**№2-ma’ruza. Dasturlash tilida klasslar.** Do‘stona funksiyalar. Istisno holatlarni qayta ishlash. Xotirani ajratish va taqsimlash. Inkapsulyatsiya. Meroslik. Polimorfizm.

**Reja:**

1. Sinf va obyekt tushunchalari.
2. Do‘stona funksiyalar.
3. Xotirani ajratish va taqsimlash
4. Inkapsulyatsiya
5. Meroslik. Polimorfizm

***Kalit so‘zlari:****obyektga yo‘naltirilgan dasturlash, sinf, funksiya, shablon, tashqi va ichki interfeys, inkapsulyatsiya, konstruktor, destruktor*

**1**. **Sinflar va obyekt tushunchalari.**

***Obyektga yo‘naltirilgan dasturlash (OYD)*** – dasturning har biri ma'lum bir sinfning namunasi bo‘lgan avtonom harakat qiluvchi obyektlar to‘plami sifatida namoyish etishga asoslangan dasturiy ta’minot yaratish texnologiyasi.

***Sinf***– yaratilmagan obyektning shabloni. Sinfning barcha ma’lumotlari uning maydonlarida saqlanadi. Sinf maydoni – sinf yaratilayotgan vaqtida dasturchi tomonidan tavsiflanadigan o‘zgaruvchi. Sinfda tatbiq etiladigan funksiya uning metodi deb ataladi.

***Sinfni e'lon qilish***. Sinfni e'lon qilish uchun class kalitli so‘zi, undan so‘ng [ochiluvchi figurali qavs](https://fayllar.org/intermediate-control-qavs-ichidagi-fellarni-togri-shaklda-qavs.html), co‘ng xossalar va metodlari ro‘yxati ishlatiladi. Sinfni e'lon qilish yopiluvchi figurali qavs va nuqtali vergul orqali yakunlanadi. Masalan, Mushuk sinfini quyidagicha e'lon qilish mumkin.

*Class Mushuk {*

*unsigned int itsYosh ;*

*unsigned int itsOgirlik ;*

*void Miyovlash()*

*}*

Mushuk sinfini e'lon qilishda xotira zaqiralanmaydi. E'lon qilish, kompilyatorga Mushuk sinfini mavjudligini, qamda unda qanday qiymatlar saqlashi mumkinligi (itsYosh, itsOgirlik) va u qanday amallarni bajarishi mumkinligi (Miyovlash() metodi) haqida xabar beradi. Bundan tashqari, bu e'lon qilish orqali kompilyatorga Mushuk sinfining o‘lchami, ya'ni qar bir Mushuk sinfi obyekti uchun kompilyator qancha joy ajratishi lozimligi qaqida qam ma'lumot beradi. Masalan, joriy misolda butun qiymat uchun to‘rt bayt talab qilinsa, Mushuk sinfi obyekti o‘lchovi sakkiz bayt bo‘ladi. (itsYosh o‘zgaruvchisi uchun to‘rt bayt, itsOgirlik o‘zgaruvchisi uchun to‘rt bayt). Miyovlash() metodi xotiradan joy ajratishni talab qilmaydi.

***Obyektni e'lon qilish.*** Yangi turdagi obyekt xuddi oddiy butun sonli o‘zgaruvchidek aniqlanadi. qaqiqatan qam ixtiyoriy butun sonli o‘zgaruvchi quyidagicha aniqlanadi:

*unsigned int MyVariable // ishorasiz butun sonni aniklaymiz*

Mushuk sinfidagi obyekt esa quyidagicha aniqlanadi:

*Mushuk Frisky // Mushuk obyektini aniklaymiz.*

Bu dasturiy kodlarda unsigned int tipidagi MyVariable nomli o‘zgaruvchi va Mushuk sinfining Frisky nomli obyekti aniqlandi.

Ko‘pgina qollarda sinf va obyekt tushunchalarini ishlatishda chalkashlikka yo‘l qo‘yiladi. Shuning uchun, obyekt sinfning biror bir ekzemplyari (nusxasi) ekanligini yana bir bor ta'kidlash joiz.

Sinf a'zolariga murojaat qilish imkoni. Mushuk sinfining real obyektini aniqlaganimizdan so‘ng bu obyektning a'zolariga murojaat qilish zaruriyati tuqilishi mumkin. Buning uchun bevosita murojaat (.) operatori qo‘llaniladi. Masalan, Frisky obyektining Weight o‘zgaruvchi - a'zosiga 50 sonini o‘zlashtirmoqchi bo‘lsak quyidagi jumlani yozishimiz lozim.

*Fresky. Weightq50;*

*Meow() metodini chaqirish uchun esa*

*Frisky. Meow();*

jumlasini yozish lozim.

qiymat sinfga emas obyektga o‘zlashtiriladi. C++ tilida berilganlar tipiga qiymat o‘zlashtirilmaydi. qiymat faqatgina o‘zgaruvchilarga beriladi. Masalan, quyidagi yozuv noto‘qridir:

*intqs // noto‘g‘ri*

Kompilyator int tipiga qiymat o‘zlashtirilishi xatolik ekanligi qaqida xabar beradi. Xuddi shu nuqtai – nazardan quyidagi yozuv qam noo‘rindir:

*Cat. itsYoshq 5 // noto‘gri*

Agarda Mushuk obyekt bo‘lmasdan sinf bo‘lsa, yuqoridagi ifodani qam kompilyator xato deb qisoblaydi. O‘zlashtirish amalini bajarishda xatolikka yo‘l qo‘ymaslik uchun oldin Mushuk sinfiga tegishli Frisky obyektini qosil qilish va uning ItsYosh maydoniga 5 qiymatini berish lozim.

*Mushuk Frisky;*

*Frisky. itsYoshq5;*

Sinf a'zolariga murojaat qilish imkonini chegaralash. Sinfni e'lon qilishda bir nechta kalit so‘zlardan foydalaniladi. Ulardan eng muqimlari publis (ochiq) va private (yopiq) kalit so‘zlari bo‘lib, ular orqali obyektning a'zolariga murojaat qilish imkoniyati chegaralanadi.

Sinfning barcha metodlari va xossalari boshlanqich qolda yopiq deb e'lon qilinadi. Yopiq a'zolarga faqatgina shu sinfning metodlari orqaligina murojaat qilish mumkin. Obyektning ochiq a'zolariga esa dasturdagi barcha funksiyalar murojaat qilishlari mumkin. Sinf a'zolariga murojaat qilish imkonini belgilash juda muqim xususiyat bo‘lib, bu masalani echishda uncha katta tajribaga ega bo‘lmagan dasturlarchilar ko‘pincha qiyinchiliklarga duch keladilar. Bu qolatni batafsilroq tushuntirish uchun mavzuni boshida keltirilgan masalamizga qaytamiz.

*Class Mushuk {*

*unsigned int itsYosh;*

*unsigned int itsOgirlik;*

*void Miyovlash();*

*}*

Bu tarzda sinfni e'lon qilishda itsYosh va itsOgirlik maydonlari qam, Miyovlash() metodi qam yopiq a'zo sifatida aniqlanadi. Dasturda yuqoridagi tartibda Mushuk sinfi e'lon qilingan bo‘lsa va bu sinf ekzemplyari bo‘lgan obyektning itsYosh a'zosiga main() funksiyasi tanasidan turib murojaat qilsak kompilyator xatolik ro‘y berganligi qaqida xabar beradi.

*Mushuk Baroq;*

*Baroq. itsYosh q 5 // Xatolik! // Yopik a'zoga murojaat kilish mumkin emas*.

Mushuk sinfi a'zolariga dasturning boshqa obyektlari tomonidan murojaat qilish imkonini qosil qilmoqchi bo‘lsak, uni public kalitli so‘zi orqali amalga oshiramiz.

*Class Mushuk {*

*public:*

*unsigned int itsYosh;*

*unsigned int itsOgirlik;*

*void Meow( );*

*}*

Endi public kalitli so‘zi orqali sinfning barcha a'zolari (itsYosh, itsOgirlik, Miyovlash()) ochiq a'zo bo‘ldi.

1. **Do‘stona funksiyalar**

***Friend funksiyalar.*** Klassning *private* va *protected* qismiga klassga tegishli bo‘lmagan *friend* funksiya murojat qilishi mumkin. Friend funksiyalar klasning ichida friend kalit so‘zi bilan yoziladi.

E’lon qilinishi:

*class myclass {*

*…..*

*friend int sum(myclass x);*

*…. };*

Albatta friend funksiyalar sinfdan tashqarida mavjud bo‘ladi va ushbu do‘stona funksiya sinfning barcha sohalariga murojaat qila olishi mumkin.

*class sm{*

*int a, b;*

*public:*

*friend int sum(myclass x);*

*void set\_ab(int i, int j) { a = I; b = j; }*

*};*

*int sum(myclass x) {*

*return x.a + x.b; //sum() hech qaysi classga tegishli emas.*

*}*

*int main() {*

*myclass n;*

*n.set\_ab(3, 4);*

*cout << sum(n);*

*return 0; }*

***Do‘stona (friend) sinflar***. Bir class boshqa bir class’ga do‘stona bo‘lishi mumkin. Bunda sinflar bir – birining a’zolaridan foydalanish imkoniyatiga ega bo‘ladi. Bunda shu narsaga e’tibor berish lozimki, biror sinfga do‘stona bo‘ladigan sinf (ya’ni friend kalit so‘zi orqali e’lon qilinadigan sinf), mazkur sinfning a’zolaridan foydalanish imkoniyatini yaratadi

***E’lon qilinishi:***

*class myclass {*

*…..*

*friend someclass b;*

*….*

*};*

Do‘stona sinfdan foydalanish uchun quyida misol keltirilgan. Bunda e’tibor berishimiz lozimki, TwoValues sinfi Min sinfiga do‘stona bo‘lib, bunda Min sinfi TwoValues sinfining a’zolaridan foydalanishi mumkin.

*class TwoValues {*

*int a, b;*

*public:*

*TwoValues(int i, int j) { a = i; b = j; }*

*friend class Min;};*

*class Min {*

*public:*

*int min(TwoValues x) { return x.a < x.b ? x.a : x.b; }*

*};*

*int main() {*

*TwoValues ob(10, 20);*

*Min m;*

*cout << m.min(ob);*

*return 0;}*

**4. Inkapsulyatsiya**

• ***Ichki*** ***interfeys***– bu faqat ushbu obyektning boshqa usullaridan foydalanish mumkin bo‘lgan xossalar va usullari, ular "xususiy" deb ham nomlanadi.

• ***Tashqi interfeys*** – bu obyekt tashqarisidan kirish mumkin bo‘lgan xossalar va usullar, ular "ommaviy" deb nomlanadi.

• ***Inkapsulyatsiya***– bu sinfda ishlatiladigan ma'lumotlar maydonlari va usullarini birlashtirishga va foydalanuvchidan ichiki interfeys detallarini yashirishga imkon beruvchi dasturlash mexanizmi.

• Inkapsulyatsiya jarayonini amalga oshirish uchun quyidagicha kirish xususiyatlari mavjud:

***public***– sinfning ochiq (oshkor) bo‘limi. Bu bo‘limdagi maydonlar va metodlarga sinfning ichki va bir vaqtning o‘zida tashqi maydon va metod sifatida murojaat qilish mumkin. Ushbu bo‘lim "tashqi interfeys" ni tashkil qiladi;

***private*** – sinfning yopiq bo‘limi. Bu bo‘limdagi metod va maydonlarga faqat sinf ichida murojaat qilish mumkin. Bu bo‘lim “ichki interfeys”ni tashkil qiladi. Jimlik holatida sinfning barcha maydonlari – private;

***protected*** – sinfning maxsus o‘lchami. Bunda maydon va metodlarga sinf ichida murojaat qilish mumkin, shuningdek, merosxo‘r sifatida “qism” sinfda murojaat o‘rnatish mumkin bo‘ladi.

Sinf yaratish va uning maydonlariga murojaat qilishga doir misol:

***1-dastur matni:***

*#include <iostream>*

*using namespace std;*

*class Point // class kalit so‘zi yordamida sinfni yaratish {*

*private: //kirish xususiyati*

*int x; //sinf maydoni*

*int y; // sinf maydoni*

*public: // kirish xususiyati*

*void Set\_Point(int x1, int y1) //nuqta koordinatasi sinfining metodi*

*{*

*x = x1;*

*y = y1;*

*}*

void Show\_Point() //ekranga nuqta koordinatasini chiqarish metodi

*{*

*cout << x << ' ' << y;*

*}*

*};*

*int main()*

*{*

***Point Example;****// Point sinfining obyektini yaratish*

***Example.*** *Set\_Point****(****10****,****12****);****// Example obyekti uchun Set\_Point metodini qo‘llash*

***Example.*** *Show\_Point****();****// Example obyekti uchun Show\_Point metodini qo‘llash*

***}***

**5**. **Meroslik. Polimorfizm**

***Merosxo‘rlik*** – OYD asosiy printsiplaridan biri. Bunda, sinf boshqa sinflarning o‘zgaruvchilari va metodlaridan o‘zining o‘zgaruvchisi va metodi sifatida foydalanishi mumkin.

• Ma’lumotlarni meros qilib oluvchi sinf qismsinf (subclass), hosila sinf (derived class) yoki “bola” sinf (child) deb ataladi. Ushbu o‘zgaruvchilar yoki metodlarni meros qilib qoldiruvchi sinf supersinf (super class), asosiy sinf (base class) yoki “ota” sinf (parent) deb yuritiladi. “Ota” va “bola” atamalari merosxo‘rlikni tushunib olish uchun juda qulay. Bola ota-onasidan harakterlarini [meros qilib olgani kabi](https://hozir.org/ishtirok-etgan-ota-obektlarni-xarakteristika-va-huquqlarni-mer.html), hosila sinflar ham metod va o‘zgaruvchilarni asosiy sinfdan meros qilib oladi [1].

• Merosxo‘rlik euda qulay, chunki dasturlash jarayonini ancha tezlashtirish uchun dastur kodini tuzish va undan qayta foydalanish imkonini beradi. Biroq, merosdan ehtiyotkorlik bilan foydalanish kerak, chunki supersinfdagi ko‘pgina o‘zgarishlar barcha subsinflarga ta'sir qiladi, bu esa kutilmagan oqibatlarga olib kelishi mumkin [1].

• Alohida ta’kidlash kerakki, yopiq (private) o‘zgaruvchilar va metodlardan meros qilish mumkin emas.

***1. Merosxo‘rlikning***[***turlari***](https://hozir.org/mavzu-qiziqarli-va-noyob-sport-turlari.html)

• C ++ dasturlash tilida bir nechta merosxo‘rlik turlari mavjud:

Ochiq (public) – ochiq (public) va himoyalangan (protected) ma’lumotlarni ularga kirishni o‘zgartirmasdan meros qilinadi;

Himoyalangan (protected) – barcha meros qilinadigan ma’lumotlar himoyalangan (protected) bo‘ladi;

Yopiq (private) – barcha meros qilinadigan ma’lumotlar (private) yopiq bo‘ladi.

Masalan:

*class Parent*

*{*

*public:*

*int m\_public; // bu a’zoga barcha obyektlar uchun kirish ochiq*

*private:*

    int m\_private; // bu a’zoga faqat Parent sinfining boshqa a’zolari uchun va do‘stona sinf/funksiyalar uchun kirish ochiq (“bola” sinflar uchun emas)

*protected*:

*int m\_protected;*  *};*

*class Child: public Parent {*

*public:*

*Child()    {*

*m\_public = 1; //ruxsat berilgan: bola sinfidan ota sinfining ochiq a’zolariga kirish*

*m\_private = 2; // taqiqlangan: bola sinfidan ota sinfining yopiq a’zolariga kirish*

*m\_protected = 3; // ruxsat berilgan: bola sinfidan ota sinfining himoyalangan a’zolariga kirish*

*}};*

*int main(){*

[*Parent parent*](https://hozir.org/the-east-orange-school-district-focus-parent-portal-what-is-th.html)*;*

*parent.m\_public = 1; // ruxsat berilgan: sinfning ochiq a'zolariga*[*tashqaridan kirish*](https://hozir.org/reja-kirish-hujayra-patologiyasi-hujayra-zararlanishining-saba.html)

*parent.m\_private = 2; // taqiqlangan: sinfning yopiq a'zolariga tashqaridan kirish*

*parent.m\_protected = 3; // taqiqlangan: sinfning himoyalangan a'zolariga tashqaridan kirish}*

**Nazorat savollari**

1. nxn matritsaning yuqori chap uchburchagidagi elementlaridan vector hosil qiling?
2. nxn matritsaning yuqori o‘ng uchburchagidagi elementlaridan vector hosil qiling?
3. nxn matritsaning pastki o‘ng uchburchagidagi elementlaridan vector hosil qiling?
4. nxn matritsaning pastki chap uchburchagidagi elementlaridan vector hosil qiling?
5. Matritsani matritsaga ko‘paytiring?
6. Jadval hosil qiling va unga ma’lumotlarni kiriting, ekranga chiqaring?
7. Talabalar ism-familiyasi, yoshi va ballaridan iborat jadval yarating va talabalarni ism-familiyasini alfavit bo‘yicha tartibga keltiring.