OPYEKTGA YO'NALTIRILGAN DASTURLASH ASOSLARIDA INTERFEYSLAR

Alisher Ismailov Shokirovich
Mahmudova Madina Bahodir qizi
Toshkent Davlat Iqtisodiyot Universiteti
https://doi.org/10.5281/zenodo.17347778

Annotatsiya :

Ushbu mustaqil ishda obyektga yo'naltirilgan dasturlash (OOP) tamoyillari va ularning muhim tarkibiy qismi — interfeyslar haqida so'z yuritiladi. Maqolada interfeyslarning mohiyati, ularning dasturlashdagi vazifasi, sinflar bilan o'zaro farqlari hamda turli dasturlash tillarida qo'llanish usullari yoritilgan. Interfeyslarning dastur arxitekturasida abstraksiya, polimorfizm va modullikni ta'minlashdagi o'rni tahlil qilinadi. Shuningdek, real hayotdagi misollar orqali interfeyslar yordamida standartlashtirilgan, kengaytiriluvchan va ishonchli tizimlar yaratish imkoniyatlari ko'rsatib o'tilgan.

Kalit so'zlar : Interfeys, obyektga yo'naltirilgan dasturlash, abstraksiya, polimorfizm, sinf, metod, merosxo'rlik, inkapsulyatsiya, modullik, dastur arxitekturasi.

Annotation:

This independent work discusses the principles of object-oriented programming (OOP) and their important component - interfaces. The article discusses the essence of interfaces, their role in programming, their differences from classes, and their methods of use in various programming languages. The role of interfaces in providing abstraction, polymorphism, and modularity in program architecture is analyzed. It also shows the possibilities of creating standardized, scalable, and reliable systems using interfaces through real-life examples.

Keywords: Interface, object-oriented programming, abstraction, polymorphism, class, method, inheritance, encapsulation, modularity, program architecture.

Аннотация:

В данной самостоятельной работе рассматриваются принципы объектно-ориентированного программирования (ООП) и их важный компонент – интерфейсы. В статье рассматривается сущность интерфейсов, их роль в программировании, их отличия от классов и способы использования в различных языках программирования. Анализируется роль интерфейсов в обеспечении абстракции, полиморфизма и модульности архитектуры программ. Также на реальных примерах показаны возможности создания стандартизированных, масштабируемых и надёжных систем с использованием интерфейсов.

Ключевые слова: Интерфейс, объектно-ориентированное программирование, абстракция, полиморфизм, класс, метод, наследование, инкапсуляция, модульность, архитектура программы.

Kirish

Obyektga yo'naltirilgan dasturlash (OOD yoki OOP) — bu dasturlash paradigmasi bo'lib, unda dastur obyektlar to'plami sifatida qaraladi. Har bir obyekt ma'lum xususiyatlar (atributlar) va xatti-harakatlar (metodlar) ga ega bo'ladi.

OOP ning asosiy tamoyillari quyidagilardir:

Inkapsulyatsiya (Encapsulation)

Merosxo'rlik (Inheritance)

Polimorfizm (Polymorphism)

Abstraksiya (Abstraction)

Shulardan abstraksiya va polimorfizm tamoyillarini amaliy tarzda qo‘llashda interfeyslar muhim rol o‘ynaydi.

Interfeys tushunchasi

Interfeys (Interface) — bu sinflar bajarishi kerak bo‘lgan funksiyalar to‘plamini aniqlovchi shartnoma (contract) hisoblanadi. Interfeyslar o‘z ichida faqat metod nomlari (va ularning imzolari) ni saqlaydi, lekin ularning ichki bajarilish kodini o‘z ichiga olmaydi. Ya’ni, interfeys faqat “nima qilinishi kerak” degan savolga javob beradi, “qanday amalga oshiriladi” deganiga esa javob bermaydi.

3. Interfeysning asosiy vazifalari

Standartlashtirish:

Interfeyslar dasturda turli sinflar uchun umumiy standart yaratadi.

Masalan, IDrawable interfeysi barcha “chizish” bilan bog‘liq sinflar uchun majburiy metodlarni belgilaydi.

Polimorfizmni ta’minlash:

Turli sinflar bir xil interfeysni amalga oshirishi mumkin, natijada ular bir xil turdagi obyekt sifatida ishlatiladi.

Abstraksiya yaratish:

Foydalanuvchi obyektning faqat tashqi imkoniyatlarini ko‘radi, lekin uning ichki kodini bilmaydi.

Bo‘laklarga ajratish (loose coupling):

Interfeyslar yordamida dastur modullari o‘zaro mustaqil ishlaydi, bu esa kodni oson kengaytirish va testlash imkonini beradi.

4. Interfeys va sinf o‘rtasidagi farqlar

№ Xususiyat Sinf (Class) Interfeys (Interface)

1 Tarkibi Maydonlar va metodlar bo‘lishi mumkin Faqat metod imzolari

2 Meros olish Bitta sinfdan meros oladi Bir nechta interfeysni amalga oshirishi mumkin

3 Maqsad Xatti-harakatni va holatni aniqlaydi Faqat xatti-harakatni belgilaydi

4 Kodni yozish Metodlar bajarilish kodi mavjud Faqat metod nomlari (realizatsiyasiz)

5 Qo‘llanish sohasi Murakkab obyektlarni yaratish Abstrakt standartlar yaratish uchun

5. Interfeyslarning dasturlash tillaridagi ko‘rinishlari

```
interface Animal {
   void sound();
   void move();
}

class Dog implements Animal {
   public void sound() {
      System.out.println("Woof!");
   }
   public void move() {
      System.out.println("Running...");
   }
}
public class Main {
```

```
    public static void main(String[] args) {
        Animal myDog = new Dog();
        myDog.sound();
        myDog.move();
    }
}
```

Bu misolda Animal interfeysi har qanday hayvon uchun “ovoz chiqarish” va “harakatlanish” funksiyalarini belgilaydi.

Dog esa bu interfeysni amalga oshirib, aniq kod yozadi.

🔷 Python misolida (abc moduli yordamida):

```
from abc import ABC, abstractmethod
class Shape(ABC):
    @abstractmethod
    def area(self):
        pass
class Circle(Shape):
    def init(self, r):
        self.r = r
    def area(self):
        return 3.14 * self.r * self.r
circle = Circle(5)
print("Aylana yuzasi:", circle.area())
```

Bu yerda Shape abstrakt sinf interfeys vazifasini bajaradi.

Har bir shakl (Circle, Rectangle, Triangle) area() metodini o‘zicha bajaradi.

```
public interface ILogger {
    void Log(string message);
}
public class FileLogger : ILogger {
public void Log(string message) {
        Console.WriteLine("Faylga yozildi: " + message);
    }
}
```

6. Interfeyslar yordamida polimorfizm

Interfeyslar polimorfizm (ko‘p shakllilik) imkonini beradi.

Masalan, quyidagi kodda turli sinflar bir xil interfeys orqali boshqariladi:

```
List<Animal> animals = new ArrayList<>();
animals.add(new Dog());
animals.add(new Cat());
animals.add(new Bird());
for (Animal a : animals) {
    a.sound();  // har biri o‘z ovozini chiqaradi
}
```

Bu orqali dasturchi turli obyektlar ustida bir xil amallarni bajarishi mumkin.

7. Interfeyslar va abstrakt sinflar farqi

№ Belgilanish Abstrakt sinf Interfeys

1 Tarkibi Metodlar va o‘zgaruvchilar bo‘lishi mumkin Faqat metodlar

2 Meros olish Faqat bitta sinfga ruxsat Bir nechta interfeysni amalga oshirish mumkin

3 Kod yozish Qisman bajarilish kodi bo‘lishi mumkin Faqat metod nomlari

4 Maqsad Umumiy xulq-atvorni aniqlash Standart shartnoma yaratish

8. Interfeysdan foydalanishning afzalliklari

✓ Kodni modulli va qayta ishlatishga yaroqli qiladi

✓ Loyihani kengaytirish osonlashadi

✓ Sinflar o‘rtasidagi bog‘liqlik kamayadi

✓ Testlash jarayoni yengillashadi

✓ Jamoaviy dasturlashda standart va tartibni saqlaydi

9. Real hayotdan misol

Misol: “To‘lov tizimi” loyihasida turli to‘lov usullari mavjud — Payme, Click, Apelsin. Har birining ishlash mexanizmi har xil, lekin umumiy funksiyalar bir xil:

```
interface Payment {
  void connect();
  void pay(double amount);
}
class Payme implements Payment {
  public void connect() { System.out.println("Payme tizimiga ulanmoqda..."); }
  public void pay(double amount) { System.out.println(amount + " so‘m Payme orqali to‘landi."); }
}
class Click implements Payment {
  public void connect() { System.out.println("Click tizimiga ulanmoqda..."); }
  public void pay(double amount) { System.out.println(amount + " so‘m Click orqali to‘landi."); }
}
```

Bu yondashuv orqali yangi to‘lov tizimini (Apelsin, UzPay, va hokazo) qo‘shish uchun faqat interfeysni amalga oshiruvchi yangi sinf yozish kifoya.

Xulosa

Xulosa qilib aytganda, interfeyslar obyektga yo‘naltirilgan dasturlashning eng muhim tarkibiy qismlaridan biridir. Ular dastur modullarini o‘zaro mustaqil, lekin yagona standart asosida ishlashini ta’minlaydi. Interfeyslar yordamida kod tartibli, kengaytiriluvchan va oson boshqariladigan shaklga ega bo‘ladi. Interfeyslarning asosiy afzalligi — ular dasturchiga “nima bajarilishi kerak” degan talabni aniq belgilab, “qanday bajarilishi”ni esa sinflarga topshiradi. Bu yondashuv kodni modulli qilish, testlashni yengillashtirish va yangi funksiyalarni tizimga qulay tarzda qo‘shish imkonini beradi. Zamonaviy dasturlash tillarida (Java, C#, Python, TypeScript va boshqalar) interfeyslar keng qo‘llanilib, yirik dasturiy loyihalarda abstraksiya, polimorfizm va standartlashtirish tamoyillarini samarali amalga oshirish vositasi bo‘lib xizmat qiladi. Shu sababli, interfeyslarni chuqur o‘rganish va ularni to‘g‘ri qo‘llash har bir dasturchining kasbiy mahoratini oshiradi.

Foydalanilgan adabiyotlar:

1. O‘zbekiston Respublikasi Axborot texnologiyalari va kommunikatsiyalarini rivojlantirish vazirligi. “Dasturlash texnologiyalari fanidan o‘quv qo‘llanma” — Toshkent: 2020.
2. Abdullayev A., Karimov B. “Dasturlash asoslari” — Toshkent: TATU nashriyoti, 2021.
3. Usmonov A. R., O‘razov S. “Java dasturlash tili” — Toshkent: Fan va texnologiya, 2020.
4. Xolmirzayev D. A., Mamatqulov M. “C# dasturlash asoslari” — Toshkent: Innovatsion rivojlanish nashriyoti, 2022.
5. Saidov U., Raxmonov B. “Kompyuter dasturlash nazariyasi” — Toshkent: O‘zbekiston Milliy universiteti nashriyoti, 2019.
6. Jo‘rayev B., Yo‘ldoshev M. “Python dasturlash asoslari” — Toshkent: TATU, 2023.
7. Rasulov N. “Algoritmlar va obyektga yo‘naltirilgan dasturlash” — Toshkent: TATU, 2022.