



Ծանոթացում «Switch» օպերատորին:

C++-ում անջատիչի հայտարարությունը կառավարման հոսքի կառուցվածք է, որն օգտագործվում է մի քանի այլընտրանքների միջև ընտրության համար՝ հիմնված արտահայտության արժեքի վրա: Այն ապահովում է բազմաթիվ գործեր վարելու արդյունավետ միջոց՝ առանց մի շարք if-else հայտարարությունների դիմելու:

Ահա C++-ում switch-ի հայտարարության մանրամասն բացատրությունը.

switch (expression) {

case constant1:

*// Կող, որը պետք է կատարվի, եթե արտահայտությունը համընկնում է հաստատուն1-ի հետ
break;*

case constant2:

*// Կող, որը պետք է կատարվի, եթե արտահայտությունը համընկնում է հաստատուն2-ի հետ
break;*

// Անհրաժեշտության դեպքում ավելի շատ հայտարարություններ...

default:

*// Կողը պետք է կատարվի, եթե արտահայտությունը չի համապատասխանում որևէ հաստատուն
break;*

}

- Արտահայտությունը սովորաբար ինտեգրալ կամ թվարկված տիպի է: Դրա արժեքը համեմատվում է գործի պիտակներում նշված հաստատունների հետ:
- Յուրաքանչյուր դեպքի պիտակ ներկայացնում է արտահայտության հնարավոր արժեքը:
- Նախնական պիտակը կամընտիր է և գործարկվում է, եթե արտահայտությունը համընկնում գործի հաստատուններից որևէ մեկին:

Switch Քաղվածքի կատարման հոսք:

1. Արտահայտությունը գնահատվում է.
2. Արտահայտության արժեքը համեմատվում է յուրաքանչյուր դեպքի հաստատունի հետ:
3. Եթե համընկնում է գտնվել, կողերի համապատասխան բլոկը գործարկվում է, և հաջորդ ծածկագիրը նույնպես գործարկվում է մինչև ընդմիջման հայտարարությունը:
4. Եթե համընկնում չի գտնվել, և առկա է լռելյայն պիտակ, ապա գործարկվում է լռելյայն հետ կապված կողի բլոկը:
5. Եթե համընկնում չի գտնվել և չկա լռելյայն պիտակ, ոչ մի կող չի կատարվում, և կառավարումն անցնում է անջատիչի բլոկին հաջորդող հայտարարությանը:

Break հայտարարություն:

- Break հայտարարությունը օգտագործվում է switch բլոկից դուրս գալու համար:
- Երբ հանդիպում է break հայտարարություն, հսկիչը դուրս է գալիս switch բլոկից, և հաջորդ case պիտակները չեն կատարվում:

Switch հայտարարությունը հզոր գործիք է ծրագրային հոսքը վերահսկելու համար, որը հիմնված է արտահայտության արժեքի վրա, կողն ավելի ընթերցանելի և արդյունավետ դարձնելով, հատկապես, երբ գործ ունենք բազմաթիվ տարբերակների հետ: