

Explicație - Operator

Petculescu Mihai-Silviu

Explicație - Operator

Petculescu Mihai-Silviu

1. În programul următor, supradefiniți operatorul `<<` astfel încât `cout<<i` să afișeze valoarea atributului `i.x`. Precizați și explicați rezultatele afișate la executarea programului astfel obținut.

1. În programul următor, supradefiniți operatorul `<<` astfel încât `cout<<i` să afișeze valoarea atributului `i.x`. Precizați și explicați rezultatele afișate la executarea programului astfel obținut.

Rezultat

```
0
2
2
0
1
```

Clasa C supraîncarcă operatorii unari de stânga, `++` și `--`, cu diferența ca pentru `operator++()` returnează adresa instanței cu referință `&i`, în vreme ce `operator--()` returnează doar pointerul.

În `main()` creăm o instanță a clasei C alocată variabilei `i`, apoi îi afișăm valoarea, prin supraîncărcarea operatorului `<<`, care by default este 0.

Linia `cout<<++(++i)<<endl<<i<<endl;` inclementează valoarea lui `i` cu două unități, înainte de a fi accesată de operatorul `<<`, astfel la final se afișează de două ori valoarea 2.

Linia `cout<<--(--i)<<endl<<i<<endl;` are în mare, aceeași efect ca și linia anterioară, cu precizările ca în cazul acesteia valoarea variabilei `i` descrește. Totuși, la final se afișează, cum se poate observa mai sus 0 urmat cu 1, asta deoarece operatorul nu returnează variabila prin referință, astfel doar una dintre scăderi este înregistrată de `i`.

Footnote: Pentru a diferenția operatorii unari plasați la începutul unei variabile cu cei plasați la finalul aceleiași variabile, C++ folosește ca diferențiator un parametru formal de tip `int`. Astfel funcția `operator++()` va supraîncărca operatorul `++x`, în vreme ce `operator++(int)` va supraîncărca operatorul `x++`. Analog pentru `--`.