



C++ “Loop”

Ծրագրավորման “Loop” վերահսկման կառուցվածք է, որը թույլ է տալիս մի շարք հրահանգներ կամ հայտարարություններ բազմիցս կատարել՝ հիմնվելով պայմանի վրա: Այս պայմանը սովորաբար բուլյան (boolean) արտահայտություն է, և քանի դեռ պայմանը գնահատվում է true, loop-ը շարունակում է գործել:

Loop-ների տեսակները C++-ում

- **“for”** - for loop-ն օգտագործվում է կողի բլոկը որոշակի քանակությամբ անգամ գործարկելու համար: Այն բաղկացած է սկզբնավորումից, պայմանից և թարմացման հայտարարությունից:

```
for (initialization; condition; update) {  
    // Code to be executed  
}
```

- **“while”** - while բազմիցս կատարում է կողի բլոկ, քանի դեռ նշված պայմանը ճշմարիտ է:

```
while (condition) {  
    // Code to be executed  
}
```

“do-while” - do-while նման է while-ին, սակայն այն երաշխավորում է, որ կողի բլոկը կկատարվի առնվազն մեկ անգամ՝ նախքան պայմանը ստուգելը:

```
do {  
    // Code to be executed  
} while (condition);
```

Ինչպես են Loops-ը գործում

Loop-ի գործարկումը ներառում է հետևյալ հիմնական քայլերը.

1. **Նախնականացում.** Փոփոխականները սկզբնավորվում են, հաճախ մինչև մեկնարկային արժեք:
2. **Պայմանների ստուգում.** հանգույցի վերնագրում նշված պայմանը գնահատվում է: Եթե պայմանը ճշմարիտ է, ապա հանգույցի մարմինը կատարվում է: Եթե կեղծ է, ապա հանգույցն ավարտվում է:
3. **Loop-ի մարմնի կատարում:** Կոդը կատարվում է loop-ի մարմնի ներսում:
4. **Թարմացում.** loop-ի մարմինը գործարկվելուց հետո կատարվում է թարմացման հայտարարությունը (օրինակ՝ փոփոխականի ավելացում կամ նվազեցում):
5. **Վերադարձ դեպի Քայլ 2.** Ծրագիրը վերադառնում է քայլ 2 և նորից գնահատում է loop-ի վիճակը: Եթե պայմանը դեռ ճշմարիտ է, ապա հանգույցի մարմինը նորից գործարկվում է, և գործընթացը կրկնվում է:

Օգտագործման դեպքեր և loop-ի նշանակություն

1. Կրկնվող գործողություններ

Loop-ը օգտագործվում են, երբ դուք պետք է մի քանի անգամ կատարեք նույն գործողությունը կամ գործողությունների մի շարք: Օրինակ՝ ցուցակի տարրերի մշակումը, ֆայլից տողերի ընթերցումը կամ մի շարք արժեքների հաշվարկը:

2. Տվյալների կառուցվածքների կրկնում

Loop-ը կարևոր են զանգվածների, վեկտորների, կապակցված ցուցակների, ճյուղերի և տվյալների այլ կառուցվածքների մեջ տարրեր անցնելու և գործելու համար:

3. Մուտքի վավերացում

Loop-ը օգտագործվում են օգտատիրոջը մուտքագրելու համար բազմիցս առաջարկելու համար, մինչև վավեր տվյալներ չտրամադրվեն՝ ապահովելով, որ ծրագիրը շարունակվի համապատասխան տեղեկություններով:

4. Մոդելավորում

Իրական գործընթացների մոդելավորման ժամանակ Loop-ներն օգտագործվում են ժամանակի ընթացքում կամ որոշակի պայմաններում բազմիցս տեղի ունեցող գործողությունների մոդելավորման համար:

5. Ալգորիթմի իրականացում

Շատ ալգորիթմներ, ինչպիսիք են տեսակավորումը և որոնումը, պահանջում են տվյալների վրա կրկնվող գործողություններ: Loop-ը հիմնարար են այս ալգորիթմների իրականացման համար:

6. Խաղերի մշակում

Խաղերի ծրագրավորման մեջ Loop օգտագործվում են խաղի վիճակները թարմացնելու, օգտվողի մուտքերը կարգավորելու և խաղի իրադարձությունները կառավարելու համար, որոնք բոլորն էլ տեղի են ունենում շարունակական loop-ում:

Loop-ը արդյունավետ և ճկուն ծրագրավորման անկյունաքարն է, որը թույլ է տալիս ավտոմատացնել կրկնվող առաջադրանքները և ծրագրերը դարձնել ավելի հարմարվող տարբեր մուտքերի և սցենարների նկատմամբ: Օղակներն արդյունավետ օգտագործելու մասին հասկանալը կարևոր է արդյունավետ և ֆունկցիոնալ C++ ծրագրեր գրելու համար: