



THIS PAGE IS  
INTENTIONALLY  
LEFT BLANK.

Kazimir Majorinc

# Landin i LISP

Povijest LISPa 32.



Razmjena vještina  
Hacklab u mami  
15. lipnja 2013.

P. Landin, *The next 700 programming Languages*, C. ACM,  
March 1966

## **ISWIM i imperativni aplikativni izrazi**

**Landin** predstavlja svoj „sistem“, zapravo, programski jezik ISWIM (If you See What I Mean).

Sintaksa jezika jedva da je izložena - na primjerima. To nije slučajno. Landin definira nekoliko nivoa apstrakcije ISWIMa, krenuvši od „fizičkog ISWIMa“, programi kakvi ih vidimo zapisane na ekranu ili papiru, uključujući i boje, fontove - do najapstraktnijeg, imperativnih aplikativnih izraza (IAE).

Fizički ISWIM nije bitan, dapače, moguće je definirati različite fizičke ISWIME, ovisno o problemu koji se rješava. Otud dolazi naslov članka. ISWIM je familija jezika koji se „prevode“ u IAE.

(Većina jezika imaju različite implementacije i moglo bi se izbjeći nepotrebne rasprave da ih se od početka naziva familijama jezika.)

Landin je, tvrdi, implementirao ISWIM i koristio ga za različite svrhe, uključujući i „komercijalno programiranje.“ (???)

# ISWIM - rezime prethodnih izlaganja

Suštinu ISWIMa već znamo. ISWIM podržava četiri osnovne vrste izraza:

ISWIM	meta-izrazi
$x$	$x$
$\lambda x.x+1$	$\lambda[[x]; x+1]$
$\{\lambda x.x+1\}[3]$	$\lambda[[x].x+1][3]$
$z \Leftarrow f(x)$	$z := f[x]$

Dodatno,

where-izraze i zatvorenja:  $x(x+1)$  **where**  $x = 3$

and-izraze (istovremeno izvođenje)  $a \Leftarrow b$  **and**  $b \Leftarrow a$ .

programska zatvorenja

# Karakteristične ekvivalencije

U većini programskih jezika postoje tvrdnje o ekvivalenciji različitih programa koje nisu baš potpune. Npr. **for + goto** vs. **while-do** vs **repeat-until**. Landin smatra da su tvrdnje o ekvivalenciji iznimno važne jer omogućuju različite vrste procesiranja programa (optimizacije, ubacivanje testova).

Cijelo poglavlje u članku je posvetio ekvivalencijama u ISWIMu. Primjerice,

**true**  $\rightarrow$  M; N je ekvivalentno M

**false**  $\rightarrow$  M; N je ekvivalentno N

# Odnos ISWIMa i LISPa.

Landin je posvetio cijelo poglavlje usporedbi ta dva jezika.

„ISWIM can be looked on as an attempt to deliver LISP from its eponymous commitment to lists, its reputation for hand-to-mouth storage allocation, the hardware dependent flavor of its pedagogy, its heavy bracketing, and its compromises with tradition.“

(1) ISWIM ne naginje prema nekoj posebnoj problemskoj orijentaciji. Postojala je eksperimentalna implementacija i ISWIM je korišten za numeričko procesiranje, „komercijalno programiranje i drugo.“

(2). ISWIM umanjuje potrebu za garbage collectorom. Memorija korištena u where-izrazima može biti odmah prikupljena.

append(L,L) **where** L=(1,2,3,4).

(3) „Tamni uglovi“ LISPa - procesiranje memorijskih adresa i povezano crtanje dijagrama. ISWIM postiže veću nezavisnost od hardvera uz pomoć „apstraktne mašine.“ (???)

(4) Tekstualna reprezentacija LISPa je loša („heavy bracketting“). ISWIM ima standardnu sintaksu sa infix operatorima, obogaćenu pomoćnim definicijama (**where**, **let**, **and**, **rec**). Koristi uvlačenje (kao Python).



(5) „Najvažniji doprinos LISPa je u logičkim svojstvima koja leže iza notacije. ISWIM tu uvodi značajna poboljšanja, jer LISP nije ostavio mjesta za njih. Ta se tvrdnja može izreći i na slijedeći način.

(a) LISP pojednostavljuje relaciju ekvivalencije koja određuje koji dijelovi programa se mogu međusobno zamijeniti, bez da se promijeni rezultat.

(b) LISP omogućuje pisanje izraza koji su bliži onima koji se pojavljuju u fizikalnim, matematičkim i logičkim sistemima. “

## Landin M.I.T.

Landin se 1966. zaposlio na M.I.T.

„He was used to sharing his ideas with his colleagues ... but in Cambridge Mass. people kept their ideas to themselves. They would not listen to him very much: in particular they did not take to his ‘closure’ idea, and as a result LISP, the first language to deal with list processing and functional values, got functions wrong in all its implementations for a decade and a half. Peter remarked that “the LISP they understood and the LISP I understood had little to do with one another”.

**Bornat, *Peter Landin: a computer scientist who inspired a generation*, 5th June 1930 - 3rd June 2009.**

# Zaključak

**Landinov** ISWIM uvodi niz poboljšanja u odnosu na tadašnji LISP. Unatoč prvotne nezainteresiranosti, Landinove ideje (osim sintakse) su kasnije našle mjesto u dijalektima LISPa, posebno u Scheme.

**Landin**, međutim, uopće ne diskutira S-izraze i mogućnost prevođenja programa u S-izraze, što pojednostavljuje ne samo meta-programiranje, nego i razvoj jezika.

