Historiaa

- 1954 IBM aloitti automaattiseen koodin generointiin tähtäävän projektin
- 1957 ensimmäinen Fortran-kääntäjä
    - ei vielä kunnollisia kontrollirakenteita

```
GOTO 100
IF (I-J) 100,200,300
```

- 1958 Fortran II
    - erillisinä käännettävät aliohjelmat
- 1960-luvulla eri valmistajien kääntäjiä, joissa erilaisia laajennuksia
- 1966 ANSI-standardoitu Fortran 66
- 1960- ja 1970-luvuilla kääntäjäteoria ja ohjelmointikielten tutkimus kehittyivät voimakkaasti (Algol, Pascal ym.)
- 1978 standardoitu Fortran 77
    - if .. then .. else -rakenne
    - merkkijonomuuttujat
    - kehittyneempi syöttö ja tulostus
- 1991 Fortran 90:n ISO-standardi ISO/IEC 1539:1991
    - hyvin paljon laajennuksia nykyaikaisempien kielten suuntaan (Ada)
    - laiteriippumattomampi muuttujien määrittely, omat tyypit mahdollisia
    - taulukko-operaatiot
    - modulirakenne
- 1992 ANSI-standardi ANSI X3.198-1992
- 1997 Fortran 95:n standardi ISO/IEC 1539:1997
- Fortran 2000 ?
- HPF (High Performance Fortran), F90:n laajennus rinnakkaislaskentaan

Yleistä ohjelmointikielistä

Yksinkertaiset muuttujat

Pienin suoraan osoitettavissa oleva yksikkö yleensä 8 bitin tavu (poikkeuksena sanakoneet, kuten Cray).

Kokonaisluku (integer) tavallisesti 2 tavua. Kokonaisluvun itseisarvo voi olla korkeintaan $2^{15} - 1 = 32\ 767$.

Usein käytettävissä myös kaksoistarkkuuden kokonaisluku, 4 tavua, jolloin suurin esitettävissä oleva luku on $2^{31} - 1 = 2\ 147\ 483\ 647$.

Reaaliluku (real) tavallisesti 4 tavua

- mantissan etumerkki
- mantissa normitettu välille 0.1 – 1
- eksponentti: etumerkki+itseisarvo tai 2:n komplementti

Esim. PC:n yksinkertaisen tarkkuuden reaaliluvun mantissa on 23 bittiä. Desimaalipisteen jälkeinen bitti on aina ykkönen, joten sitä ei tarvitse tallettaa. Tarkkuus on 24 bittiä eli $\log_{10} 2^{24} \approx 7$ desimaalia. Eksponenttiosan pituus 8 bittiä. Eksponenttiosa $e = 127$ (bias) + todellinen eksponentti; $1 \leq e \leq 254$. Lukualue $2^{-126} \approx 10^{-38}$ – $2^{128} \approx 3 \times 10^{38}$.

Kaksoistarkkuus: merkitsevien numeroiden määrä noin kaksinkertainen ja mahdollisesti myös lukualue laajempi. Esimerkiksi PC:n kaksoistarkkuuden muuttujien tarkkuus on noin 15 desimaalia ja lukualue noin $10^{-308} \ldots 10^{308}$.

Looginen muuttuja (logical) voi saada vain arvot tosi tai epätosi.

Osoitin (pointer)

- osoitetun muuttujan osoite muistissa + muuta tietoa

Yksinkertaisista muuttujista voidaan muodostaa mutkikkaampia tietorakenteita.

Taulukot (array)

- joukko samantyyppisiä muuttujia
- yleensä peräkkäin muistissa
- alaraja riippuu kielestä: C: 0, ei voi muuttaa
  Fortran: 1, voi määritellä vapaasti (F77, F90)
  Pascal: määriteltävä, ei oletusarvoa
- yläraja annetaan yleensä määrittelyssä
- indeksin tarkistus riippuu kielestä ja toteutuksesta. C, Fortran: ei yleensä tarkisteta

Tietueet (record, structure)

- joukko muuttujia, jotka voivat olla eri tyyppisiä
- omat tehtävään liittyvät abstraktit tietotyypit

Pino (stack)

- LIFO (Last In First Out) (vrt. HP:n laskimet)
- käytetään mm. paikallisten muuttujien talletukseen

Keko (heap)

- muistialue, josta varataan esim. taulukoiden tila

Kontrollirakenteet

1 Peräkkäisyys

```
S1
S2
S3
```

2 Valinta (1, 2 tai useita vaihtoehtoja)

```
if ehto then S1
if ehto then S1 else S2
case ehto { S1; S2; ... }
```

3 Toisto

```
while ehto do S
do S until ehto
for i=1,n do S
```

Kaikki ovat erikoistapauksia ns. $n + 1/2$ kierroksen silmukasta

```
do
{
 S1
 if ehto exit
 S2
}
```

Kaikki ohjelmointitehtävät voidaan toteuttaa em. rakenteilla.

Useissa kielissä lisäksi hyppykäsky (go to)

- yleensä tarpeeton
- sopii lähinnä poikkeustilanteisiin
- vaikeuttaa ohjelman luettavuutta
- kontrollireitit mutkistuvat
- osoitteesta ei näy, mistä siihen on tultu

Aliohjelmat

- tehtävän jäsentely pienempiin ja helpommin hallittaviin osiin
- toiston välttäminen
- testauksen helpottaminen
- aliohjelmakirjastot