

# 8101000 Ohjelmointikielten periaatteet

Tentti 3.10.2005

1. Selitä lyhyesti seuraavien termien merkitys ja anna tarvittaessa myös esi- (6 p.)  
merkit:

- (a) muuttujan elinaika
- (b) invariantti
- (c) dynaaminen näkyvyys
- (d) moninimisyys
- (e) geneerinen aliohjelma
- (f) ääretön tietorakenne

2. Pascalin ehdollinen lause on muotoa (6 p.)

```
<Statement> ::= <IfStatement> | ...  
<IfStatement> ::= if <Expression> then <Statement> <ElsePart>  
<ElsePart> ::= else <Statement>  
<ElsePart> ::=
```

missä <Expression> kuvaa lausekkeen, mutta sen tarkemmasta rakenteesta ei tarvitse välittää. <ElsePart>:n toisen säännön oikea puoli on tyhjä, mikä tarkoittaa, että <ElsePart> voi tuottaa pelkän tyhjän merkkijonon. Onko tämä kielioppi yksikäsitteinen? Perustele. Jos kielioppi ei ole yksikäsitteinen, miten siitä saataisiin yksikäsitteinen?

3. Selvitä, mitä tarkoitetaan tyyppien rakenne-ekvivalenssilla ja nimiekvivalenssilla. Tarkastele alla olevia esittelyitä. Millä muuttujista A, B, C, D ja E on sama tyyppi nimiekvivalenssin mielessä ja millä rakenne-ekvivalenssin mielessä? (6 p.)

```
type T1 = array [ 1..10 ] of 1..10;  
      T2 = array [ 1..10 ] of Integer;  
      T3 = array [ 2..11 ] of Integer;  
var A: T1;  
    B: T2;  
    C: T3;  
    D, E: array [ 1..10 ] of Integer;
```

4. Vertaile toistorakenteen ja rekursion eroja tehokkuudessa. Mistä erot pohjimmiltaan johtuvat. (6 p.)

5. Millä eri tavoilla monimuotoisuutta (polymorphism) voi esiintyä ohjelmointikielissä? (6 p.)