VORISLIK YORDAMIDA SINFLAR IERARXIYASI YARATISH

Mamatxonova Gulasal Saidjon qizi
Farg'ona davlat universiteti talabasi
tulanovagulixon@gmail.com

Tursunaliyeva Mohinur Zoirjon qizi
Farg'ona davlat universiteti talabasi
mohinurtursunaliyeva2907@gmail.com

Yusupov Mirsaidbek
Farg'ona davlat universiteti Amaliy matematika va informatika kafedrasi o'qituvchisi
mirsaidbeky@gmail.com

Annotatsiya: Ushbu maqolada vorislik yordamida sinflar ierarxiyasini yaratishning nazariy va amaliy jihatlari tahlil qilinadi. Ta'lim muassasasidagi shaxslarni modellashtirish misolida, umumiy xususiyatlarga ega bazaviy sinf va undan hosila sinflar yaratish orqali kodning takrorlanishi kamaytirilib, dasturiy ta'minotning kengaytiriluvchanligi oshiriladi. Natijalar vorislik mexanizmi dasturiy tizimlarni samarali tashkil etishda muhim rol o'ynashini ko'rsatadi.

Аннотация: В данной статье анализируются теоретические и практические аспекты создания иерархии классов с использованием наследования. На примере моделирования персонала учебного заведения показано, как создание базового класса с общими свойствами и производных классов снижает дублирование кода и повышает расширяемость программного обеспечения. Результаты подчеркивают важную роль механизма наследования в эффективной организации программных систем.

Abstract: This article examines the theoretical and practical aspects of creating class hierarchies using inheritance. Using the example of modeling personnel in an educational institution, it demonstrates how defining a base class with common

attributes and derived classes reduces code duplication and enhances software extensibility. The findings highlight the important role of inheritance in effectively structuring software systems.

Kalit so‘zlar: Vorislik, sinflar ierarxiyasi, obyektga yo‘naltirilgan dasturlash, kodni qayta foydalanish, polimorfizm, bazaviy va hosila sinflar, dasturiy modellashtirish, dasturiy kengaytiriluvchanlik.

Ключевые слова: Наследование, иерархия классов, объектно-ориентированное программирование, повторное использование кода, полиморфизм, базовый и производные классы, программное моделирование, расширяемость программного обеспечения.

Keywords: Inheritance, class hierarchy, object-oriented programming, code reuse, polymorphism, base and derived classes, software modeling, software extensibility.

KIRISH

Zamonaviy dasturiy ta’minotlarni ishlab chiqishda murakkab tizimlarni mantiqiy jihatdan to‘g‘ri tashkil etish, ularning kengaytiriluvchanligi va qayta foydalanish imkoniyatlarini ta’minlash muhim ahamiyat kasb etadi. Ayniqsa, obyektga yo‘naltirilgan dasturlash paradigmasida dasturiy kodning strukturaviy aniqligi va modulliligi samaradorlikni belgilovchi asosiy omillardan biri hisoblanadi. Ushbu jarayonda vorislik (inheritance) mexanizmi orqali sinflar o‘rtasida ierarxik bog‘lanishlarni yaratish muhim metodologik yondashuv sifatida namoyon bo‘ladi.

Vorislik yordamida sinflar ierarxiyasini yaratish dasturiy ta’minot arxitekturasini soddalashtirish, umumiy xususiyat va funksiyalarni yagona bazaviy sinfda jamlash hamda ularni hosila sinflar orqali kengaytirish imkonini beradi. Bu esa kodning takrorlanishini kamaytirish, uni qo‘llab-quvvatlashni yengillashtirish va tizimning moslashuvchanligini oshirishga xizmat qiladi. Shuningdek, sinflar ierarxiyasi orqali real hayotdagi obyektlar va ularning o‘zaro munosabatlarini dasturiy model ko‘rinishida ifodalash mumkin bo‘lib, bu dasturiy modellashtirishning aniqligi va mantiqiy izchilligini ta’minlaydi.

Bugungi kunda ko‘plab dasturlash tillari, jumladan, Java, C++, Python va C# tillari vorislik mexanizmini qo‘llab-quvvatlaydi va undan samarali foydalanish dasturchidan chuqur nazariy bilim hamda amaliy ko‘nikmalarni talab etadi. Noto‘g‘ri tashkil etilgan sinflar ierarxiyasi esa dastur murakkabligini oshirib, uning kengaytirilishida qiyinchiliklar keltirib chiqarishi mumkin. Shu bois vorislik tamoyillarini to‘g‘ri qo‘llash, “is-a” munosabatlarini aniq belgilash va ierarxik tuzilmani mantiqiy asosda qurish dolzarb masalalardan biri hisoblanadi.

Mazkur maqolada vorislik yordamida sinflar ierarxiyasini yaratishning nazariy asoslari, uning afzalliklari hamda dasturiy tizimlarni loyihalashdagi o‘rni tahlil qilinadi. Shuningdek, sinflar o‘rtasidagi ierarxik munosabatlarni to‘g‘ri tashkil etish orqali dasturiy ta’minot sifatini oshirish masalalariga alohida e’tibor qaratiladi.

Mavzuga oid masala tanlab oldik, u shundan iborat: Ta’lim muassasasida faoliyat yurituvchi shaxslarni dasturiy model orqali ifodalash talab etiladi. Bunda barcha shaxslar uchun umumiy bo‘lgan xususiyatlarni o‘z ichiga olgan **Shaxs** nomli bazaviy sinf yaratiladi. Ushbu sinfdan vorislik orqali **Talaba va O‘qituvchi** sinflari hosil qilinadi. Har bir hosila sinf o‘ziga xos qo‘shimcha atribut va metodlarga ega bo‘lishi lozim. Dastur orqali sinflar ierarxiyasining to‘g‘ri ishlashi va vorislik mexanizmidan samarali foydalanilganligi namoyish etilsin.

ASOSIY QISM:

Obyektga yo‘naltirilgan dasturlashda vorislik tushunchasi sinflar o‘rtasida mantiqiy ierarxik munosabatlarni shakllantirish imkonini beradi. Ushbu yondashuv real hayot obyektlarini umumlashtirish va ularning o‘ziga xos jihatlarini ajratib ko‘rsatishda muhim ahamiyat kasb etadi. Mazkur masalada ta’lim muassasasida uchraydigan shaxslar o‘rtasidagi umumiy va farqli jihatlar vorislik mexanizmi orqali modellashtiriladi.

Avvalo, barcha shaxslar uchun umumiy bo‘lgan ism, yosh kabi atributlarni o‘z ichiga olgan **Shaxs** bazaviy sinfi yaratiladi. Ushbu sinf umumiy metodlar orqali obyekt haqidagi asosiy ma’lumotlarni qaytaradi. Keyinchalik, **Talaba** va **O‘qituvchi** sinflari ushbu bazaviy sinfdan voris olinib, ularga mos ravishda kurs, yo‘nalish yoki fan nomi,

ish staji kabi qo‘shimcha atributlar kiritiladi. Bu esa sinflar ierarxiyasining mantiqiy izchilligini ta’minlaydi.

Vorislikdan foydalanish orqali kodning takrorlanishi kamayadi, chunki umumiy xususiyatlar bir marotaba bazaviy sinfda aniqlanadi. Hosila sinflar esa ushbu xususiyatlardan bevosita foydalanish yoki ularni kengaytirish imkoniyatiga ega bo‘ladi. Natijada dasturiy ta’minotning kengaytiriluvchanligi va qayta foydalanish darajasi sezilarli darajada oshadi. Mazkur yondashuv obyektga yo‘naltirilgan dasturlash tamoyillaridan biri bo‘lgan **DRY (Don’t Repeat Yourself)** prinsipiga ham to‘liq mos keladi.

Masalaning pythonda tuzilgan kodi:

```
# Bazaviy sinf
class Shaxs:
    def __init__(self, ism, yosh):
        self.ism = ism
        self.yosh = yosh
    def info(self):
        return f"Ism: {self.ism}, Yosh: {self.yosh}"
# Hosila sinf - Talaba
class Talaba(Shaxs):
    def __init__(self, ism, yosh, kurs, yonalish):
        super().__init__(ism, yosh)
        self.kurs = kurs
        self.yonalish = yonalish
    def info(self):
        return (f"{super().info()}, "
                f"Kurs: {self.kurs}, Yo'nalish: {self.yonalish}")
# Hosila sinf - O‘qituvchi
class Oqituvchi(Shaxs):
    def __init__(self, ism, yosh, fan, tajriba):
        super().__init__(ism, yosh)
        self.fan = fan
        self.tajriba = tajriba
```

```
    def info(self):
        return (f"{super().info()}, "
                f"Fan: {self.fan}, Tajriba: {self.tajriba} yil")
# Obyektlar yaratish
talaba1 = Talaba("Ali", 20, 2, "Kompyuter injiniringi")
oqituvchi1 = Oqituvchi("Malika", 35, "Dasturlash", 10)
# Natijani chiqarish
print(talaba1.info())
print(oqituvchi1.info())
```

Kodning natijasi quyida:

```
Ism: Ali, Yosh: 20, Kurs: 2, Yo'nalish: Kompyuter injiniringi
Ism: Malika, Yosh: 35, Fan: Dasturlash, Tajriba: 10 yil
```

Mazkur Python dasturida obyektga yo‘naltirilgan dasturlashning muhim tushunchalaridan biri bo‘lgan vorislik mexanizmi orqali sinflar ierarxiyasi yaratilgan. Dastur ta’lim muassasasidagi shaxslarni modellashtirishga qaratilgan bo‘lib, real hayot obyektlari dasturiy muhitda mantiqiy va strukturaviy jihatdan ifodalangan.

Dastlab, barcha shaxslar uchun umumiy bo‘lgan atributlarni o‘z ichiga olgan **Shaxs** nomli bazaviy sinf aniqlangan. Ushbu sinfda shaxsning ismi va yoshi kabi asosiy ma’lumotlar saqlanadi hamda `info()` metodi orqali ushbu ma’lumotlar yagona ko‘rinishda qaytariladi. Bu yondashuv umumiy xususiyatlarni bir joyda jamlash orqali kodning soddaligini va tushunarliligini ta’minlaydi.

Keyingi bosqichda **Talaba** sinfi **Shaxs** sinfidan vorislik orqali yaratilgan. `super()` funksiyasi yordamida bazaviy sinf konstruktori chaqirilib, umumiy atributlar qayta yozmasdan olinadi. Shundan so‘ng, talaba obyektiga xos bo‘lgan kurs va yo‘nalish atributlari qo‘shiladi. `info()` metodining qayta aniqlanishi (method overriding) orqali bazaviy sinf metodi kengaytirilgan va talaba haqidagi to‘liq ma’lumotlarni aks ettirish imkoniyati yaratilgan.

Xuddi shu tarzda, **O‘qituvchi** sinfi ham **Shaxs** sinfidan voris olinib, unga fan nomi va ish tajribasi kabi maxsus atributlar kiritilgan. Ushbu sinfda ham `info()`

metodi qayta aniqlanib, bazaviy sinf ma’lumotlariga qo‘shimcha tarzda o‘qituvchiga tegishli axborotlar chiqariladi. Bu holat polimorfizm tamoyilining amaliy ko‘rinishini namoyon etadi, ya’ni bir xil nomdagi metodlar turli sinflarda turlicha natija qaytaradi.

Dastur yakunida **416alab ava O‘qituvchi** sinflaridan obyektlar yaratiladi va ularning `info()` metodlari chaqiriladi. Natijada, vorislik orqali olingan umumiy va xususiy atributlar birgalikda ekranga chiqarilib, sinflar ierarxiyasining to‘g‘ri ishlashi yaqqol ko‘rinadi. Ushbu yondashuv dasturiy ta’minotning kengaytiriluvchanligini oshiradi va obyektga yo‘naltirilgan dasturlashning asosiy prinsiplarini samarali qo‘llash imkonini beradi.

MUHOKAMA

Mazkur tadqiqotda vorislik mexanizmi yordamida sinflar ierarxiyasini yaratish masalasi ta’lim muassasasi misolida ko‘rib chiqildi. Tanlangan masala real hayotdagi obyektlar o‘rtasidagi mantiqiy munosabatlarni dasturiy muhitda to‘g‘ri modellashtirish imkonini bergani bilan ahamiyatlidir. Shaxs, Talaba va O‘qituvchi sinflari o‘rtasida tashkil etilgan ierarxiya obyektga yo‘naltirilgan dasturlashdagi “umumiydan xususiyga” tamoyilining amaliy ifodasini namoyon etadi.

Muhokama jarayonida shuni ta’kidlash mumkinki, bazaviy sinfda umumiy atribut va metodlarning jamlanishi hosila sinflar strukturasini soddalashtiradi hamda dastur kodining o‘qilishi va tushunilishini yengillashtiradi. Ayniqsa, `info()` metodining turli sinflarda qayta aniqlanishi orqali polimorfizm tamoyilining amalda qo‘llanilishi dasturiy moslashuvchanlikni oshiradi. Bu esa kelgusida tizimga yangi sinflar qo‘shish jarayonini murakkablashtirmasdan amalga oshirish imkonini beradi.

Shuningdek, mazkur yondashuv dasturiy tizimlarni loyihalashda uchraydigan asosiy muammolardan biri — kodning ortiqcha takrorlanishini bartaraf etishga xizmat qiladi. Agar vorislik mexanizmidan foydalanilmaganida, har bir sinfda umumiy atributlar alohida yozilishi talab etilar edi. Bu esa kod hajmining ortishiga va xatoliklar ehtimolining ko‘payishiga olib kelishi mumkin. Demak, muhokama natijalari vorislik asosida yaratilgan sinflar ierarxiyasi dasturiy ta’minot sifatini oshirishda muhim rol o‘ynashini ko‘rsatadi.

NATIJALAR

Olib borilgan tahlil va amaliy misol natijasida vorislik yordamida sinflar ierarxiyasini yaratish obyektga yo‘naltirilgan dasturlashda muhim va samarali yondashuv ekanligi aniqlandi. Python dasturlash tilida ishlab chiqilgan model orqali real hayotdagi shaxslar o‘rtasidagi munosabatlar mantiqiy va strukturaviy jihatdan to‘g‘ri ifodalandi.

Mazkur kodning asosiy foydali jihatlaridan biri shundaki, u dasturiy ta’minotni kengaytirish imkoniyatini ta’minlaydi. Masalan, kelajakda “Xodim”, “Magistrant” yoki “Administrator” kabi yangi sinflarni yaratish zarurati tug‘ilganda, ular mavjud Shaxs bazaviy sinfidan vorislik olish orqali tizimga osonlik bilan qo‘shilishi mumkin. Bu holat dasturiy loyihalarning uzoq muddatli barqarorligini ta’minlaydi.

Shuningdek, vorislik mexanizmidan foydalanish dastur kodining ixchamligi, qayta foydalanish darajasi va texnik xizmat ko‘rsatish qulayligini oshiradi. Dasturchi uchun kodni tahlil qilish va xatoliklarni aniqlash jarayoni soddalashadi, bu esa ishlab chiqish vaqtini qisqartiradi. Natijada, ishlab chiqilgan dasturiy yechim nafaqat o‘quv jarayonida, balki real axborot tizimlarini loyihalashda ham amaliy ahamiyatga ega bo‘ladi.

Xulosa qilib aytganda, vorislik yordamida sinflar ierarxiyasini yaratish dasturiy modellashtirishning muhim vositasi bo‘lib, obyektga yo‘naltirilgan dasturlashning asosiy tamoyillarini samarali qo‘llashga xizmat qiladi hamda yuqori sifatli, moslashuvchan va kengaytiriluvchan dasturiy ta’minot yaratish imkonini beradi.

XULOSA

Ushbu maqolada obyektga yo‘naltirilgan dasturlashning muhim mexanizmlaridan biri bo‘lgan vorislik yordamida sinflar ierarxiyasini yaratish masalasi nazariy va amaliy jihatdan tahlil qilindi. Tadqiqot davomida vorislik mexanizmi orqali sinflar o‘rtasida mantiqiy va izchil bog‘lanishlarni tashkil etish dasturiy modellashtirishning aniqligi hamda samaradorligini oshirishi isbotlandi. Ta’lim muassasasi misolida ishlab chiqilgan model real hayot obyektlarini dasturiy muhitda to‘g‘ri aks ettirish imkonini berdi.

Amaliy misol asosida bazaviy va hosila sinflar o‘rtasidagi munosabatlarni to‘g‘ri tashkil etish kodning strukturaviy soddaligini ta’minlashi, uning qayta foydalanish darajasini oshirishi hamda kengaytiriluvchan dasturiy yechimlar yaratishga zamin hozirlashi aniqlandi. Shuningdek, metodlarni qayta aniqlash orqali polimorfizm tamoyilining qo‘llanilishi dastur moslashuvchanligini yanada kuchaytirishi ko‘rsatib berildi.

Olib borilgan tahlil natijalari shuni ko‘rsatadiki, vorislik asosida qurilgan sinflar ierarxiyasi dasturiy tizimlarni loyihalash jarayonida muhim metodologik ahamiyatga ega bo‘lib, dasturchiga murakkab tizimlarni mantiqiy jihatdan to‘g‘ri va samarali tashkil etish imkonini beradi. Natijada, ushbu yondashuv obyektga yo‘naltirilgan dasturlash prinsiplarini chuqur o‘zlashtirishda, shuningdek, zamonaviy va sifatli dasturiy ta’minot yaratishda muhim ahamiyat kasb etadi.

Foydalanilgan adabiyotlar:

1. *Stroustrup, B. The C++ Programming Language. 4th ed., Addison-Wesley, 2013.*
2. *Lippman, S., Lajoie, J., Moo, B. C++ Primer. 5th ed., Addison-Wesley, 2012.*
3. *Eckel, B. Thinking in Java. 4th ed., Prentice Hall, 2006.*
4. *Lutz, M. Learning Python. 5th ed., O'Reilly Media, 2013.*
5. *Gamma, E., Helm, R., Johnson, R., Vlissides, J. Design Patterns: Elements of Reusable Object-Oriented Software. Addison-Wesley, 1994.*
6. *Meyer, B. Object-Oriented Software Construction. 2nd ed., Prentice Hall, 1997.*
7. *Horstmann, C. S. Object-Oriented Design and Patterns. Wiley, 2004.*
8. *Sebesta, R. W. Concepts of Programming Languages. 11th ed., Pearson, 2015.*
9. *Tursunaliyeva M. Energy efficiency in neural networks: problems of optimizing large models // Science and Innovation. International Scientific Journal. - 2021. - Vol. 4, No. 11. - P. 59-61. - DOI: 10.5281/zenodo.17674536.*
10. *OpenAI. GPT-4 Technical Report. - 2021. - URL: https://openai.com (accessed: 20.11.2025).*