

SINFLARNI TAVSIFLASH USULLARI. SINFLAR TUSHUNCHASI: ASOSIY TUSHUNCHALAR

Halimjonov Sunnatillo

Toshkent Davlat Iqtisodiyot Universiteti

halimjonov2005@icloud.com

Ismoilov Alisher

Toshkent Davlat Iqtisodiyot Universiteti

alisherismoilov1991@gmail.com

Annotatsiya: Ushbu maqolada men “Sinflarni tavsiflash usullari haqida” yoritib beraman. Bu maqolada Sinflarni tavsiflash usullarini asosiy tushunchalari, tamoyillari va ularning dasturlashda qo'llanilishi keltirilgan. C++ dasturlash tilida sinflar (classes) ob'ektga yo'naltirilgan dasturlashning asosiy komponentlaridan biridir. Sinf — ma'lum bir ob'ekt yoki tizimni tavsiflash uchun ishlatiladigan ma'lumotlar va funktsiyalarni o'z ichiga olgan shablondir. Sinf orqali, dasturchilar ob'ektlarni yaratish, ularga xos xususiyatlar va xatti-harakatlarni (methodlar) aniqlash imkoniga ega bo'ladi. Sinf tarkibida, odatda, o'zgaruvchilar (atributlar) va ular bilan ishlovchi metodlar (funktsiyalar) bo'ladi. C++ sinflari ob'ektga yo'naltirilgan dasturlashning uchta asosiy printsiplini — inkapsulyatsiya, meros olish (nasl olish) va polimorfizmni ta'minlaydi. Sinf yaratishda uning atributlari va metodlari xususiy (private) yoki umumiy (public) bo'lishi mumkin. Bu dasturchilarga ma'lumotlar xavfsizligini ta'minlash va kerakli funktsiyalarni samarali boshqarish imkonini beradi. Ushbu maqolada sinflarning yaratish, ularning strukturasi, metodlar va atributlar bilan ishlash usullari haqida batafsil ma'lumot beriladi. Shuningdek, sinflar yordamida kompleks tizimlar yaratish va ularda ob'ektlar bilan ishlash jarayonlarini tushuntirishga harakat qilinadi. Maqola shuningdek ning dasturlash tillaridagi C++ roli va ulardagi qo'llanilishini ko'rib chiqadi. tizimlarining o'zaro bog'liqligi va ularning zamonaviy dasturlashdagi ahamiyati haqida ham so'z yuritiladi..

Kalit so'zlar:obyekt, polymorfizm, abstractsinflar, inkapsulyatsiya, class, dasturlash paradigmalari, meros olish, munozara chiziqlari, konstruktorlar, komponent.

I. Kirish

Obyektga yo'naltirilgan dasturlash (OYD) va loyihalashning markaziy elementlaridan biri bu sinf (class) tushunchasi hisoblanadi. Sinflar yordamida dasturiy ta'minotda murakkab tizimlarni modellashtirish va rejalashtirish amalga oshiriladi. Sinflar va obyektlar tushunchasi dastur tuzish jarayonini modullarga ajratish, ma'lumotlar va funktsionallikni boshqarish hamda tizimni yaxshiroq tuzishda muhim ahamiyat kasb etadi. **Sinf.** Kodning qayta ishlatilishi: Sinflar yordamida kodni qayta ishlatish mumkin. Sinfda bir marta yozilgan xususiyatlar va metodlarni bir nechta ob'ektlar uchun ishlatish imkonini beradi. Bu, ishlab chiqish vaqtini qisqartiradi va kodni samarali boshqarishga yordam beradi. Strukturali tashkil etish: Sinflar dasturdagi kodni tuzish va tartibga solish imkonini beradi, bu esa kodni o'qish va tushunishni osonlashtiradi. Sinflar ob'ektlar o'rtasida o'xshashliklarni aniqlash va boshqarish uchun qulay.

Quyida sinflarni tushuntirish usullari, tavsiflash yo'llari va sinflar bilan bog'liq terminlar keng tushuntiriladi:

1. Sinf Tushunchasi (Class Concept)

- Sinf — obyektga yo'naltirilgan dasturlashda ma'lumotlarni va metodlarni (funksiyalarni) bir joyga jamlagan tuzilma hisoblanadi. Sinf o'z ichida obyektning turli xususiyatlarini (holatlarini) va u bilan bog'liq operatsiyalarni (harakatlarni) aniqlaydi.

- Obyekt — sinfning namunasi bo'lib, u sinfning aniqlangan xususiyatlari va metodlarini saqlaydi. Masalan, 'Avtomobil' sinfi mashinaning turli xususiyatlarini (rangi, turi) va operatsiyalarini (tezlashish, to'xtash) aniqlaydi. Bunda 'BMW' yoki 'Toyota' obyektlari sinfning real namunalari sifatida qaraladi.

Sinf tuzilmasi quyidagilarni o'z ichiga oladi:

1. Xususiyatlar (Attributes): Obyektning holatlarini ifodalaydigan o'zgaruvchilar. Masalan, avtomobil sinfi uchun: rang ('color'), model ('model'), narx ('price') kabi xususiyatlar.
2. Metodlar (Methods): Obyekt bilan bog'liq bo'lgan operatsiyalar yoki funksiya bajaruvchi qismlar. Masalan, avtomobil sinfi uchun: tezlash ('accelerate()'), to'xtash ('stop()') va burilish ('turn()') kabi metodlar.
3. Konstruktorlar va Destruktorlar: Sinf yaratilganda ('constructor') va yo'q qilinganda ('destructor') chaqiriladigan maxsus metodlar.
4. Inkapsulyatsiya va Ma'lumotlarni Yashirish (Encapsulation): Xususiyatlarni tashqi kirishdan himoya qilish va ularga faqat sinfning metodlari orqali kirishni ta'minlash. 'private', 'public', 'protected' kabi kalit so'zlar bilan amalga oshiriladi.

2. Sinf va Obyektlar O'rtasidagi Bog'lanishlar

Sinflar obyektlar uchun shablon hisoblanadi va obyektlar sinfning xususiyat va metodlarini o'zida aks ettiradi. Sinflar va obyektlar o'rtasidagi bog'lanishlar quyidagicha tasvirlanadi:

- Kompozitsiya (Composition): Bir obyekt boshqa obyektning o'z ichiga olishi yoki uning bir qismi sifatida ishlatishi.

- Merosxo'rlik (Inheritance): Yangi sinf yaratish uchun mavjud sinfdan foydalanish.

- Assotsiatsiya (Association): Sinflar o'rtasidagi oddiy bog'lanishlar. Masalan, talabalar va universitet o'rtasidagi bog'lanish.

II. Sinflarni Tavsiflash Usullari

Sinflar dasturlashda quyidagi usullar bilan tavsiflanishi mumkin:

Verbal Tavsiflash (Verbal Description): Sinflarni so'z bilan izohlash. Ushbu usul tizimni loyihalash bosqichida foydalaniladi. Sinfning nomi, vazifasi, xususiyatlari va u bilan bog'liq bo'lgan funksiyalar haqida to'liq izoh beriladi.

UML Sinf Diagrammasi (UML Class Diagram): UML (Unified Modeling Language) yordamida sinflar va ularning bog'lanishlarini diagramma ko'rinishida ifodalaydi. UML sinf diagrammasi quyidagilarni ko'rsatadi:

- Sinfning nomi
- Sinfning atributlari
- Sinfning metodlari
- Sinflar o'rtasidagi bog'lanishlar (Assotsiatsiya, Agregatsiya, Merosxo'rlik)

UML sinf diagrammasi tizimning vizual ko'rinishini yaratib, murakkab bog'lanishlarni aniq tushunishga yordam beradi.

A. Sinfning asosiy tushunchalari

Sinf - bu obyektlarni yaratish uchun shablon yoki tavsifdir. Sinf ob'ektlarning umumiy xususiyatlarini ta'riflaydi va ular asosida konkret ob'ektlar yaratiladi. Sinf ichida atributlar (ob'ektning xususiyatlari) va metodlar (ob'ektga tegishli amallar) belgilangan bo'ladi. Sinf ob'ektlarning xususiyatlari va amallarini umumiy shaklda ta'riflashga imkon beradi, bu esa kodni soddalashtiradi va tizimni boshqarishni osonlashtiradi. Har bir yangi ob'ekt sinfga asoslanib yaratiladi va shu asosda ishlaydi.

Konstruktor (Constructor): Konstruktor — bu sinfga tegishli obyekt yaratishda avtomatik tarzda chaqiriladigan maxsus funksiyadir. U obyektning boshlang'ich holatini

belgilash uchun ishlatiladi. Konstruktor nomi sinf nomiga mos kelishi kerak va qaytish turi bo'lmaydi.

Destructor (Destructor): Destructor — bu obyektни yo'q qilganda chaqiriladigan maxsus funksiyadir. Destructor obyektни xotiradan olib tashlash uchun ishlatiladi. Uning nomi sinf nomi bilan bir xil bo'ladi, lekin oldiga ~ belgi qo'yiladi va qaytish turi bo'lmaydi.

Private, Public va Protected Access Modifiers: C++ sinfida ma'lumotlarga kirishni boshqarish uchun **access modifiers** ishlatiladi:

- public: Bu ma'lumotlarga yoki metodlarga barcha joylardan kirish mumkin.
- private: Faqat sinf ichidagi funksiyalar bu ma'lumotlarga kirishi mumkin.
- protected: Bu ma'lumotlar sinf ichidan va undan meros olgan sinflardan kirish mumkin.

Komponenta o'zgaruvchilar va komponenta funktsiyalar.

Sinf komponenta o'zgaruvchilari sifatida o'zgaruvchilar, massivlar, ko'rsatkichlar ishlatilishi mumkin. Elementlar ta'riflanganda initsializatsiya qilish mumkin emas. Buning sababi shuki sinf uchun hotiradan joy ajratilmaydi. Komponenta elementlariga komponenta funktsiyalar orqali murojat qilinganda faqat nomlari ishlatiladi. Sinfdan tashqarida sinf elementlariga emas ob'ekt elementlariga murojaat qilish mumkin. Bu murojaat ikki hil bo'lishi mumkindir.

Ob'ekt- nomi . Element - nomi.
Ob'ktga – korsatgich – element nomi.

SINFNI TAVSIFLASHNI AVZALLIKLARI:

Inkapsulyatsiya afzalligi. Ma'lumotlarning himoyasi: Inkapsulyatsiya ob'ektning ma'lu motlarini tashqi aralashuvlardan himoya qiladi. Ma'lumotlar faqat metodlar yordamida o'zgartirilishi mumkin, bu esa xatoliklarni kamaytiradi va tizimning xavfsizligini ta'minlaydi. Xatoliklarni kamaytirish: Ma'lumotlar va metodlarning ajratilishi, tizimdagi noaniqlik va noxush holatlarning oldini olishga yordam beradi. Dasturchi faqatgina kerakli metodlar orqali ob'ektga kirish imkoniyatiga ega bo'ladi.

Meros. Kodning qayta ishlatilishi: Meros yordamida mavjud sinflar asosida yangi sinflar yaratish mumkin. Bu, bir xil kodni bir nechta joyda yozishdan saqlaydi va tizimning modifikatsiyasini osonlashtiradi. Oson kengaytirilish: Yangi sinflar eski sinflardan meros qilib olish orqali ularni kengaytirishi va ularning xususiyatlarini o'ziga moslashtirishi mumkin. Bu, dasturda yangi xususiyatlarni qo'shishni osonlashtiradi.

Polimorfizm. Moslashuvchanlik: Polimorfizm bir xil metodni turli ob'ektlar uchun ishlatishga imkon beradi. Bu, bir nechta sinfga mansub ob'ektlar o'rtasida umumiy metodni qayta ishlatish imkoniyatini beradi, bu esa tizimni moslashuvchan va kengaytiriladigan qiladi. Kodning soddalashtirilishi: Turli sinflardan bir xil metoddan foydalanish orqali dasturchilar ko'proq samarali va sodda kod yozadilar. Polimorfizm tizimni yangi talablar asosida kengaytirishni osonlashtiradi.

Abstraksiya. Tizimning soddalashtirilishi: Abstraksiya orqali tizimning faqat kerakli qismlari ko'rsatiladi, keraksiz tafsilotlar esa yashiriladi. Bu foydalanuvchi yoki boshqa dasturchilar uchun tizimni oson tushunishga yordam beradi. Ishlatish qulayligi: Abstraksiya interfeyslarni yaratish imkonini beradi, shunda foydalanuvchi faqatgina zarur bo'lgan xususiyatlar va metodlardan foydalanadi. Bu orqali dastur bilan ishlash qulaylashadi va tizimning ichki ishlashini bilmasdan foydalanish mumkin.

B. Sinf tavsiflashga misollar

Oddiy sinf yaratish

```
#include <iostream>
using namespace std;
class Car {
public:
    string brand;
    string model;
    int year;

    void displayInfo() {
        cout << "Car Brand: " << brand << endl;
        cout << "Car Model: " << model << endl;
        cout << "Manufacture Year: " << year << endl;
    }
};

int main() {
    Car car1;
    car1.brand = "Toyota";
    car1.model = "Corolla";
    car1.year = 2020;

    car1.displayInfo();

    return 0;
}
```

Sinfda konstruktor yaratish

```
#include <iostream>
using namespace std;

class Car {
public:
    string brand;
    string model;
    int year;

    Car(string b, string m, int y) {
        brand = b;
        model = m;
        year = y;
    }

    void displayInfo() {
        cout << "Car Brand: " << brand << endl;
        cout << "Car Model: " << model << endl;
        cout << "Manufacture Year: " << year << endl;
    }
};
```

```
int main() {
    Car car1("Honda", "Civic", 2019);
    car1.displayInfo();
    return 0;
};
```

Sinfda public va private xususiyatlari

```
#include <iostream>
using namespace std;

class BankAccount {
private:
    double balance;

public:
    void deposit(double amount) {
        if (amount > 0) {
            balance += amount;
            cout << "Deposited: $" << amount << endl;
        } else {
            cout << "Invalid deposit amount!" << endl;
        }
    }

    void withdraw(double amount) {
        if (amount <= balance) {
            balance -= amount;
            cout << "Withdrew: $" << amount << endl;
        } else {
            cout << "Insufficient funds!" << endl;
        }
    }

    void checkBalance() {
        cout << "Balance: $" << balance << endl;
    }
};

int main() {
    BankAccount account;
    account.deposit(500);
    account.withdraw(200);
    account.checkBalance();

    return 0;
}
```

III. Xulosa

C++ **dasturlash tilida sinfni tavsiflash** dasturlashning asosiy tushunchalaridan biridir va bir nechta muhim sabablarga ko'ra zarur. Sinfni tavsiflash orqali dasturchilar bir nechta afzalliklarga ega bo'ladilar. C++ dasturlash tilida sinfni tavsiflash juda muhim, chunki bu

ILM FAN YANGILIKLARI KONFERENSIYASI

DEKABR

ANDIJON,2024

modullik, inkapsulyatsiya, meros olish, polimorfizm kabi ob'ektga yo'naltirilgan dasturlashning asosiy tamoyillarini amalga oshirishga imkon beradi. Keyin esa kodlarni Kichik va oson sinflarga ajratish mumkin, asosiysi malumotlarni himoya qiladigani Inkapsulyatsiya, keyin bular yordamida katta dasturlarni tuzish imkoniyatini beradi. Sinflar yordamida xatoliklar va dasturdagi noaniqliklarni kamaytirish mumkin Sinflar yordamida kodni tashkil etish, qayta ishlash, kengaytirish va boshqarish osonlashadi. Dasturchilar sinf yordamida dasturlarini yanada samarali va ixcham tarzda yaratishlari mumkin.

Foydalanilgan adabiyotlar:

1. TATU t.f.d. professor A.X. Nishanov “C++ tilida dasturlash asoslari”
2. “Informatika asoslari” . A.Z.Maxmudov
3. “Kompyuterda dasturlash” Z.Sh. Abdullayeva, X.E. Xujamatov
4. J. Axmadaliyev “C++ dasturlash tili”
5. B.Boltayev “C++ asoslari”