

**81141 Ohjelmointikielten periaatteet/Principles of Programming Languages  
Tentti/Examination 18.12.2000**

1. Virhetarkistuksia voidaan tehdä (1) leksikaali- ja syntaksianalyysissä, (2) semanttisessa analyysissä tai (3) ohjelmaa ajettaessa, taikka (4) kattava tarkistus ei ole käytännössä mahdollista. Olettaen, että kieli on hyvin suunniteltu ja toteutettu, mihin luokkaan seuraavat tarkistukset kuuluvat:

- (a) aliohjelmakutsussa parametrien lukumäärä on oikea,
- (b) sijoituslauseen molemmat puolet ovat samaa tyyppiä,
- (c) taulukkoindeksi on sille annettujen rajojen välissä,
- (d) mikään osoitin ei jää osoittamaan "heapistä" deallokoitavaan tietoon,
- (e) jokaista alkusulkuu vastaa oikeanlainen loppusulku,
- (f) kaikki lausekkeissa käytetyt muuttujat on esitelty.

2. Selitä ohjelmasilmukkaan

```
{P}          -- P on esiehto
while b loop -- b on totuusarvoinen lauseke
  S;         -- S on jono suoritettavia lauseita
end loop;
{Q}          -- Q on jälkiehto
```

liittyen, mitä silmukkainvariantilla tarkoitetaan. Missä ohjelman kohdassa/kohdissa sen tulee olla tosi? Miten todistetaan, että jos silmukkaan tullaan esiehdolla  $P$ , niin se joskus päättyy, ja että jälkiehto  $Q$  on tällöin tosi?

3. Anna esimerkki, missä aktivaatietietueen staattinen ja dynaaminen linkki eivät ole samat.
4. Minkälaisista eri syistä poikkeus voi virittyä? Mitä tapahtuu tämän jälkeen? Mistä ohjelman suoritusta jatketaan poikkeuksen käsittelyn jälkeen?

- 
1. Error checking can take place in (1) lexical and syntax analysis, (2) semantic analysis, (3) program execution, or (4) cannot, in general, be done in practice. To which category does each of the following checks belong:

- (a) the number of parameters is correct in procedure calls,
- (b) each side is of the same type in assignment statements,
- (c) an array index is between the given bounds,
- (d) no pointer remains pointing to data that is deallocated from the heap,
- (e) for each opening parenthesis there is a corresponding closing parenthesis,
- (f) all variables in expressions have been declared.

2. Explain the notion of a loop invariant in terms of the following loop:

```
{P}          -- P is a precondition
while b loop -- b is a Boolean expression
  S;         -- S is a sequence of statements
end loop;
{Q}          -- Q is a postcondition
```

In which place(s) should it hold? How do you prove that, if the loop is entered with precondition  $P$  true, it will eventually terminate with postcondition  $Q$  true?

3. Give an example where the static (access) and dynamic (control) links of an activation record are different.
4. What are the different kinds of reasons for an exception to be raised? What happens then? From where is execution continued after exception handling?