

Lvrednosti

- Operator `&` zahteva operand koji jeste lvrednost, ali mu rezultat nije lvrednost. Nasuprot tome, operator indirekcije `*` ne zahteva da operand bude lvrednost, ali mu rezultat jeste lvrednost:

`&a` — Ispravno pod uslovom da je *a* ime varijable ili funkcije, što jeste lvrednost

`&a = ...` — Greška u prevođenju, jer levi operand operatora `=` nije lvrednost

`&(a+3)` — Greška u prevođenju, jer operand operatora `&` nije lvrednost

`*(p+3)` — Ispravno, jer operator `*` ne zahteva da operand jeste lvrednost, dok mu rezultat jeste lvrednost

`*(p+3) = ...`

- Operator `a<<b` daje rezultat koji predstavlja binarnu vrednost celobrojnog operanda *a* pomerenu za *b* bita ulevo. Kako znati da li taj operator proizvodi bočni efekat ili ne, tj. da li menja (pomera) svoj levi operand? Ako ga menja, taj operand bi morao biti lvrednost, ali ovaj operator ne zahteva da levi operand bude lvrednost, pa onda sigurno ne proizvodi bočni efekat

- Rezultat poziva funkcije je lvrednost akko funkcija vraća referencu na lvrednost, na primer:

`int& f ();` — Funkcija *f* vraća referencu na lvrednost, pa rezultat poziva te funkcije jeste lvrednost

`f() += 1;` — Rezultat poziva funkcije *f* je referenca koja upućuje na neki objekat tipa *int*

`X& X::operator+=(int);` — Ova operatorska funkcija klase *X* ima parametar tipa *int* i vraća referencu na lvrednost

`X x;`

`X* p = &(x += 3);` — Za operaciju `x+=3` poziva se `x.operator+=(3)`. Ova operatorska funkcija vraća referencu na neki objekat tipa *X* i taj rezultat jeste lvrednost, pa može biti operand operatora `&`

Lvrednosti

- ❖ Termin “lvrednost” ima poreklo u tome što “može da stoji sa leve strane znaka dodele”, ali to ne važi za svaku lvrednost: samo *promenljive lvrednosti* (*modifiable lvalues*) mogu da stoje sa leve strane znaka dodele (i složene dodele)
- ❖ Ime niza, funkcije ili konstatnog objekta nije promenljiva lvrednost, pa ne može stajati sa leve strane operatora dodele ili složene dodele:

```
int a[5], f();
```

```
const int c = 3;
```

```
a = ...
```

```
f = ...
```

```
c = ...
```

```
&a
```

```
&f
```

```
&c
```

- ❖ Pojam lvrednosti je na jeziku C++ upravo značajan zbog toga što postoji mogućnost korišćenja objekata klasa po vrednosti, preklapanja operatora za korisničke tipove (klase), sa notacijom kao za ugrađene tipove i njihove operande; blisko povezan sa svim tim jeste i pojam reference na lvrednost, koja je i uvedena sa ciljem podrške takvom korišćenju, ali je sve to i značajno zakomplikovalo jezik