# Dars-1 print narmativ

### Masala: Do'kondagi savdo hisoboti

Do'koningizda bugun sotilgan mahsulotlar haqida hisobot tayyorlang. Quyidagi ma'lumotlarni kiritishingiz va natijalarni hisoblab chiqarishingiz kerak:

#### Talablar:

- 1. Mahsulot nomini kiritish (str() orgali).
- 2. Bugun sotilgan mahsulotlar sonini kiritish (int() orqali).
- 3. Har bir mahsulotning narxini kiritish (float() orqali).
- 4. Umumiy savdo summasini hisoblash:
- 5. Hisoblangan natijalarni print() yordamida foydalanuvchiga ko'rsatish.

### Savollar va ko'rsatmalar:

- 1. Mahsulot nomini qanday kiritasiz va qanday saqlaysiz?
  - Mahsulot nomi oddiy matn bo'lib, uni input() orqali o'qing va str turida saqlang.
- 2. Mahsulotlar sonini qanday kiritasiz va qanday saqlaysiz?
  - Mahsulot soni butun son bo'ladi, uni input() yordamida kiritib, int() orqali butun son turiga o'tkazing.
- 3. Mahsulot narxini qanday kiritasiz va qanday saqlaysiz?
  - Mahsulot narxi haqiqiy son (float), uni input() orqali kiritib, float() turiga o'tkazing.
- 4. Umumiy savdo summasini qanday hisoblaysiz?
  - Umumiy savdo summasi mahsulot sonini mahsulot narxiga ko'paytirish orqali hisoblanadi.
- 5. Natijalarni qanday chiqarasiz?
  - Natijalarni print() yordamida chiqarib, foydalanuvchiga mahsulot nomi, sotilgan soni, narxi va savdo summasini ko'rsating.

#### Misol kiritish va natijalar:

Sotilgan mahsulot nomini kiriting: Non

Bugun sotilgan Non sonini kiriting: 50

Har bir Nonning o'rtacha narxini kiriting (so'mda): 3000

Chiqish:

Natijalar:

Mahsulot nomi: Non

Bugun sotilgan mahsulotlar soni: 50

Har bir Nonning o'rtacha narxi: 3000.00 so'm

Umumiy savdo summasi: 150000.00 so'm

# Dars-2 data type

### Masala: Sinfdagi o'quvchilarni boshqarish tizimi

Maktab sinfi uchun o'quvchilarni boshqarish tizimini yarating. Quyidagi ma'lumotlarni saqlash va boshqarish kerak:

- 1. O'quvchilar ro'yxati (list).
- 2. Har bir o'quvchining reytingi (tuple).
- 3. O'quvchilarning tug'ilgan kunlari (dictionary).
- 4. Darsga qatnashayotgan o'quvchilar (set).

### Vazifa:

Quyidagi talablar asosida dastur yozing:

### 1. O'quvchilarni qo'shish va o'chirish (list):

- Yangi o'quvchini sinf ro'yxatiga qo'shish uchun append() funksiyasidan foydalaning.
- O'quvchi sinfdan ketganida, uni remove() yordamida o'chiring.

### 2. Reytingni ko'rish (tuple):

- Har bir o'quvchi uchun reytinglar tuple shaklida saqlansin (masalan, baholari: (5, 4, 3)).
- Reytinglarni chiqarish uchun for siklidan foydalaning.

### 3. Tug'ilgan kunlarni boshqarish (dictionary):

- o O'quvchining ismi va tug'ilgan kunini saqlang.
- Tug'ilgan kun ma'lumotlarini keys() orqali ko'rsating.

### 4. Darsga qatnashuvchilarni boshqarish (set):

- O'quvchilarni add() yordamida qatnashuvchilar ro'yxatiga qo'shing.
- o Agar darsga kelmagan bo'lsa, uni ro'yxatdan olib tashlang.

### 5. Shartli operatorlardan foydalaning (if, elif, else):

o Agar o'quvchi ro'yxatda bo'lmasa, qo'shimcha xabar chiqaring.

o Agar reytingi yaxshi bo'lsa, uni tabriklang.

```
oquvchilar = ["Ali", "Vali", "Laylo", "Mavluda"]

# Reytinglar (tuple)

reytinglar = { "Ali": (5, 4, 3), "Vali": (4, 4, 5), "Laylo": (5, 5, 5), "Mavluda": (3, 4, 3) }

# Tug'ilgan (dictionary)

tugilgan_kunlar = { "Ali": "2005-05-01", "Vali": "2004-08-15", "Laylo": "2005-09-12", "Mavluda": "2004-12-25" }
```

Yangi o'quvchi ismini kiriting: Dilshod

Dilshodning tug'ilgan kunini kiriting (YYYY-MM-DD): 2005-11-20

O'chiriladigan o'quvchi ismini kiriting: Vali

Darsga qatnashayotgan o'quvchini kiriting: Dilshod

Darsga kelmagan o'quvchi ismini kiriting: Ali

**CHIQISH** 

Dilshod ro'yxatga qo'shildi.

Vali ro'yxatdan o'chirildi.

O'quvchilar va ularning reytinglari:

Ali: (5, 4, 3)

- Alining reytingi yaxshi!

Laylo: (5, 5, 5)

- Layloning reytingi a'lo! Tabriklaymiz!

Mavluda: (3, 4, 3)

Dilshod: (0, 0, 0)

O'quvchilarning tug'ilgan kunlari:

Ali: 2005-05-01

Laylo: 2005-09-12

Mavluda: 2004-12-25

Dilshod: 2005-11-20

Dilshod qatnashuvchilar ro'yxatiga qo'shildi.

Ali qatnashuvchilar ro'yxatidan o'chirildi.

Darsga qatnashayotgan o'quvchilar:

{'Dilshod', 'Laylo'}

Dars-3 if.elif,else,while

### Masala: Mahsulot Ombori Boshqaruvi

Siz bir ombor boshqaruv tizimini yaratishingiz kerak. Omborda turli mahsulotlar saqlanadi, va ushbu dastur quyidagi funktsiyalarni bajarishi lozim:

### 1. Mahsulotni tekshirish (if, elif, else):

- Foydalanuvchi mahsulot nomini kiritsa, dastur omborda mahsulot bor-yo'qligini tekshiradi.
- Agar mahsulot mavjud bo'lsa, uning sonini ko'rsatadi.
- Agar mahsulot omborda qolmagan bo'lsa, bu haqda xabar beradi.
- Agar mahsulot umuman ro'yxatda bo'lmasa, yangi mahsulot qo'shishni taklif qiladi.

### 2. Mahsulotlarni ko'rish (for):

o Ombordagi barcha mahsulotlarni va ularning miqdorini chiqaradi.

### 3. Yangi mahsulot qo'shish yoki miqdorini yangilash (while):

- Omborga yangi mahsulot qo'shish imkoniyatini beradi.
- Foydalanuvchi mahsulot va uning miqdorini kirita oladi.

### 4. Shartli davom ettirish (while):

 Foydalanuvchi "davom ettirishni xohlaysizmi?" savoliga qarab, dasturni davom ettiradi yoki to'xtatadi.

ombor = { "olma": 50, "banan": 30, "anor": 20, "shaftoli": 15 }

### Misol kirish va chiqish:

#### Kirish:

Tanlovni kiriting (1-4): 1

Qaysi mahsulotni tekshirmoqchisiz? banan

#### Chiqish:

banan omborda bor. Migdori: 30 dona.

#### Kirish:

Tanlovni kiriting (1-4): 3

Yangi mahsulot nomini kiriting: uzum Uzum uchun miqdorni kiriting: 25 Yana mahsulot qo'shishni xohlaysizmi? (ha/yo'q): yo'q chiqish Uzum omborga qo'shildi. Miqdor: 25 dona. **KIRISH** Tanlovni kiriting (1-4): 2 Chiqish === Ombordagi mahsulotlar === Olma: 50 dona Banan: 30 dona Anor: 20 dona Shaftoli: 15 dona Uzum: 25 dona

## Dars-4 def

### Masala: Talabalar Baho Hisoblash Tizimi

Siz bir maktab yoki universitetda talabalar uchun baho hisoblash tizimi yaratishingiz kerak. Ushbu tizimda talabalar ismlari, baholari va ularning umumiy natijalari bilan ishlash uchun **funksiyalar**dan foydalaniladi.

### Masala Tavsifi:

- 1. Foydalanuvchidan ma'lumot olish:
  - Talabaning ismini va fanlar bo'yicha baholarini kiritish imkoniyatini bering.
- 2. Funksiyalar yordamida hisoblash:
  - Har bir talaba uchun baholarni qo'shib, o'rtacha bahoni hisoblang.
  - O'rtacha bahoga qarab, talabaning darajasini ("A'lo", "Yaxshi", "Qoniqarli",
     "Qoniqarsiz") aniqlang.
- 3. Talabalarni ro'yxatga olish va natijalarni ko'rsatish:
  - Foydalanuvchiga barcha talabalarni va ularning baholarini ko'rsatish imkoniyatini bering.
- 4. Dasturni davom ettirish:
  - Dastur foydalanuvchining xohishiga ko'ra yangi talaba qo'shishi yoki tizimdan chiqishi mumkin.

### Misol Kirish va Chiqish:

Kirish:

```
Tanlovni kiriting (1-3): 1

Talabaning ismini kiriting: Aziz

Talabaning baholarini vergul bilan kiriting (masalan: 85, 90, 78): 90, 85, 92
```

#### Chiqish:

```
Aziz muvaffaqiyatli qo'shildi!
Kirish:
Tanlovni kiriting (1-3): 2
Chiqish:
=== Barcha Talabalar Ro'yxati ===
Ism: Aziz, O'rtacha: 89.00, Daraja: Yaxshi
Kirish:
Tanlovni kiriting (1-3): 1
Talabaning ismini kiriting: Gulnoza
Talabaning baholarini vergul bilan kiriting (masalan: 85, 90, 78):
60, 55, 70
Chiqish:
Gulnoza muvaffaqiyatli qo'shildi!
Kirish:
```

```
Tanlovni kiriting (1-3): 2
```

### Chiqish:

=== Barcha Talabalar Ro'yxati ===

Ism: Aziz, O'rtacha: 89.00, Daraja: Yaxshi

Ism: Gulnoza, O'rtacha: 61.67, Daraja: Qoniqarli

# Dars-5 file handling

### ★ Vazifa: Talabalar Ma'lumotlarini Boshqarish

Siz talabalar roʻyxati bilan ishlaydigan interaktiv dastur yaratishingiz kerak. Ushbu dastur:

- Fayldan talabalar roʻyxatini oʻqiydi
- **V** Yangi talabalarni qoʻshish imkonini beradi
- Muayyan talabaning ma'lumotlarini qidirish va chiqarish funksiyasiga ega boʻladi
- Ma'lumotlarni o'zgartirish va o'chirish imkonini beradi
- Barcha talabalar roʻyxatini chiqaradi

### Bosqichma-bosqich talablar:

students.txt degan fayl yarating va ichiga talabalar roʻyxatini yozing:

- 1,Ali,Mathematics,85i
- 2, Vali, Physics, 90
- 3, Hasan, Computer Science, 95
- 4,Akmal,Chemistry,80

Format: ID, Ism, Fan, Baho

- 2 Dastur ushbu fayldan ma'lumotlarni oʻqisin
- 3 Yangi talaba qoʻshish funksiyasini bajarsin
- 4 Muayyan ID boʻyicha qidirish imkonini yarating
- 5 Talabaning bahosini oʻzgartirish imkonini bering
- 6 Talabani roʻyxatdan oʻchirish funksiyasini qoʻshing
- Barcha talabalarni faylga yozib qoʻying

- Talabalar Boshqaruv Tizimi
- 1. Barcha talabalarni koʻrish
- 2. Yangi talaba qoʻshish
- 3. Talabani qidirish
- 4. Bahoni yangilash
- 5. Talabani oʻchirish
- 6. Chiqish

Tanlang (1-6): 1

### \*\*Talabalar Roʻyxati:\*\*

ID: 1, Ism: Ali, Fan: Mathematics, Baho: 85

ID: 2, Ism: Vali, Fan: Physics, Baho: 90

ID: 3, Ism: Hasan, Fan: Computer Science, Baho: 95

ID: 4, Ism: Akmal, Fan: Chemistry, Baho: 80

Dars-6 try

Masala: Bankdagi hisobdan pul yechish

Tasavvur qiling, sizda bank tizimi mavjud, unda foydalanuvchilar hisobdan pul yechish imkoniyatiga ega. Biroq, pul yechish jarayonida ba'zi xatoliklar yuzaga kelishi mumkin, masalan, hisobda yetarli

mablag'ning bo'lmasligi yoki foydalanuvchi noto'g'ri miqdor kiritishi.

**Talablar:** 

1. Foydalanuvchi hisobidan pul yechishi kerak.

2. Pul yechishda xatoliklar yuzaga kelishi mumkin: mablag' yetarli emas yoki noto'g'ri miqdor

kiritilishi.

3. Har doim jarayon oxirida hisob holatini chiqarish kerak (bunda finally ishlatiladi).

balance=1000

### Misol ishlash jarayoni:

1. Pul yechish muvaffaqiyatli bo'lsa:

Pul yechish tizimiga xush kelibsiz!

Qancha pul yechmoqchisiz? 500

500.0 so'm hisobdan yechildi. Joriy balans: 500.0 so'm.

Jarayon tugadi. Joriy balans: 500.0 so'm.

Noto'g'ri miqdor kiritilishi:

Pul yechish tizimiga xush kelibsiz!

Nechta pul yechmoqchisiz? -100

Xatolik: Pul miqdori manfiy yoki nol bo'lmasligi kerak.

Jarayon tugadi. Joriy balans: 1000 so'm.

### Hisobda yetarli mablag' yo'qligi:

Pul yechish tizimiga xush kelibsiz!

Nechta pul yechmoqchisiz? 1500

Xatolik: Hisobda yetarli mablag' yo'q.

Jarayon tugadi. Joriy balans: 1000 so'm.

# Dars-7 oop

#### Savol:

Siz restoran uchun taomlar menyusini boshqaruvchi dastur yaratmoqchisiz. Har bir taom turini (Starter, Asosiy taom, Desert) alohida klasslar bilan ifodalashni xohlaysiz. Bu dasturda foydalanuvchilar quyidagi funksiyalarni bajarishlari kerak:

- 1. **Yangi taom qo'shish** Foydalanuvchi yangi taomni menyuga qo'shishi kerak. Har bir taom nomi, tavsifi, narxi va turi (starter, asosiy taom yoki desert) bo'lishi kerak. Asosiy taomlar va desertlar uchun, shuningdek, vegetarian yoki gluten-free bo'lishi mumkinligini belgilash kerak.
- 2. **Taomni o'chirish** Foydalanuvchi menyudan taomni nomi orqali o'chirishi kerak. Agar taom menyuda bo'lmasa, foydalanuvchiga xabar berilishi kerak.
- 3. **Menyuni ko'rish** Foydalanuvchi menyudagi barcha taomlarni ko'rish imkoniyatiga ega bo'lishi kerak
- 4. **Buyurtma qilish** Foydalanuvchi menyudan taom tanlab buyurtma qilishi kerak. Buyurtma yakunlangach, jami narx hisoblanishi va agar buyurtma 100,000 so'mdan katta bo'lsa, 10% cheqirma qo'llanilishi kerak.
- 5. **Dasturdan chiqish** Foydalanuvchi dasturni tuqatishi kerak.

Shu asosda, quyidagi kodni yozishingiz kerak. Kodda taomlarni yaratish uchun Dish, Starter, MainCourse, Dessert kabi klasslar va buyurtmalarni boshqarish uchun Order va Menu klasslari ishlatiladi. Har bir klassda kerakli metodlar va xususiyatlar mavjud bo'lishi kerak.

#### Savolga javob:

Dasturda quyidagi klasslar va metodlar bo'lishi kerak:

- 1. **Dish klassi** Taom nomi, tavsifi va narxini saglash uchun.
- 2. **Starter, MainCourse, Dessert klasslari** Maxsus taom turlari uchun.
- 3. Menu klassi Menyudagi barcha taomlarni boshqarish uchun.
- 4. **Order klassi** Buyurtma qilish va narxni hisoblash.

Dastur foydalanuvchiga menyudagi taomlarni qo'shish, o'chirish, buyurtma qilish va narxni hisoblash imkoniyatini beradi.

### Dars-8 Decorator va Generator

### 📌 Masala: Restorandagi Buyurtmalarni Qabul Qilish va Tayyorlash

Endi har bir bosqich uchun chiqadigan natijalarni ham koʻrib chiqamiz.

# 1-qadam: Oddiy funksiya orqali barcha buyurtmalarni birdan tayyorlash

### Vazifa:

- Quyidagi buyurtmalar bor: ["osh", "shashlik", "mastava"]
- Funksiya barcha buyurtmalarni birdan chiqaradi.

### **X** Kod mantiqiy tushuntirish:

- 1. Funksiya roʻyxatni qabul qiladi.
- 2. Har bir buyurtmani tayyor deb chiqaradi.
- 3. Barcha natijalar birdan qaytadi.

### Chiqadigan natija:

Tayyor boʻldi: osh

Tayyor boʻldi: shashlik

Tayyor boʻldi: mastava

X Muammo: Barcha buyurtmalar bir vaqtda chiqdi, navbat bilan emas!

# **?** 2-qadam: Generator yordamida buyurtmalarni navbat bilan chiqarish

### **V** Vazifa:

- Buyurtmalar **birma-bir** chiqarilsin.
- Har bir buyurtma navbat bilan qaytarilsin.

### **Yes** Kod mantiqiy tushuntirish:

- 1. Funksiya **generator** (yield) yordamida ishlaydi.
- 2. Har bir buyurtma bir martada bittadan qaytadi.
- 3. Xotirani tejash uchun butun roʻyxat bir vaqtda qaytmaydi.

### Chiqadigan natija:

Tayyor boʻldi: osh

Tayyor boʻldi: shashlik

Tayyor boʻldi: mastava

✓ Afzallik: Buyurtmalar navbat bilan chiqariladi, lekin hali kechikish yoʻq.

### **3-qadam: Dekorator yordamida buyurtmalarni logga yozish**

### **V** Vazifa:

- Funksiya boshlanishida "Buyurtmalar qabul qilindi" degan yozuv chiqsin.
- Har bir buyurtma tayyor boʻlganda logga yozilsin.

### **X** Kod mantiqiy tushuntirish:

- 1. Dekorator funksiya chaqirilishidan oldin logga yozib qoʻyadi.
- 2. Buyurtmalar avvalgidek chiqariladi.

### Chiqadigan natija:

[LOG] Buyurtmalar qabul qilindi

Tayyor boʻldi: osh

Tayyor boʻldi: shashlik

Tayyor boʻldi: mastava

✓ Afzallik: Endi loglar bor, buyurtmalar kuzatib boriladi.

### 4-qadam: Har bir buyurtmani 2 soniya kechiktirib tayyorlash

### ✓ Vazifa:

- Har bir buyurtma 2 soniya kechikish bilan chiqsin.
- Buyurtmalarni ketma-ket chiqarish real oshxona jarayoni kabi boʻlsin.
- **Y** Kod mantiqiy tushuntirish:
  - 1. Har bir buyurtmadan oldin time.sleep(2) bilan 2 soniya kutiladi.
  - 2. Buyurtmalar navbat bilan chiqadi, hammasi birdan emas.
- Chiqadigan natija (2 soniya oraligʻida chiqadi):

[LOG] Buyurtmalar qabul qilindi
(2 soniya kutish...)
Tayyor boʻldi: osh
(2 soniya kutish...)
Tayyor boʻldi: shashlik
(2 soniya kutish...)

Tayyor boʻldi: mastava

Afzallik: Buyurtmalar real vaqtda navbat bilan chiqadi.

### **5**-qadam: Cheksiz buyurtmalarni qabul qilish

### ✓ Vazifa:

- Buyurtmalar cheksiz qabul qilinsin.
- "stop" kiritilsa, buyurtmalar qabul qilish toʻxtatiladi.
- Kod mantiqiy tushuntirish:
  - 1. Har safar foydalanuvchi yangi buyurtma kiritadi.
  - 2. "stop" kiritilmaguncha buyurtmalar davom etadi.
  - 3. Har bir buyurtma 2 soniya kechikish bilan tayyorlanadi.
- Chiqadigan natija (foydalanuvchi tomonidan kiritilgan buyurtmalarga qarab chiqadi):

[LOG] Buyurtmalar qabul qilindi

Qaysi taomni buyurtma qilasiz? > osh

(2 soniya kutish...)

Tayyor boʻldi: osh

Qaysi taomni buyurtma qilasiz? > shashlik

(2 soniya kutish...)

Tayyor boʻldi: shashlik

Qaysi taomni buyurtma qilasiz? > mastava

(2 soniya kutish...)

Tayyor boʻldi: mastava

Qaysi taomni buyurtma qilasiz? > stop

Buyurtmalar qabul qilish tugatildi.

✓ Afzallik: Endi foydalanuvchi istalgan vaqtda buyurtma bera oladi va uni toʻxtatish ham mumkin.

### ★ Yakuniy Xulosa:

- 1 Oddiy funksiya barcha buyurtmalarni birdan qaytardi.
- 2 Generator (yield) har bir buyurtmani bittadan chiqarishga imkon berdi.
- 3 Dekorator log qoʻshib, buyurtmalarni kuzatish imkonini berdi.
- 4 Kechikish (sleep(2)) qoʻshilib, buyurtmalar real oshxona jarayoni kabi boʻldi.
- 5 Cheksiz buyurtma qabul qilish orqali foydalanuvchi interaktiv boshqaruv oldi.

# Dars-9 Context Manager

#### √ Vazifa:

Siz logging moduli va context manager (klass ichida \_\_enter\_\_() va \_\_exit\_\_() metodlari) yordamida fayl ochish, unga yozish va xatolarni logga qayd qilish funksiyasini yozishingiz kerak.

### Talablar:

### **1** Loglar yuritish:

- Dastur ishga tushishi bilan "app.log" fayliga "Dastur boshlandi" yoziladi.
- Har bir jarayon INFO yoki ERROR logga yozilishi kerak.

### 2 Context Manager (FileManager klassi) yaratish:

- \_\_enter\_\_() → Faylni ochib log yozadi.
- \_\_exit\_\_() → Faylni yopib, xatolarni ushlaydi va logga yozadi.

### 3 Faylga yozish:

- "data.txt" nomli fayl ochiladi.
- "Bu test ma'lumoti" matni yoziladi.

### 4 Xatolarni ushlash va qayd qilish:

- Noto'g'ri fayl nomi (data?.txt) bilan sinov qiling.
- Agar xatolik yuz bersa, ERROR logga yozilishi kerak.

### 📌 Kutilayotgan natija (konsolda va app. 1 og faylida):

### Muvaffaqiyatli bajarilsa:

- [INFO] Dastur boshlandi.
- [INFO] Fayl "data.txt" ochildi.
- [INFO] Faylga yozilmoqda...
- [INFO] Faylga muvaffaqiyatli yozildi.
- [INFO] Fayl yopildi.

### X Xatolik yuz bersa:

• [ERROR] Fayl ochishda xatolik yuz berdi: [Xato tafsiloti]

### **®** Bonus:

- Kodni shunday yozingki, u boshqa fayllar bilan ishlaganda ham moslashuvchan boʻlsin.
- with operatori yordamida FileManager dan foydalaning.
- Barcha talablarni bajaring va natijalarni app. log faylida tekshiring!

•

# Dars-10 Multithreading

### 1-QISM: Foydalanuvchi ma'lumotlarini olish (Multithreading)

### Vazifa:

- Sizda 5 ta foydalanuvchi bor: ["Ali", "Vali", "Sami", "Lola", "Dina"]
- Har bir foydalanuvchi uchun ma'lumot olish 2 soniya vaqt oladi.
- Multithreading yordamida hamma foydalanuvchilarni parallel ravishda tekshiring.

### **P** E'tibor bering:

- Agar multithreading ishlatilmasa, umumiy vaqt 10 soniya boʻladi.
- Agar multithreading ishlatilsa, umumiy vaqt 2 soniya boʻladi.

### **Sizning maqsadingiz:**

 Threading moduli yordamida har bir foydalanuvchini tekshirishni alohida thread orqali bajaring.

### 2-QISM: Ma'lumotlarni qayta ishlash (Multiprocessing)

### Vazifa:

- Har bir foydalanuvchi uchun ma'lumotlar olish jarayoni tugagach, ularning ismlarini bosh harf bilan yozish kerak.
- Har bir foydalanuvchining ismini oʻzgartirish **1 soniya vaqt oladi**.
- Multiprocessing yordamida har bir foydalanuvchini alohida process orqali qayta ishlang.

### **£** E'tibor bering:

- Agar multiprocessing ishlatilmasa, umumiy vaqt 5 soniya boʻladi.
- Agar multiprocessing ishlatilsa, umumiy vaqt 1 soniya boʻladi.

### **©** Sizning maqsadingiz:

 Multiprocessing moduli yordamida har bir foydalanuvchini alohida process orqali qayta ishlang.

### 3-QISM: Natijalarni taqqoslash

### Vazifa:

- Dasturni multithreading va multiprocessing bilan yozing.
- Boshlanish vaqti va yakunlanish vaqtini hisoblang.

• Ikkita yondashuvni solishtiring: qaysi biri tezroq ishlaydi?

### **★** E'tibor bering:

- Multithreading ma'lumot olish tezlashadi
- Multiprocessing ma'lumotni qayta ishlash tezlashadi

### **©** Sizning maqsadingiz:

- Threading va multiprocessing birgalikda ishlaydigan dasturni yozing.
- Oʻquvchilar ushbu kodni yozib, natijani koʻrib chiqsinlar.

# Dars-11 Testing

### Masala: Elektron doʻkondagi mahsulot savdolarini tekshirish

### Vazifa:

Sen **elektron do'kon uchun savdo tizimi** yozayotgan backend dasturchi san. Do'konning asosiy qismi – mahsulotlarni sotib olish va hisob-kitob qilish. Biz ushbu tizimning ishlashini **unittest** yoki **pytest** yordamida test qilamiz.

### Berilgan shartlar:

- Product (Mahsulot) klassi:
  - o name: mahsulot nomi
  - o price: mahsulot narxi (musbat son boʻlishi kerak)
  - o quantity: mavjud miqdor (musbat son boʻlishi kerak)
- ShoppingCart (Savat) klassi:
  - add\_product(): savatga mahsulot qo'shadi
  - o remove\_product(): savatdan mahsulotni olib tashlaydi
  - get\_total\_price(): jami narxni qaytaradi

### ✓ Vazifalar:

- 1. Klasslarni yozish Product va ShoppingCart klasslarini yoz.
- 2. Test yozish Quyidagi holatlar uchun test yoz:
  - Mahsulot yaratishda narx yoki miqdor notoʻgʻri berilsa, ValueError qaytishi kerak.
  - Savatga mahsulot qoʻshilganda, jami narx toʻgʻri hisoblanishi kerak.
  - Savatdan mahsulot oʻchirilganda, jami narx yangilanishi kerak.
  - Mahsulot miqdori nol boʻlsa, savatga qoʻshib boʻlmasligi kerak.

### Konsolga chiqishi kutilgan natija:

==== 4 passed in 0.12s ====

Agar hamma testlar oʻtadigan boʻlsa, shunday natija chiqishi kerak:				
==== test session starts ====				
collected 4 items				
test_shopping_cart.py # (Barcha testlar muvaffaqiyatli oʻtdi)				