14

O‘N TO‘RTINCHI MA’RUZA

Dasturlash 1 · C++ · 9-mavzu

Obyektga yo‘naltirilgan dasturlash asoslari

Sinf va obyekt · inkapsulyatsiya · konstruktor va destruktor · this, static, const · merosxo‘rlik · polimorfizm

class | private | Konstruktor | virtual

ISINISH

Telefon qanday “obyekt”: u nimani biladi va nima qila oladi?

Har bir narsaning xususiyatlari (holati) va amallari (xatti-harakati) bor.

1

Telefonning 3 ta xususiyatini ayting (rangi, ...).

2

Telefon bajaradigan 3 ta amalni ayting (qo‘ng‘iroq qilish, ...).

3

Batareya foizini qo‘lda “100%” deb yozib qo‘yish mumkinmi? Nega?

O‘QUV MAQSADLARI

Bu ma’ruza oxirida siz quyidagilarni bajara olasiz...

1. **Tushuntirish** — OYD ning to‘rt tamoyilini: inkapsulyatsiya, merosxo‘rlik, polimorfizm, abstraksiya
2. **Farqlash** — sinf (qolip) va obyekt (nusxa), maydon va metodni
3. **Qo‘llash** — public, private, protected kirish spetsifikatorlarini
4. **Yozish** — getter/setter va tekshiruvchi metodlarni
5. **Yaratish** — konstruktorlar (initsializatsiya ro‘yxati bilan) va destruktorni
6. **Ishlatish** — this ko‘rsatkichi, static a’zolar va const metodlarni
7. **Qurish** — asosiy va hosila sinflar ierarxiyasini
8. **Namoyish qilish** — virtual funksiyalar va abstrakt sinflar orqali polimorfizmni

01

Sinf va obyekt

OYD tamoyillari · class · maydon va metod · . orqali murojaat · struct

class | Obyekt | Maydon | Metod

OYD TAMOYILLARI

Ma’lumot va amallar — bitta obyektda

1 Inkapsulyatsiya

Ma’lumotlarni yashirish va ularga faqat metodlar orqali kirish.

2 Merosxo‘rlik

Mavjud sinf asosida yangi sinf yaratish va qayta ishlatish.

3 Polimorfizm

Bir xil nomli amal turli obyektlarda turlicha bajariladi.

4 Abstraksiya

Muhim xususiyatlarni ajratib, keraksiz tafsilotlarni yashirish.

Protsedurali uslubda ma’lumotlar va funksiyalar alohida; OYD da ular obyektga birlashtiriladi — xuddi real hayotdagi narsalar kabi.

ATAMALAR

Sinf — chizma, obyekt — qurilgan uy

Tushuncha	Ma’nosi	O‘xshatish
Sinf (class)	Foydalanuvchi aniqlagan yangi tur	Uyning chizmasi
Obyekt	Sinfning xotiradagi aniq nusxasi	Chizma asosida qurilgan uy
Maydon	Obyekt holatini saqlaydi	Rangi, qavatlari soni
Metod	Obyekt bajaradigan amal	Eshikni ochish

`struct` va `class` farqi: struct da a’zolar standart bo‘yicha public, class da — private.

Talaba sinfi va ikki obyekt

```
class Talaba {
public:
    string ism;
    double ball;
    bool alochimi() { return ball >= 86; }
    void ballQosh(double x) { ball += x; }
};
int main() {
    Talaba t1;
    t1.ism = "Aziza"; t1.ball = 91;
    Talaba t2 = {"Bekzod", 78};
    t2.ballQosh(10);
    cout << t1.alochimi() << " " << t2.ball << " "
         << t2.alochimi();
}
```

▶ NATIJA

```
1 88 1
```

`class Talaba { ... };`

oxiridagi `;` majburiy!

`t1.ism`

nuqta (.) orqali a’zoga murojaat

`t2.ballQosh(10)`

faqat t2 o‘zgaradi: 78 → 88

TUSHUNISHNI TEKSHIRISH

O‘ylang – Juftlikda – Bo‘lishing · 3 daqiqa

1-bo‘lim yakuni: Sinf va obyekt

1

Sinf va obyekt o‘rtasidagi munosabat qanday?

Savol 1

2

Maydon va metod farqi nima?

Savol 2

3

class va struct ning yagona farqi?

Savol 3

AMALIY TOPSHIRIQ

Har kim o‘z qurilmasida · 10 daqiqa · Oson · 10 ball

1-topshiriq: To‘rtburchak sinfi

`Tortburchak` sinfini yozing: `en` va `boy` maydonlari, `yuza()` va `perimetr()` metodlari. `main()` tayyor.

Kirish: Ikki haqiqiy son — eni va bo‘yi ($0 < \text{son} \le 10000$).

Chiqish: Ikki qator: yuza va perimetr, verguldan keyin 2 xona.

MISOL · KIRISH	MISOL · CHIQISH
`3 4`	`12.00` `14.00`

Telefon kamerasi bilan skanerlang

1. Saytga kiring (login va parol)
2. Kodni yozing va sinab ko‘ring
3. ✓ Topshirish — natija o‘qituvchiga

02

Inkapsulyatsiya

public · private · protected · getter va setter · tekshiruv

private | public | getter | setter

Kim nimani ko‘ra oladi?

Spetsifikator	Sinf ichida	Hosila sinfda	Tashqarida
`public`	✓	✓	✓
`protected`	✓	✓	✗
`private`	✓	✗	✗

Odatda maydonlar `private`, ularga kirish — `public` getter (`getBalans()`) va setter (`setYosh()`) metodlari orqali.

BANK HISOBI

Balansni faqat metodlar o‘zgartiradi

```
class BankHisob {
    double balans = 0;
public:
    bool qoyish(double s) {
        if (s <= 0) return false;
        balans += s; return true;
    }
    bool yechish(double s) {
        if (s <= 0 || s > balans) return false;
        balans -= s; return true;
    }
    double getBalans() const { return balans; }
};
int main() {
    BankHisob h;
    h.qoyish(1000);
    cout << h.yechish(2000) << " " << h.yechish(300) << " "
            << h.getBalans();
}
```

▶ NATIJA

```
0 1 700
```

`double balans`

class da standart bo‘yicha private

`yechish(2000)`

mablag‘ yetarli emas — rad etildi (0)

`h.balans = 1e6;`

XATO: 'balans' is private

Noto‘g‘ri qiymatni rad etish

```
class Vaqt {
    int soat = 0, daqiqa = 0;
public:
    bool setSoat(int s) {
        if (s < 0 || s > 23) return false;
        soat = s; return true;
    }
    void qosh(int m) {
        int j = (soat * 60 + daqiqa + m) % 1440;
        soat = j / 60; daqiqa = j % 60;
    }
    void chop() const { cout << soat << ":" << daqiqa << endl; }
};
int main() {
    Vaqt v;
    cout << v.setSoat(25) << endl;
    v.setSoat(23); v.qosh(90); v.chop();
}
```

▶ NATIJA

```
0
0:30
```

`setSoat(25)`

25 — noto‘g‘ri, qabul qilinmadi (false)

`qosh(90)`

23:00 + 90 daqiqa = 00:30

`% 1440`

sutkadagi daqiqalar — yarim tundan keyin qayta boshlanadi

TUSHUNISHNI TEKSHIRISH

O‘ylang – Juftlikda – Bo‘lishing · 3 daqiqa

2-bo‘lim yakuni: Inkapsulyatsiya

1

class ichida spetsifikator ko‘rsatilmagan a’zolar qanday bo‘ladi?

Savol 1

2

protected a’zo qayerdan ko‘rinadi?

Savol 2

3

Setter metodning asosiy vazifasi nima?

Savol 3

AMALIY TOPSHIRIQ

Har kim o‘z qurilmasida · 10 daqiqa · O‘rta · 20 ball

2-topshiriq: Bank hisobi (inkapsulyatsiya)

`Hisob` sinfini yozing: `balans` maydoni **yopiq** (`private`). Metodlar: `qoyish(x)`, `bool yechish(x)` — mablag‘ yetarli bo‘lmasa yoki `x ≤ 0` bo‘lsa yechmaydi va `false` qaytaradi, `long long qoldiq() const`. `qoyish` ham `x ≤ 0` ni rad etadi.

Kirish: Birinchi qatorda `q` ($1 \le q \le 1000$). Keyin `q` ta buyruq: `+ x` (qo‘yish), `- x` (yechish) yoki `?` (qoldiq). $|x| \le 10^9$.

Chiqish: Har bir `?` uchun qoldiq; muvaffaqiyatsiz `-` uchun `RAD ETILDI`.

MISOL · KIRISH

```
5
+ 100
- 30
?
- 100
?
```

MISOL · CHIQISH

```
70
RAD ETILDI
70
```

Telefon kamerasi bilan skanerlang

1. Saytga kiring (login va parol)
2. Kodni yozing va sinab ko‘ring
3. ✓ Topshirish — natija o‘qituvchiga

03

Konstruktor va destruktor

Standart va parametrli konstruktor · initsializatsiya ro‘yxati · ~Destruktor · hayot sikli

Kasr()

: surat(0)

~Kasr()

Hayot sikli

Obyekt tug‘ilganda avtomatik chaqiriladi

Konstruktor — nomi sinf nomi bilan bir xil, hech qanday tur qaytarmaydigan maxsus metod. U maydonlarga boshlang‘ich qiymat beradi.

Qayta yuklanadi

`Kasr()`, `Kasr(int b)`, `Kasr(int s, int m)` — bir nechta konstruktor.

Initsializatsiya ro‘yxati

`Kasr() : surat(0), maxraj(1) { }` — tavsiya etilgan usul.

Destruktor

`~Kasr()` — obyekt yo‘q qilinganda; parametrsiz, har sinfda bitta.

Birorta konstruktor yozilmasa, kompilyator standart konstruktorni o‘zi yaratadi.

Bir nechta konstruktor

```
class Kasr {
    int s, m;
public:
    Kasr() : s(0), m(1) {}
    Kasr(int b) : s(b), m(1) {}
    Kasr(int s_, int m_) : s(s_), m(m_ == 0 ? 1 : m_) {}
    void chop() const { cout << s << "/" << m << " "; }
};
int main() {
    Kasr a;
    Kasr b(3);
    Kasr c(6, 8);
    Kasr d(1, 0);
    a.chop(); b.chop(); c.chop(); d.chop();
}
```

▶ NATIJA

```
0/1 3/1 6/8 1/1
```

`Kasr a;`

standart konstruktor — 0/1

`Kasr b(3);`

butun son — 3/1

`m_ == 0 ? 1 : m_`

maxraj 0 bo‘lishining oldini olish

Konstruktor va destruktor qachon ishlaydi?

```
class Test {
    string nom;
public:
    Test(string n) : nom(n) { cout << "+" << nom << " "; }
    ~Test() { cout << "-" << nom << " "; }
};
int main() {
    Test a("a");
    {
        Test b("b");
        Test c("c");
    }
    Test *p = new Test("d");
    delete p;
}
```

▶ NATIJA

```
+a +b +c -c -b +d -d -a
```

{ ... }

blok tugaganda c, keyin b o‘chiriladi — teskari tartib

new / delete

dinamik obyekt destruktori faqat delete da

-a

main tugaganda eng oxirida

TUSHUNISHNI TEKSHIRISH

O‘ylang – Juftlikda – Bo‘lishing · 3 daqiqa

3-bo‘lim yakuni: Konstruktor va destruktor

1

Destruktor haqida qaysi gap to‘g‘ri: parametrsiz, ~ bilan, qayta yuklanmaydi?

Savol 1

2

Initsializatsiya ro‘yxati qanday yoziladi?

Savol 2

3

`{ Test x("x"); Test y("y"); }` — destruktorlar tartibi?

Savol 3

AMALIY TOPSHIRIQ

Har kim o‘z qurilmasida · 10 daqiqa · Oson · 10 ball

3-topshiriq: Talaba sinfi (konstruktor)

`Talaba` sinfini yozing: konstruktor `Talaba(string ism, int b1, int b2, int b3)`, metodlar `string getIsm() const` va `double ortacha() const`. `main()` tayyor: har bir talabaning o‘rtacha bahosini va eng yuqori o‘rtachali talabani chiqaradi.

Kirish: Birinchi qatorda `n` ($1 \le n \le 100$), keyin `n` qatorda: ism va uchta baho (0…100).

Chiqish: Har bir talaba uchun `ism o‘rtacha` (2 xona), oxirida `Eng yaxshi: ism` (teng bo‘lsa — birinchisi).

MISOL · KIRISH

```
3
Ali 80 90 100
Vali 100 100 70
Guli 95 95 95
```

MISOL · CHIQISH

```
Ali 90.00
Vali 90.00
Guli 95.00
Eng yaxshi: Guli
```

Telefon kamerasi bilan skanerlang

1. Saytga kiring (login va parol)
2. Kodni yozing va sinab ko‘ring
3. ✓ Topshirish — natija o‘qituvchiga

04

this, static va const

Yashirin this ko‘rsatkichi · zanjirli chaqiruvlar · static a’zolar · const metodlar

this->

return *this

static

const

UCH TUSHUNCHA

Obyekt, sinf va o‘zgarmaslik

1 this

Metod chaqirilgan obyekt manzili: `t1.chop()` da `this == &t1`.

2 this->ism = ism

Parametr nomi maydon nomi bilan bir xil bo‘lsa farqlash.

3 return *this

Obyektning o‘zini qaytarish: `t.a().b().c();` zanjiri.

4 static maydon

Sinfga tegishli — barcha obyektlar uchun bitta umumiy nusxa.

5 static metod

Faqat static a’zolar bilan ishlaydi: `Sinf::metod()`.

6 const metod

`int get() const` — obyektni o‘zgartirmaydi; const obyekt uchun faqat shular.

Zanjirli chaqiruvlar

```
class Talaba {
    string ism;
    int ball;
public:
    Talaba(string ism, int ball) {
        this->ism = ism;
        this->ball = ball;
    }
    Talaba& ballQosh(int x) { ball += x; return *this; }
    void chop() const { cout << ism << ": " << ball << endl; }
};
int main() {
    Talaba t("Aziza", 70);
    t.ballQosh(5).ballQosh(10).chop();
}
```

▶ NATIJA

```
Aziza: 85
```

this->ism

maydon; `ism` — parametr

Talaba&

havola qaytariladi — nusxa emas

70 + 5 + 10

zanjir: har chaqiruv o‘sha obyektni qaytaradi

Nechta obyekt yaratilgan?

```
class Talaba {
    int id;
    static int soni;
    static int keyingi;
public:
    Talaba() : id(keyingi++) { soni++; }
    ~Talaba() { soni--; }
    int getId() const { return id; }
    static int getSoni() { return soni; }
};
int Talaba::soni = 0;
int Talaba::keyingi = 1001;
int main() {
    Talaba a, b;
    { Talaba c; cout << Talaba::getSoni() << " "; }
    cout << Talaba::getSoni() << " " << b.getId();
}
```

▶ NATIJA

```
3 2 1002
```

`static int soni`

barcha obyektlar uchun bitta hisoblagich

`int Talaba::soni = 0;`

sinfdan tashqarida aniqlash

blokdan keyin

c o‘chdi — soni 2

TUSHUNISHNI TEKSHIRISH

O‘ylang – Juftlikda – Bo‘lishing · 3 daqiqa

4-bo‘lim yakuni: this, static va const

1

static maydonning xususiyati nima?

Savol 1

2

const metod ichida maydonni o‘zgartirish mumkinmi?

Savol 2

3

`return *this;` nima uchun ishlatiladi?

Savol 3

AMALIY TOPSHIRIQ

Har kim o‘z qurilmasida · 10 daqiqa · O‘rta · 20 ball

4-topshiriq: static hisoblagich

`Nuqta` sinfiga `static int soni` maydonini qo‘shing: konstruktor uni 1 ga oshirsin, destruktor 1 ga kamaytirsin. `static int jami()` metodi hozir mavjud obyektlar sonini qaytarsin. `main()` tayyor: buyruqlar bo‘yicha obyektlarni yaratadi va o‘chiradi.

Kirish: Birinchi qatorda `q` ($1 \le q \le 1000$). Buyruqlar: `yarat`, `ochir` (oxirgi yaratilganini o‘chiradi, bo‘lsa) yoki `son`.

Chiqish: Har bir `son` buyrug‘i uchun mavjud `Nuqta` obyektlari soni.

MISOL · KIRISH

```
5
yarat
yarat
son
ochir
son
```

MISOL · CHIQISH

```
2
1
```

Telefon kamerasi bilan skanerlang

1. Saytga kiring (login va parol)
2. Kodni yozing va sinab ko‘ring
3. ✓ Topshirish — natija o‘qituvchiga

05

Merosxo‘rlik

Asosiy va hosila sinf · is-a · protected · konstruktorlar tartibi · metodni qayta aniqlash

class B : public A

is-a

protected

Tartib

Mavjud sinf asosida yangi sinf

Hosila (derived) sinf asosiy (base) sinfning maydon va metodlarini meros qilib oladi va o‘zinikini qo‘shadi: “Dasturchi — Xodimning bir turi” (is-a).

Sintaksis

```
class Dasturchi : public
Xodim { ... };
```

protected

Tashqaridan yopiq, lekin hosila sinf metodlarida ko‘rinadi.

Konstruktorlar

Avval asosiy, keyin hosila; destruktorlar — teskari tartibda.

Asosiy konstruktorga argumentlar initsializatsiya ro‘yxatida: `Dasturchi(...) : Xodim(i, m), til(t) { }`

IERARXIYA

Xodim → Dasturchi

```
class Xodim {
protected:
    string ism;
public:
    Xodim(string i) : ism(i) { cout << "Xodim+ "; }
    ~Xodim() { cout << "Xodim- "; }
    void malumot() { cout << ism << " "; }
};
class Dasturchi : public Xodim {
    string til;
public:
    Dasturchi(string i, string t) : Xodim(i), til(t) { cout
            << "Dasturchi+ "; }
    ~Dasturchi() { cout << "Dasturchi- "; }
    void kodYoz() { cout << ism << ":" << til << " "; }
};
int main() {
    Dasturchi d("Aziza", "C++");
    d.malumot(); d.kodYoz();
}
```

▶ NATIJA

```
Xodim+ Dasturchi+ Aziza Aziza:C++ Dasturchi- Xodim-
```

Xodim(i)

asosiy konstruktor birinchi ishlaydi

d.malumot()

meros qilingan metod

ism

protected — hosila sinf metodida ko‘rinadi

Destruktorlar

teskari tartibda

QAYTA ANIQLASH

Hosila sinf metodni o‘zgartiradi

```cpp
class Hayvon {
protected:
    string nom;
public:
    Hayvon(string n) : nom(n) {}
    void ovoz() { cout << nom << ": ...\n"; }
};
class It : public Hayvon {
public:
    It(string n) : Hayvon(n) {}
    void ovoz() { cout << nom << ": Vov!\n"; }
};
int main() {
    It it("Bobik");
    it.ovoz();
    it.Hayvon::ovoz();
    Hayvon *p = &it;
    p->ovoz();
}
```

▶ NATIJA

```
Bobik: Vov!
Bobik: ...
Bobik: ...
```

`it.ovoz()`

It versiyasi

`Hayvon::ovoz()`

asosiy sinf versiyasini aniq chaqirish

`p->ovoz()`

?! Hayvon versiyasi — virtual yo‘q (keyingi bo‘lim)

TUSHUNISHNI TEKSHIRISH

O‘ylang – Juftlikda – Bo‘lishing · 3 daqiqa

5-bo‘lim yakuni: Merosxo‘rlik

1

Hosila obyekt yaratilganda konstruktorlar tartibi?

Savol 1

2

private a’zolar meros qilinadimi?

Savol 2

3

Asosiy sinf metodini hosila sinfdan qanday chaqirish mumkin?

Savol 3

06

Polimorfizm va abstrakt sinflar

virtual · override · dinamik bog‘lanish · virtual destruktor · sof virtual = 0

virtual | override | = 0 | Abstrakt

Statik yoki dinamik?

virtual yo‘q — statik bog‘lanish

- Metod ko‘rsatkich turiga qarab tanlanadi
- Kompilyatsiya vaqtida hal qilinadi
- `Hayvon *p = ⁢ p->ovoz();` → Hayvon versiyasi

virtual — dinamik bog‘lanish

- Metod obyektning haqiqiy turiga qarab tanlanadi
- Dastur ishlayotganda hal qilinadi
- Bitta `p->ovoz()` — it uchun “Vov”, mushuk uchun “Miyov”

Qoida: sinf asosiy sinf sifatida ishlatilsa, destruktorini ham `virtual` qiling — aks holda `delete p;` hosila qismni to‘g‘ri o‘chirmaydi.

Bitta chaqiruv — uch xil natija

```
class Hayvon {
public:
    virtual ~Hayvon() {}
    virtual string ovoz() { return "..."; }
};
class It : public Hayvon {
public:
    string ovoz() override { return "Vov!"; }
};
class Mushuk : public Hayvon {
public:
    string ovoz() override { return "Miyov!"; }
};
int main() {
    Hayvon *zoo[3] = {new It, new Mushuk, new Hayvon};
    for (Hayvon *h : zoo) cout << h->ovoz() << " ";
    for (Hayvon *h : zoo) delete h;
}
```

▶ NATIJA

```
Vov! Miyov! ...
```

virtual

haqiqiy turga qarab tanlanadi

override

C++11: “aynan qayta aniqlayapman” — xato imzoni kompilyator ushlaydi

virtual ~Hayvon()

delete to‘g‘ri ishlashi uchun

ABSTRAKT SINF

Sof virtual metod: = 0

```
class Shakl {
public:
    virtual double yuza() const = 0;
    virtual ~Shakl() {}
};
class Doira : public Shakl {
    double r;
public:
    Doira(double r) : r(r) {}
    double yuza() const override { return 3.14 * r * r; }
};
class Kvadrat : public Shakl {
    double a;
public:
    Kvadrat(double a) : a(a) {}
    double yuza() const override { return a * a; }
};
int main() {
    Shakl *s[] = {new Doira(1), new Kvadrat(3)};
    for (Shakl *x : s) { cout << x->yuza() << " "; delete x; }
}
```

▶ NATIJA

```
3.14 9
```

= 0

sof virtual — tanasi yo‘q

Shakl s;

XATO: abstrakt sinfdan obyekt yaratib bo‘lmaydi

Interfeys

hosila sinf barcha sof virtual metodlarni aniqlashi shart

TUSHUNISHNI TEKSHIRISH

O‘ylang – Juftlikda – Bo‘lishing · 3 daqiqa

6-bo‘lim yakuni: Polimorfizm va abstrakt sinflar

1

Asosiy sinf ko‘rsatkichi orqali hosila metod chaqirilishi uchun nima kerak?

Savol 1

2

Abstrakt sinf nima?

Savol 2

3

Nima uchun virtual destruktor kerak?

Savol 3

XULOSA

Asosiy xulosalar

1 Sinf va obyekt

Sinf — qolip, obyekt — nusxa; a’zolarga `.` orqali murojaat.

2 Inkapsulyatsiya

Maydonlar `private`, kirish — tekshiruvchi `public` metodlar.

3 Konstruktor / destruktor

Obyekt tug‘ilganda va o‘lganda avtomatik; destruktorlar teskari tartibda.

4 this, static, const

this — joriy obyekt; static — sinfga umumiy; const metod — o‘zgartirmaydi.

5 Merosxo‘rlik

`class B : public A`; avval asosiy konstruktor; protected hosilada ko‘rinadi.

6 Polimorfizm

`virtual` + `override`; sof virtual `= 0` — abstrakt sinf.

TAKRORLASH

Yakunlashdan oldin sakkizta savol

1. OYD ning to‘rt tamoyili?
2. `class A { int x; };` da x qanday?
3. Konstruktorning qaytarish turi?
4. Destruktor nomi qanday boshlanadi?
5. `this` nima?
6. static maydon qayerda aniqlanadi?
7. Hosila sinf yaratilganda qaysi konstruktor birinchi?
8. Abstrakt sinfdan obyekt yaratish mumkinmi?

LABORATORIYA ISHI

9-laboratoriya ishi: 6 ta dasturlash topshirig‘i

5 **Kasr sinfi**
O‘rta · 20 ball

6 **Vaqt sinfi**
O‘rta · 20 ball

7 **Shakllar (merosxo‘rlik)**
O‘rta · 20 ball

8 **Stek sinfi**
O‘rta · 20 ball

9 **Xodimlar maoshi (abstrakt sinf)**
O‘rta · 20 ball

10 **Kutubxona**
Qiyin · 30 ball

Telefon kamerasi bilan skanerlang

Kod saytda yoziladi va server tomonidan avtomatik tekshiriladi.

O‘qituvchi kodingizni ko‘rib, baho qo‘yadi.

CHIQISH CHIPTASI

Darsni tark etishdan oldin

Javoblaringizni qog‘ozga yozing va o‘qituvchiga topshiring.

1

`Kitob` sinfi uchun 3 ta maydon va 2 ta metod o‘ylab toping.

2

Nima uchun maydonlarni private qilish foydali? Bitta misol keltiring.

3

virtual bo‘lmagan metod asosiy ko‘rsatkich orqali chaqirilsa nima bo‘ladi?