

ILM FAN XABARNOMASI

Ilmiy elektron jurnali

SINF VA OB'EKTLAR, VORISLIK, POLIMORFIZM VA INKAPSULYATSIYA HAQIDA TUSHUNTIRISH

Ziyodillo Tursunov

Toshkent Davlat Iqtisodiyot Universiteti

Alisher Ismailov

Toshkent Davlat Iqtisodiyot Universiteti

Annotatsiya: Bu maqola C++ dasturlash tilida sinf va ob'ektlar, vorislik, polimorfizm va inkapsulyatsiya kabi obyektga yo'naltirilgan dasturlash (OOP) tamoyillariniqamrab oladi. Sinflar va ob'ektlarning asosiy tushunchalari, ularning amaliy dastur yaratishda foydalanilishi to'g'risida yetarlicha ma'lumot beriladi. Shu bilan birga vorislik yordamida kodlarni qayta ishlatish, polimorfizm bilan esa moslashuvchanlikka erishish, hamda inkapsulyatsiya orqali ma'lumot xavfsizligini ta'minlash masalalari maqolada yoritiladi. Misollar yordamida yuqoridagi tushunchalar amaliy ko'rsatib o'tiladi.

Kalit so'zlar: C++, sinf, polimorfizm, OOP, vorislik, inkapsulyatsiya, private, protected, public, dasturlash tillari.

Kirish.

Zamonaviy dasturlash tillarining ko'pchiligi obyektga yo'naltirilgan dasturlash (OOP) paradigmasiga asoslangan. C++ tili OOP tamoyillarini qo'llashda yetakchi hisoblanadi va dasturchilarga murakkab tizimlarni boshqarish imkonini beradi. Ushbu maqola C++ dasturlash tili misolida OOPning asosiy tamoyillari: sinflar va ob'ektlar, vorislik, polimorfizm hamda inkapsulyatsiya haqida tushuncha beradi.

Sinf va ob'ekt tushunchalari dastur tuzilmasining poydevori hisoblanib, real hayotdagi muammolarni modellashtirish uchun qo'llaniladi. Vorislik orqali mavjud kodni qayta ishlatish va dasturiy tizimlar o'rtasidagi bog'liqlikni soddalashtirishga erishiladi. Polimorfizm esa bir xil interfeys orqali turli xil natijalar olishga imkon beradi. Inkapsulyatsiya esa ma'lumotlarni himoya qilish va boshqarishning samarali usulidir.

Bu asosiy tamoyillar dasturlash jarayonini soddalashtiradi, dasturni tezroq ishlab chiqishga imkon beradi va dastur kodini qayta ishlatish samaradorligini oshiradi. Ularni turli xil dasturlash tillarida, shu jumladan C++, JavaScript, Python, Java va boshqalarda qo'llash mumkin. Ular veb-saytlar, mobil ilovlar, stol ilovlari va boshqa turli xil dasturlarni yaratish uchun ishlatilishi mumkin. Ushbu maqolada bu tamoyillarni ta'rifi va amaliy misollar orqali ko'rib chiqamiz.

Sinf va ob'yekt

C++ dasturlash tilida **sinf** ob'yektlarni yaratish uchun reja yoki shablon bo'lib, u atributlar va metodlarni belgilaydi. Sinfda yaratilgan ob'yektlar ushbu atributlar va metodlardan foydalanishi mumkin. Masalan, "Avtomobil" sinfi barcha avtomobillar uchun umumiy bo'lgan xususiyatlar va harakatlar to'plamini ifodalaydi, shundan so'ng turli branddagi, turli yildagi avtomobillar obyekt sifatida yaratiladi.

Ob'yekt – deb sinfni ichidagi funksiya yoki o'zgaruvchini chaqirishi mumkin bo'lgan maxsus o'zgaruvchi. Har bir obyekt o'ziga xos xususiyatlarga va funksiyalarga ega bo'ladi. Obyektning xususiyatlari uning holatini, funksiyalari esa u bajaradigan amallarni ifodalaydi. Har bir obyekt sinfda yaratiladi va sinfning misoli sifatida qaraladi.

Misol:

```
#include <iostream>

using namespace std;

class Avtomobil {
public:
    string brand;
    int yil;
    void displayInfo() {
        cout << "Brandi: " << brand << "Yili: " << yil << endl;
    }
};

int main() {
    Avtomobil obj; // Ob'yekt yaratish

    obj.brand = "Toyota";
    obj.yil = 2020;
    obj.displayInfo();
    return 0;
}
```

Yuqorida keltirilgan misolda “Avtomobil” sinfi yaratilib, uning asosida ob’yekt (obj) yaratildi.

Vorislik

Vorislik ya’ni meros olish yaratilgan bir sinfning ichidagi funksiya va xususiyatlarini boshqa sinfda foydalanish. C++ dasturlash tilida bir sinfdan meros olish orqali, dasturchi o’sha sinf ma’lumotlarini boshqa sinfda qo’llagan holda yangi sinf yaratishi mumkin. Meros olingan sinfni ota class (parent class, base class), undan meros olgan sinfni bola class (child class, derived class) deyiladi.

Meros turlari:

Public meros - a’zolarga dasturdagi barcha sinflar va funksiyalar orqali to’g’ridan-to’g’ri kirish imkonini beradi.

Protected meros – a’zolarga faqat o’sha sinf va uning merosxo’r sinflari tomonidan kirish mumkin.

Private meros – faqat sinfning o’z ichidagi funksiyalar va friend funksiyalar private a’zolarga kirish huquqiga ega. Meros turlari ichida eng xavfsizi ya’ni himoyasi yuqori daajadagisi “private” meros hisoblanadi.

Misol:

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

ILM FAN XABARNOMASI

Ilmiy elektron jurnali

```
class Hayvon {
public:
    void yemoq() {
        cout << " Bu hayvon ovqat yeydi." << endl;
    }
};

class Kuchuk : public Hayvon {
public:
    void hurimoq() {
        cout << "Kuchuk huriydi." << endl;
    }
};

int main() {
    Kuchuk k;
    k.yemoq(); // Ota sinf metodi
    k.hurimoq(); // Voris sinf metodi
    return 0;
}
```

Bu yerda “Kuchuk” sinfi “Hayvon” sinfiga vorislik qilgan va ota classning metodidan foydalangan.

Polimorfizm

C++ da virtual funksiyalar yordamida amalga oshiriladigan va bir xil nomdagi funksiyalar turli xil ob’yektlar uchun turlicha ishlatilishiga polimorfizm deyiladi.

Bu OOPning qulay xususiyatlaridan biri bo‘lib, ko‘p maqsadli kodlarni yaratishda yordam beradi.

Misol:

```
#include <iostream>
using namespace std;
class Shakl {
public:
    virtual void chizish() {
        cout << "Shaklni chizish." << endl;
    }
};

Class doira: public Shakl {
public:
```

```
void chizish() override {
    cout << "Doirani chizish." << endl;
}
};
Class To'rtburchak: public Shakl {
public:
    void chizish() override {
        cout << "To'rtburchakni chizish." << endl;
    }
};
int main() {
    Shakl* sh1 = new Doira();
    Shape* sh2= new To'rtburchak();
    sh1->chizish();
    sh2->chizish();
    delete sh1;
    delete sh2;
    return 0;
}
```

Ushbu kodda virtual funksiyalar orqali bir xil xatti-harakatlar amalga oshirildi.

Inkapsulyatsiya

Inkapsulyatsiya — obyekt ma'lumotlarini va xususiyatlarini tashqi dunyodan yashirish va faqat kerakli qismiga kirish imkonini berish jarayonidir. Inkapsulyatsiya obyektning xavfsizligini oshiradi, chunki ma'lumotlar faqat obyektning o'zidagi metodlar orqali boshqariladi.

C++ dasturlash tilida inkapsulyatsiya uchun “private”, “protected”, va “public” kalit so'zlari ishlatiladi. “private” so'zi bilan belgilangan xususiyatlarga faqat o'sha class ichidagi metodlar orqali kirish mumkin bo'ladi.

Misol:

```
#include <iostream>

using namespace std;

class Hisob {
private:
    double balans;
public:
    Hisob (double boshlangichbalans) {
        balans = boshlangichbalans;
    }
    void depozit(double miqdor) {
        if (miqdor > 0) {
            balans +=miqdor;
        }
    }
};
```


ILM FAN XABARNOMASI

Ilmiy elektron jurnali

```
}  
}  
void qaytmoq(double hisob) {  
    if (hisob > 0 && hisob <= balans) {  
        balan -= hisob;  
    }  
}  
}  
double balansniolish() {  
    return balan;  
}  
};  
int main() {  
    Hisob h(1000);  
    h.depozit(500);  
    h.qaytish(300);  
    cout << "Joriy balans: $" << h.Balansniolish () << endl;  
    return 0;  
}
```

Bu misolda “Balans” o’zgaruvchisi tashqi kirishdan himoyalangan, unga faqat sinf metodlari orqali kirish mumkin

Xulosa

Ushbu maqola yuzasidan xulosa qiladigan bo’sak, maqolada sinf va ob’ektlar, vorislik, polimorfizm va inkapsulyatsiya kabi tamoyillar muhokama qilindi va ularning C++ tilida qo’llanilishi haqida ma’lumot berildi.

Mazkur maqolada ko’rib chiqilgan misollar ushbu tamoyillarni real amaliyotda qanday qo’llash mumkinligini yoritadi. C++ dasturlash tilining kuchli imkoniyatlari tufayli dasturchilar OOP asosidagi dasturlarni tezkor va samarali tarzda ishlab chiqish imkoniyatiga ega bo’lishadi. Shu sababli, C++ ni chuqur o’rganish nafaqat ta’lim, balki amaliy dasturlashda ham muhimdir.

Yuqoridagi tamoyillarning afzalliklari qatorida kodni qayta ishlatilishi, xavfsizlik, moslashuvchanlik, real dunyo muammolarini modellashtirish va katta loyihalar uchun moslashuvchanlikni sanashimiz mumkin.

Tamoyillarning kamchiliklariga keladigan bo’lsak, ular qatoriga murakkablik, resurs talabchanligi, kodning noto’g’ri ishlatilishi, o’rganish qiyinligi, qayta ishlashdagi murakkabliklarni kiritishimiz mumkin.

Lekin, dasturchilar ushbu tamoyillarning kamchilik va afzalliklarini inobatga olib, ularni muvozanatlashtirgan holda qo'llashlari mumkin. Bu esa nafaqat samarali dastur yaratishga yordam beradi, balki kelajakda kodni boshqarish va kengaytirishni ham osonlashtiradi.

Kelajakda dasturlash tillari va uslublari rivojlanishda davom etgan taqdirda ham, obyektga yo'naltirilgan dasturlash tamoyillari dasturchilar uchun asosiy qoidalardan biri bo'lib qolaveradi. Bu tamoyillar, ayniqsa, katta hajmdagi va murakkab tizimlarni ishlab chiqishda o'z ahamiyatini saqlab qoladi.

Foydalanilgan adabiyotlar:

1. Abdurakhmonova, N., Alisher, I., & Toirova, G. (2022, September). Applying Web Crawler Technologies for Compiling Parallel Corpora as one Stage of Natural Language Processing. In 2022 7th International Conference on Computer Science and Engineering (UBMK) (pp. 73-75). IEEE.

2. Abdurakhmonova, N., Alisher, I., & Sayfulleyeva, R. (2022, September). MorphUz: Morphological Analyzer for the Uzbek Language. In 2022 7th International Conference on Computer Science and Engineering (UBMK) (pp. 61-66). IEEE.

3. NAZIROVA, E., ABDURAKHMONOVA, N., & ALISHER, I. Exploring Linguistic Roots (stem) and Word Categories in Uzbek Language through Advanced Natural Language Processing Techniques for Text Analysis.

4. Ismailov, A. S., Akbarov, A., Qodirova, G. T. X. Q., & Yigitaliyeva, M. (2023). Dasturiy ta'minotni ishlab chiqish bosqichlari. *Science and Education*, 4(3), 187-191.

5. Ismailov, A. S. "Obyektga yo'naltirilgan dasturlash va loyihalash". O'quv qo'llanma.

6. Tursunov, Z., To'ychiyev, B., & SH, I. A. (2024, October). KOMPONENTA TUSHUNCHASI VA XUSUSIYATLARI. In INTERNATIONAL CONFERENCE ON ADVANCE SCIENCE AND TECHNOLOGY (Vol. 1, No. 10, pp. 39-45).

7. Ne'matov, D., Sodiqov, Q., & Ismoilov, A. (2024, October). STANDART ALGORITMLAR. In INTERNATIONAL CONFERENCE ON ADVANCE SCIENCE AND TECHNOLOGY (Vol. 1, No. 10, pp. 54-58).

8. Abdumutaliyev, A., Nemadjonov, F., & Ismailov, A. (2024, October). KONTEYNERLAR BILAN ISHLAYDIGAN ALGORITMLAR. In INTERNATIONAL CONFERENCE ON ADVANCE SCIENCE AND TECHNOLOGY (Vol. 1, No. 10, pp. 32-38).

9. Umarjanov, S., Choriyev, M., & Ismailov, A. (2024, October). STL KUTUBXONALARI. In INTERNATIONAL CONFERENCE ON ADVANCE SCIENCE AND TECHNOLOGY (Vol. 1, No. 10, pp. 46-53).