi
akramovasevinchxon08@gmail.com

Annotatsiya:Ushbu maqolada vorislik va chatishtirish tushunchalari, ularning dasturiy ta'minot arxitekturasi va obyektga yo'naltirilgan dasturlashdagi ahamiyati o'rganiladi. Shuningdek, ushbu usullarning zamonaviy dasturiy texnologiyalardagi qo'llanilishi, afzalliklari va muammolari ko'rib chiqiladi.

Annotation: This article examines the concepts of inheritance and polymorphism, their importance in software architecture and object-oriented programming. It also discusses their application in modern software technologies, advantages, and challenges.

Аннотация: В данной статье рассматриваются понятия наследования и полиморфизма, их значение в архитектуре программного обеспечения и объектно-ориентированном программировании. Также обсуждаются их применение в современных программных технологиях, преимущества и проблемы.

Kalit so'zlar: Vorislik, Chatishtirish, Obyektga yo'naltirilgan dasturlash, Polimorfizm, Dasturiy ta'minot arxitekturasi.

Key words: Inheritance, Polymorphism, Object-oriented programming, Software architecture.

Ключевые слова: Наследование, Полиморфизм, Объектно-ориентированное программирование, Архитектура программного обеспечения.

Kirish

Vorislik va chatishtirish obyektga yo'naltirilgan dasturlash (OOP) ning eng asosiy tushunchalari hisoblanadi. Ushbu tushunchalar dasturiy tizimlarni qayta foydalanish va moslashuvchanlikni oshirish maqsadida ishlatiladi. Vorislik bir sinfning xususiyat va metodlarini boshqa sinfga o'tkazish imkonini bersa, chatishtirish esa ob'ektlar va metodlarning turli ko'rinishlarda ishlashini ta'minlaydi. Zamonaviy dasturlash tillari, jumladan, Java, C++, Python, va C# kabi tillar ushbu tushunchalarni o'zining asosiy elementlari sifatida qabul qilgan. Sun'iy intellekt (SI) hozirgi kunda ilm-fan va texnologiya sohalarida katta o'zgarishlarni keltirib chiqarmoqda va yangi texnologiyalarni rivojlantirishga turtki bermoqda. Sun'iy intellektda vorislik deganda tizimlarning bir-biridan xususiyat va bilimlarni "meros qilib olishi" hamda o'zaro

bog'lanishi tushuniladi. Vorislik va chatishtirish (inglizcha inheritance va polymorphism) obyektga yo‘naltirilgan dasturlash (OOP) ning eng muhim tushunchalari bo‘lib, sun'iy intellekt dasturlarini samarali boshqarish, kodni qayta ishlatish va turli maqsadlarga moslashuvchan tizimlar yaratishga yordam beradi.

Vorislikning mohiyati

Vorislik – obyektga yo‘naltirilgan dasturlash (OOP) tamoyillaridan biri bo‘lib, dasturiy tizimlarni samarali boshqarishda va qayta foydalanishda muhim ahamiyatga ega. Bu tamoyil bir sinfning xususiyatlari va metodlarini boshqa sinfga o‘tkazish imkoniyatini beradi. Vorislik orqali yangi sinf eski sinfning barcha xususiyatlariga ega bo‘lib, zaruratga qarab o‘zining xususiyat va metodlarini qo‘shishi yoki mavjudlarini o‘zgartirishi mumkin. Ushbu yondashuv kodni takrorlamaslik, strukturaviylikni saqlash va tizimni kengaytirishni osonlashtirishda katta yordam beradi.

Vorislikning asosiy turlari

1. Yagona vorislik (Single Inheritance)
   Yagona vorislikda bir sinf faqat bitta asosiy sinfdan (ota sinf) meros oladi. Bu dasturlashning oddiy va ko‘p hollarda yetarli bo‘lgan yondashuvlaridan biri. Masalan, hayvonlar haqidagi dasturda "Hayvon" degan asosiy sinfdan "Mushuk" yoki "It" sinflarini hosil qilish mumkin.
2. **Ko‘p darajali vorislik (Multilevel Inheritance)**
   Bu turda vorislik bir nechta darajalarda amalga oshiriladi, ya’ni bir sinf boshqa sinfdan meros oladi, keyin bu sinfning o‘zi ham yangi sinf uchun asos bo‘lib xizmat qiladi. Masalan, "Hayvon" sinfidan "Quyruqli hayvon", undan esa "Mushuk" sinfi yaratilishi mumkin.
3. **Ko‘p tomonlama vorislik (Multiple Inheritance)**
   Bu usulda bir sinf bir nechta ota sinflardan meros oladi. Masalan, "Uchuvchi" va "Hayvon" sinflaridan "Uchuvchi qush" sinfi hosil qilinishi mumkin. Biroq ba’zi dasturlash tillari, masalan, Java, bu turdagi vorislikni qo‘llab-quvvatlamaydi, chunki u murakkablikni oshirishi va "nomuvofiqlik muammosi" (diamond problem) yuzaga kelishiga sabab bo‘lishi mumkin.

Vorislikning Afzalliklari

1. **Kodning qayta ishlatilishi**
   Vorislik yordamida bir marta yozilgan kod boshqa sinflar tomonidan foydalanilishi mumkin. Bu dasturlarni yaratishda va qo‘llab-quvvatlashda vaqtni tejaydi hamda tizimni soddalashtiradi.
2. **Dasturiy tizimni kengaytirish osonligi**
   Vorislik tizimni kengaytirishni osonlashtiradi. Yangi sinflarni yaratishda mavjud sinflarning xususiyatlaridan foydalanish imkoniyati tufayli dasturiy kodni boshqarish qulay bo‘ladi.
3. **Moslashuvchan dasturiy arxitektura yaratish**

Vorislik kodni modularlashtirish va strukturaviy tartibni saqlash imkonini beradi. Bu yondashuv dasturiy tizimlarni murakkablik darajasidan qat'i nazar boshqarishni yengillashtiradi.

Vorislik dasturlashning samaradorligini oshiruvchi va kodni qayta ishlatishga imkon beruvchi kuchli vositadir. Uning yordamida dasturiy tizimlarni kengaytirish va boshqarish osonlashadi. Biroq, uni ishlatishda o‘zaro bog‘liqlik va murakkablikni nazorat qilish muhim ahamiyatga ega. Vorislikni to‘g‘ri qo‘llash orqali dasturiy arxitekturani yanada mustahkam va moslashuvchan qilish mumkin.

Chatishtirishning mohiyati

Chatishtirish (polimorfizm) obyektga yo‘naltirilgan dasturlash (OOP) ning asosiy tamoyillaridan biri bo‘lib, bir metod yoki ob’ektning turli ko‘rinishlarda ishlash imkoniyatini ifodalaydi. Bu yondashuv dasturiy tizimlarni moslashuvchan, umumiylashtirilgan va samarali boshqariladigan qilishda muhim rol o‘ynaydi. Chatishtirish orqali bir xil metodni turli ob’ektlar uchun har xil usullar bilan ishlatish mumkin bo‘ladi, bu esa dasturiy kodni qayta yozmasdan moslashuvchanlikni ta’minlaydi.

Chatishtirishning asosiy turlari

1. **Statik chatishtirish (kompilyatsiya vaqtida chatishtirish)**
   Statik chatishtirish bir metodning bir nechta shaklda mavjud bo‘lishi, ya’ni metodning bir nechta versiyasini yaratishga imkon beradi. Bunday yondashuv metodni ortiqcha yuklash (method overloading) orqali amalga oshiriladi. **Misol:** Java dasturlash tilida bir xil nomli metod turli parametrlar soni yoki turlariga ega bo‘lishi mumkin. Bu metodni qo‘llashni sodda qiladi.
2. **Dinamik chatishtirish (ishlash vaqtida chatishtirish)**
   Dinamik chatishtirish bir metodning turli sinflarda farqli implementatsiyasini ta’minlaydi. Bu yondashuv metodni qayta yozish (method overriding) orqali amalga oshiriladi. Bu usulda asosiy sinfda (parent class) metod yaratiladi, va undan voris sinflar (child classes) uni o‘ziga moslab qayta yozadi. **Misol:** "Hayvon" sinfida "ovoz chiqarish()" metodi mavjud bo‘lsa, "It" sinfi bu metodni "vov-vov" ovoz chiqarishga moslab qayta yozishi mumkin.

Chatishtirishning Afzalliklari

1. Moslashuvchanlikni oshirish

Chatishtirish dasturiy tizimlarni moslashuvchan qilishda muhim o‘rin tutadi. Ushbu tamoyil yordamida kodni turli holatlarga moslashtirish va uni kengaytirish osonlashadi. Yangi funksiyalarni qo‘shishda yoki tizimni yangilashda chatishtirish katta qulaylik yaratadi. Bu yondashuv dasturiy loyihalarni turli sharoitlarga tez va samarali moslashish imkoniyatini beradi.

2. Kodning umumiylashtirilishi

Polimorfizm orqali bir xil metod yoki ob’ektni turli joylarda qayta-qayta qo‘llash mumkin. Bu dasturiy tizimda bir xil kod bloklaridan turli kontekstlarda foydalanishni ta’minlab, kodning samaradorligini oshiradi. Umumiylashtirilgan kod nafaqat dasturiy tizimni soddalashtiradi, balki uning tahlil va boshqaruvini ham osonlashtiradi.

3. Murakkab tizimlarni boshqarish samaradorligi

Chatishtirish dasturiy tizimlarning murakkablik darajasini samarali boshqarishga yordam beradi. Turli ob’ektlarni bir xil interfeys orqali boshqarish imkoniyati tufayli murakkab tizimlarda ham ishlash qulaylashadi. Masalan, bir xil metodni turli sinflarda ishlatish orqali turli funksiyalarni boshqarish osonlashadi. Bu esa dasturiy tizimlarni yaratish va ulardan foydalanishni sezilarli darajada samarali qiladi.

Chatishtirishning ushbu afzalliklari obyektga yo‘naltirilgan dasturlashda asosiy yondashuv sifatida uni qo‘llashni yanada muhim qiladi.

Chatishtirish obyektga yo‘naltirilgan dasturlashning asosiy tamoyillaridan biri bo‘lib, dasturiy tizimlarni moslashuvchan va samarali qilishda muhim ahamiyatga ega. Bu yondashuv yordamida kodni umumiylashtirish, yangi funksionallar qo‘shish va tizim murakkabligini boshqarish osonlashadi. Ammo chatishtirishni ishlatishda abstraksiyaning yaxshi rejalashtirilishi va resurslardan samarali foydalanish zarur. To‘g‘ri yondashilgan chatishtirish kodni strukturaviy, tushunarli va kengaytirishga qulay qilishda eng samarali vositalardan biridir.

Zamonaviy dasturlashda qo‘llanilishi

Vorislik va chatishtirish obyektga yo‘naltirilgan dasturlashning asosiy tamoyillari sifatida zamonaviy dasturiy texnologiyalar va ramkalar (frameworks) da keng qo‘llaniladi. Ular dasturiy tizimlarni samarali boshqarish, kodni qayta ishlatish va kengaytirishga katta yordam beradi. Quyida vorislik va chatishtirishning zamonaviy texnologiyalar va vositalardagi qo‘llanilishiga misollar keltiriladi.

- Spring Framework (Java)
  Java dasturlash tilidagi mashhur Spring Framework vorislik va chatishtirish tamoyillariga asoslanadi. Spring komponentlari o‘zaro bog‘lanish, qayta ishlash va kengaytirish jarayonlarida vorislikdan foydalanadi. Bu yondashuv orqali turli komponentlar bir xil interfeys orqali boshqariladi va polimorfizm yordamida tizimning moslashuvchanligi oshiriladi. Masalan, turli xil xususiyatlarga ega komponentlarni bir xil tamoyil asosida ishlatish imkonini beradi.
- Django Framework (Python)
  Django framework model darajasida vorislikni samarali qo‘llaydi. Ma’lumotlar bazasi bilan ishlashda asosiy model sinflaridan yangi sinflarni hosil qilib, ularning xususiyatlarini kengaytirish va o‘zgartirish imkonini beradi. Bu orqali ma’lumotlarni boshqarish jarayoni soddalashtiriladi va kodning strukturasi

yaxshilanadi. Misol sifatida, bir xil asosiy modelga ega turli ma’lumot turlari uchun moslashtirilgan sinflar yaratish mumkin.

- Unity Engine (C#)
  Unity o‘yin dvigatelida chatishtirish tamoyili keng qo‘llaniladi. O‘yin obyektlari va komponentlarni boshqarishda polimorfizm asosida turli xatti-harakatlarni yaratish mumkin. Masalan, "harakat qilish" funksiyasi har xil turdagi o‘yin obyektlari uchun moslashtirilgan holda ishlaydi, ammo asosiy interfeys bir xil bo‘lib qoladi. Bu yondashuv o‘yin loyihalarini boshqarish va kengaytirishda katta yordam beradi.

Xulosa

Vorislik va chatishtirish obyektga yo‘naltirilgan dasturlashning poydevorini tashkil etadi. Ular dasturiy tizimlarni kengaytirish, qayta ishlash va boshqarish imkoniyatlarini oshiradi. Kelajakda ushbu tushunchalar yanada rivojlanib, sun’iy intellekt va avtomatlashtirilgan dasturiy tizimlarda yangi imkoniyatlarni ochib beradi.

Foydalanilgan adabiyotlar

1. Nurmatovich, T. I. (2024). Bir qatlamli va ko ‘p qatlamli neyron to ‘rlari. ILM FAN XABARNOMASI, 1(1), 190-191.
2. Goldberg, D. E. (1989). Genetic Algorithms in Search, Optimization, and Machine Learning. AddisonWesley Longman Publishing Co., Inc.
3. Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep Learning. MIT Press. http://www.deeplearningbook.org.
4. Holland, J. H. (1992). Adaptation in Natural and Artificial Systems. University of Michigan Press.
5. Mitchell, M. (1998). An Introduction to Genetic Algorithms. MIT Press.
6. Haykin, S. (1999). Neural Networks: A Comprehensive Foundation. Prentice Hall.
7. Eiben, A. E., & Smith, J. E. (2015). Introduction to Evolutionary Computing. Springer.
8. Yao, X. (1999). "Evolving Artificial Neural Networks". Proceedings of the IEEE, 87(9), 14231447.
9. Stanley, K. O., & Miikkulainen, R. (2002). "Evolving Neural Networks through Augmenting Topologies". Evolutionary Computation, 10(2), 99127.
10. Nurmamatovich, T. I., & Kudratullo o‘g, K. U. B. (2024). THE EVOLUTION OF AI: FROM EARLY CONCEPTS TO MODERN BREAKTHROUGHS. Лучшие интеллектуальные исследования, 20(2), 42-46.