

### 3-AMALIY MASHG'ULOT

#### 3-Mavzu: Dasturlash tillari klassifikatsiyasi.

**Ishning maqsadi:** Dasturlash tillari klassifikatsiyasi bo'yicha ko'nikma v amalaka hosil qilish.

#### Nazariy qism

Dastur faqat translyatsiyadan o'tib qolmay, balki ishonchli ishlashi va dastur muallifi bo'lmaganlar uchun ham tushunarli bo'lishi uchun qat'iy amal qilish shart bo'lgan qoidaga e'tiborni qaratish kerak.

**1-ta'rif:** Dasturning *ishonchliligi* – bu dasturning muayyan vaqtda xatosiz ishlashi bilan tavsiflanuvchi xossadir.

**2-Ta'rif:** Dasturning *tushunariligi* - bu dasturning qanday algoritm asosida tuzilganligi va ishlashi foydalanuvchiga tushunarli bo'lishi uchun zarur bo'lgan mahorat asosida baholanuvchi xossadir.

Dasturning tushunariligi va ishonchliligi o'zaro bog'liqdir. Tushunarli dasturda barcha boshqaruvni o'tkazish yo'llari va dasturning yetarlicha ishonchliligini kuzatish qiyin emas. Ishonchlilik bu dasturlarning butunlay xatosiz degani emas. Afsuski, jiddiy qo'llanilmaydigan kam sonli o'quv dasturlargina butunlay xatodan holi bo'lishi mumkin. Zamonaviy dasturlar katta va murakkab bo'lib, ular ustida dasturchilar guruhi ish olib boradi, shundagina dasturlarda xatolar ko'zga tashlanmaydi va bir qancha muddat to'g'ri ishlaydi.

Ishonchlilik va tushunarlilikni ta'minlaydigan usullardan biri bu - strukturali dasturlashdir.

**3- Ta'rif:** *Strukturali dasturlash* - bu uchta mantiqiy (boshqaruvchi) tuzilish: *ketma-ketlik*, *tarmoqlanish*, *takrorlanish* va qismdasturlar orqali dasturlashdir.

Ketma-ketlik tushunchasi  $S_1, S_2, \dots, S_n$  bo'limlarning dastur matnida joylashish tartibi bo'yicha ketma-ket bajarilishini bildiradi.

Tarmoqlanish tushunchasi tanlash amalga oshirilishini bildiradi, ya'ni, B shartning bajarilishiga bog'liq tarzda tanlov aniqlanishini va  $S_1$  yoki  $S_2$  bo'limlardan biri bajarilishini bildiradi.

Takrorlanish tushunchasi takrorlashni bildiradi, ya'ni B shart rost bo'lgan holatda S bo'limning bir necha marta takrorlanishini ifodalaydi. Qachonki, shart bajarilmay qolgan holatda takrorlanish tugatilishini bildiradi. Agarda shart biror marta ham bajarilmasa, u holda S bo'lim umuman ishlamaydi.

Har uchala tushuncha kiritilgan va ularning xossalari isbotlangan. Xususan yagona kirish va chiqish ularning har biri uchun xosdir. Bu xossa dastur tizilmasini o'tish operatori orqali boshqaruvni nisbiy o'tkazishni qo'llashga nisbatan tushunarliroq bo'lishini ta'minlaydi.

Ixtiyoriy dasturni ortiqcha operator va takrorlanishlarsiz strukturali dasturlash orqali amalga oshirish mumkinligi isbotlangan. Bunga esa struktura tuzish orqali erishiladi, masalan, takrorlash bo'limi ichida tarmoqlanish bajarilishi mumkin va h.k. Paskaldagi shart (tarmoqlanishga), takrorlash operatori (takrorlanishlarga) va tarkibli operator (ketma-ketlikka) strukturali dasturlash imkonini beradi.

Ammo ayrim hollarda (masalan, darhol takrorlanishdan chiqish talab qilinganda) strukturali dasturlashdagi qat'iy ketma-ketlik bajarilishi tufayli dasturlar

samarasiz bo'radi. Bunda o'tish operatorini qo'llagan ma'qul, lekin dasturning tushunarliigi kamayadi.

Endi dasturlarni qayta ishlashning qadamli detallashtirishdek mashhur bo'lgan usulini ko'rib chiqamiz. Ma'lumotlarni qayta ishlash masalasi qo'yilgan va uni yechadigan dastur tuzish talab qilingan bo'lsin. Nima qilinadi, oq qog'ozga dastur yozish boshlanadimi? Avval uning strukturali dasturlash shaklidagi sxemasini tasavvur qilish kerak. Bu sxemada asosiy masala tarkibidagi kichik masalalarni alohida yechishga mo'ljallangan 3-5 ta katta bo'limni qo'llash yetarli. Masalan, odatda dasturni bajarilishlar ketma-ketligi deb tasavvur qilinadi:

1. Boshlang'ich ma'lumotlarni kiritish.
2. Ma'lumotlarni qayta ishlash.
3. Natijalarni chiqarish.

Har bir qo'llanilayotgan bo'lim masalaning yanada kichikroq qismlarini yechishga mo'ljallangan qismlarga ajratiladi. Dastur tuzilishini aniqlashtirish jarayoni qo'yilgan masala ma'lum algoritm orqali bajariladigan trivial yechimga ega bo'lgunga qadar davom ettiriladi.

Ushbu jarayon qadamma-qadam detallashtirish deyiladi.

Keyigi bosqichda dastur tuzishning kodlashtirish jaryoni – qadamli detallashtirish natijasida hosil qilingan yozuvdagi dasturni dasturlash tilida shakllantirish bajariladi.

Qadamli detallashtirishda dastur tuzilishini blok-sxema shaklida yoki psevdokodlar bilan tasvirlash mumkin. Psevdokod-algoritmik tillarga yaqin bo'lib, translyatsiya qilinmaydi. Unda ham xizmatchi so'zlar ishlatiladi, dasturlash tili bo'lmagan so'zlashuv tiliga yaqin bo'lgan iboralardan iborat psevdoooperatorlardan foydalaniladi.

## II. Amaliy qism.

Misol uchun ayrim yig'indini hisoblash masalasini ko'raylik:

$$s(n)=1/(1!)+1/(2!)+...+1/(n!)$$

1- bosqich detallashtirish qadami

begin

    Boshlang'ich ma'lumotlarni kiritish

    Xususiy yig'indilarni hisoblash  $s(n)$ .

    Natijalarni chiqarish.

end.

2-Bosqich detallashtirish qadami

var  $n, i$ :integer;  $s$ :real;

{Boshlang'ich berilganlarni kiritish.}

writeln(' n ni kiriting:');

read( $n$ );

{Xususiy yig'indilarni hisoblash  $s(n)$ .}

$i:=1$ ;

$s:=0$ ;

while ( $i \leq n$ ) do

    begin

        Navbatdagi qo'shiluvchini hisoblash.

        ( $i$ ).

```

        Hisoblagichning oshishi.
    end;
{Natijalarni chiqazish.}
writeln('s(n)= ', s:10:6);
end.

```

### 3-Bosqich detallashtirish qadami

```

var n,i:integer; s,x:real;
{Boshlang'ich berilganlarni kiritish.}
writeln(' n ni kiriting:');
read(n);
{Xususiyl hosilani hisoblash s(n).}
i:=1;
s:=0;
x:=1;
while (i<=n) do
    begin
        {Navbatdagi qo'shiluvchini hisoblash.}
        x:=x/i;
        {Navbatdagi xususiyl yig'indini hisoblash s(i).}
        s:=s+x;
        {Hisoblagichni oshirish.}
        i:=i+1
    end;
{Natijalarni chiqarish.}
writeln('s(n)= ', s:10:6);
end.

```

Dasturni aniqlashtirishda psevdokod operatorlari o'rniga Paskal operatorlari ishlatilgan, ularni esa izohga kiritilgan.

### TOPSHIRIQ

1. Dasturlash tillari qanday darajalarga bo'linadi?
2. Dasturlash tillarining funksional imkoniyati
3. Dasturlash paradigmasi va metodologiyasi.
4. Masalalar tipi bo'yicha klassifikatsiyalash.
5. Dastlabki yuqori darajali dasturlash tillari qaysi nomlarini va ularning yaratilish tarixini ayting.
6. Algoritmik tillar deganda nimani tushunasiz va bu tillar tabiiy til hisoblanadimi yoki sun'iy til?
7. Algoritmik tillar qanday komponentalardan tashkil topgan?
8. Strukturali dasturlash nima?
9. Strukturali dasturlash qanday mantiqiy qismlarda tashkil topgan bo'lishi mumkin?
10. Ob'yektga yo'naltirilgan dasturlashning ma'nosini tushuntiring.
11. Ob'yektga yo'naltirilgan dasturlash tillariga misollar keltiring.