

§- 7. AMALIY MASHG'ULOT

Funksional dasturlash va unda amaliy masalalarni hal qilish

Mashg'ulot maqsadi: Funksional dasturlash va unda amaliy masalalarni hal qilish haqidagi bilimlarni nazariy va amaliy jihatdan talabalarga o'rgatish.

Mashg'ulotda foydalaniladigon texnik jihozlar va vositalar: ko'rgazmali qurollar, proektor, kompyuter, elektron doska, zamonaviy (xususan, interfaol) ta'lim usullari, fanning o'quv uslubiy majmuasi va videodarslar.

Nazariy qism

Katta hajmli ma'lu motlarni qayta ishlashda ma'lumotlar bazalarida funksional dasturlashga kirishjarayoni o'zgarmaslik, sof funksiyalar va yuqori tartibli funksiyalar kabi funksional dasturlash tushunchalari haqida qisqacha ma'lumot beradi.Ma'lumotlar bazalari kontekstida funksional dasturlashdan foydalanishning ma'lumotlar yaxlitligi, sinovdan o'tkazish qulayligi va masshtablilik kabi afzalliklarini muhokama qiladi.Funksional ma'lumotlarni modellashtirish.

Ma'lumotlar bazalarida ma'lumotlar modellarini loyihalashda funksional dasturlash tushunchalarini qanday qo'llash mumkinligini o'rganadi.Ma'lumotlarni modellashtirish kontekstida algebraik ma'lumotlar turlari, naqshlarni moslashtirish va o'zgarmaslik kabi usullarni muhokama qiladi.Funksional ma'lumotlarni modellashtirish usullaridan foydalangan holda murakkab munosabatlar va cheklovlarni qanday ko'rsatishga misollar keltiradi.

Funksional so'rovlar va ma'lumotlarni manipulyatsiya qilish. Funksional dasturlashdan ekspressiv va tuziladigan so'rovlarni yozish uchun qanday foydalanish mumkinligini ko'rsatadi.

- Ma'lumotlarni so'rash va o'zgartirish uchun yuqori tartibli funktsiyalardan, xaritalash/kamaytirish operatsiyalaridan va monadlardan foydalanishni muhokama qiladi.
- Funktsional dasturlash usullaridan foydalangan holda so'rov va ma'lumotlarni manipulyatsiya qilishda keng tarqalgan muammolarni hal qilish bo'yicha misollar keltiradi. Ma'lumotlar bazasi operatsiyalari bilan ishlash.

Dasturlash sohasidagi rivojlanishlar bilan birga, funktsional dasturlash paradigmasi ham yuqori darajada o'zgarish yaratilgan. Funktsional dasturlash, amaliy masalalarini hal qilishga oid yangi usullar va qulayliklar taqdim etadi. Funktsional dasturlashning maqsadi, undagi amaliy masalalarni hal qilishning asosiy vazifalari va unda foydalaniladigan dasturi vositalardan funktsional dasturlash, ma'lumotlar bazasi, immutability, Rekursiv funktsiyalar, Side-effect, Lamb, Map, Filter, Reduce

- Ma'lumotlar bazalarida tranzaksiya yaxlitligini ta'minlash uchun funktsional dasturlashdan qanday foydalanish mumkinligini o'rganadi.
- tranzaksiyalarni boshqarish kontekstida o'zgarmaslik, ma'lumotnoma shaffofligi va sof funktsiyalar kabi usullarni muhokama qiladi.
- Funktsional dasturlash tamoyillaridan foydalangan holda murakkab tranzaksiya stsenariylarini qanday hal qilish haqida misollar keltiradi.

Funktsional dasturlashning asosiy tushunchalari:

- Funktsional dasturlashning matn tili:

Funktsional dasturlash paradigmasi, matn tilisifatida funktsiyalarni asosiy element sifatida qo'llaydi. Bu, amaliy masalalarni hal qilishda matematik va algoritmik qo'llanma usullarini qo'llashga imkon beradi.

- Side-effect yuqori darajada kamayadi: Funktsional dasturlashda funktsiyalar o'zining argumentlari va kuchlari bilan modifikatsiyalarni amalga oshiradi, lekin o'zgartirishlar funktsiya tashqi o'zgaruvchilariga ta'sir qilmaydi.

- Tilning immutability sifati: Funksional dasturlash tilida o'zgaruvchilar o'zgartirilmaydigan (immutability) sifatga ega bo'ladi. Bu, dasturning xatolarini kamaytiradi va amaliyotni qulaylashtiradi.

Funksional dasturlashning asosiy qadamlari:

- Funksiyalar yaratish: Funksional dasturlashda funksiyalar birinchi - klass obyektlar sifatida qo'llaniladi. Funksiyalar parametrlar sifatida berilishi va natijalarni qaytarishi mumkin.
- Rekursiv funksiyalar: Funksional dasturlash paradigmasi rekursiv funksiyalarni osonlik bilan yaratishga imkon beradi. Bu, amaliy masalalarni hal qilishda o'zgaruvchilar va qaytuvchi funksiyalar bilan birga ishlatiladi.
- Anonim funksiyalar: Funksional dasturlash tilida funksiyalar o'zgaruvchilar sifatida yaratilishi va anonim funksiyalar sifatida ishlatilishi mumkin. Bu, kodning qisqa, amaliyotga yo'naltirilgan yoki filtrlash funksiyalarni yaratishda foydali bo'ladi.

Funksional dasturlashning amaliy masalalarni hal qilishdagi afzalliklari:

- Bo'sh vaqt xatolari kamayadi: Funksional dasturlash tilida o'zgaruvchilar o'zgartirilmaydigan sifatga ega bo'lib, bo'sh vaqt xatolari minimumga tushadi.
- Qo'llanmalar qisqava aniqligi: Funksional dasturlash tilida qo'llanmalar qisqa va aniqligi bilan yaratiladi. Bu, dasturchilarning kodni o'qishini va tahlil qilishini osonlashtiradi.
- Paralell ishlab chiqish: Funksional dasturlash paradigmasi parallel ishlashni osonlashtiradi. Funksiyalar o'zgaruvchilar bilan bog'liq bo'lmaganligi uchun, ular parallel ishlashda xatolar va qisqartirishlarga olib kelmaydi.

Funksional dasturlashning cheklanishlari:

- Yodda tashish: Funksional dasturlash paradigmasida funksiyalar birinchi-klass obyektlar sifatida qo'llaniladi. Bu esa kodni o'qish va tahlil qilishni qiyinlashtirishi mumkin.
- Tizim ustida takrorlanuvchi operatsiyalar: Funksional dasturlashda takrorlanuvchi operatsiyalar uchun o'zgaruvchilar va stavkalarni qo'llash kerak bo'lishi mumkin, bu esa dasturlashni qiyinchilik qilishiga olib kelishi mumkin.
- Yodda bo'lmagan xatolar: Funksional dasturlash paradigmasida o'zgaruvchilarning o'zgartirilmaydigan sifatga ega bo'lishi xatolarni aniqlashni qiyinlashtiradi. Funksional ma'lumotlarni qidirish-Ma'lumotlar bazalarida ma'lumotlarni samarali va ifodali qidirish operatsiyalarini bajarish uchun funksional dasturlashdan qanday foydalanish mumkinligini muhokama qiladi.-Muayyan ma'lumotlar to'plamini ajratib olish uchun yuqori darajadagi funktsiyalardan, xaritalash/kamaytirish operatsiyalaridan va filtrlash usullaridan foydalanishni o'rganadi.
- Funksional dasturlash murakkab so'rov stsenariylarini qanday soddalashtirishi va so'rovlar samaradorligini oshirishi mumkinligiga misollar keltiradi.Funksional dasturlash yordamida ma'lumotlarni manipulyatsiya qilish
- Ma'lumotlar bazalarida ma'lumotlarni manipulyatsiya qilish vazifalarini bajarish uchun funksional dasturlashni qanday qo'llash mumkinligini ko'rsatadi.
- Yangilanishlar va transformatsiyalar paytida ma'lumotlar yaxlitligini ta'minlash uchun o'zgarmaslik va sof funksiyalar kabi usullarni muhokama qiladi.
- Funksional dasturlash ma'lumotlarni tozalash, o'zgartirish va yig'ish kabi vazifalarni qanday soddalashtirishi mumkinligiga misollar keltiradi.Funksional dasturlash bilan tranzaksiyalarni boshqarish-Ma'lumotlar bazasi operatsiyalarida tranzaksiya yaxlitligini ta'minlash

uchun funktsional dasturlash tamoyillaridan qanday foydalanish mumkinligini o'rganadi.

- Tranzaktsiyalarni samarali boshqarish uchun ma'lumotnomaning shaffofligi va o'zgarmasligi kabi usullarni muhokama qiladi.
- Funktsional dasturlash murakkab tranzaksiya stsenariylarini qanday soddalashtirishi va parallellik masalalarini hal qilishi mumkinligiga misollar keltiradi.

Xatolarni boshqarish va chidamlilik.

- Funktsional dasturlash ma'lumotlar bazasi operatsiyalarida xatolarni qayta ishlash va barqarorlikni qanday yaxshilashi mumkinligini ta'kidlaydi.
- Ma'lumotlar bazasi xatolarini oqilona boshqarish uchun monadlardan foydalanish va funktsional xatolarni qayta ishlash kabi usullarni muhokama qiladi.
- Funktsional dasturlash xatolarga chidamlilik va xatolarni tiklash mexanizmlarini qanday oshirishi mumkinligiga misollar keltiradi.

O'zgarmas ma'lumotlar. Funktsional tilda ma'lumotlar o'zgarmasdir, ya'ni siz avval belgilangan qiymatlarni o'zgartira olmaysiz.

Amaliy qism

Funksional dasturlashga kirish uchun, Haskell dasturlash tilini o'rganishingiz juda yaxshi bo'ladi. Haskell funksional dasturlash tilining namoyon namunalar bilan o'rganish juda samarali bo'ladi. Quyidagi namuna orqali, bir nechta asosiy funksiyalarni va amaliy masalalarni yechishni o'rganishingiz mumkin:

1. **Hello, World!** - Oddiy Hello, World! dasturini yozish:

```
main :: IO ()
main = putStrLn "Hello, World!"
```

2. **Fibonacci ketma-ketligi** - Fibonacci ketma-ketligini hisoblash uchun dastur:

```

fib :: Int -> Int
fib 0 = 0
fib 1 = 1
fib n = fib (n-1) + fib (n-2)

main :: IO ()
main = do
    putStrLn "Enter the number to calculate Fibonacci sequence for:"
    num <- readLn
    putStrLn $ "Fibonacci sequence for " ++ show num ++ " is: " ++ show (fib num)

```

3. Sodda sonlar ro'yxati - Berilgan oraliqda sodda sonlar ro'yxatini topish:

```

isPrime :: Int -> Bool
isPrime n | n <= 1 = False
          | otherwise = all (\x -> n `mod` x /= 0) [2..floor (sqrt (fromIntegral n))]

primesInRange :: Int -> Int -> [Int]
primesInRange a b = filter isPrime [a..b]

main :: IO ()
main = do
    putStrLn "Enter the range to find prime numbers (start end):"
    start <- readLn
    end <- readLn
    putStrLn $ "Prime numbers in range " ++ show start ++ " to " ++ show end ++ " are: " ++ show

```

Bu namunalar sizga Haskell dasturlash tili bilan tanishishga yordam beradi va funksional dasturlashga oid asosiy tushunchalarni o'rganish uchun foydalanish mumkin. Yangi vazifalarni yechish va Haskell tilidagi bilimlarini oshirish uchun bu namunalar bilan mashg'ul bo'lishingiz muhim.

Amaliy mashg'ulotni bajarish uchun topshiriqlar:

Talabalar “Funksional dasturlash va unda amaliy masalalarni hal qilish” mavzusini o'rganib, funksional dasturlash va unda amaliy masalalarni hal qilish jarayonlarini tariflaydi va uni hisobot shaklida topshiradi.

Nazorat uchun savollar:

1. Funksional dasturlash va unda amaliy masalalarni hal qilishning asosiy prinsiplarini izohlang.
2. Funksional dasturlash va unda amaliy masalalarni hal qilishda katta hajmli ma'lumotlar bilan ishlash jarayonini tariflang.