

1. Selvitä osoittimiin liittyvien staattisten ja dynaamisten tarkistusten mahdollisuuksia hyvin suunnitellussa kielessä.
2. Selvitä silmukkainvariantin idea seuraavan (harjoitustehtävässäkin esiintyneen) ohjelmasilmukan valossa, joka laskee Fibonaccin luvun $\text{fib}(n)$, $n \geq 0$:

```
k := 0;
a := 0;
b := 1;
while k < n loop
  b := a + b;
  a := b - a;
  k := k + 1;
end loop;
```

Liitä tähän silmukkaan esiehto, silmukkainvariantti ja jälkiehto. Mitä kaikkea pitää todistaa, jotta voitaisiin vakuuttaa tämän silmukan oikeellisuudesta?

3. Selitä, mitä turvallisuuden kannalta huomattavaa on Adan seuraavissa piirteissä:
 - (a) vaihtoehdotiset kentät tietuetyypissä,
 - (b) valintalause (case-rakenne lauseitten tasolla),
 - (c) toistomuuttujaa käyttävä silmukkalause.
 4.
 - (a) Mitä tarkoitetaan ohjelmointikielen syntaksilla ja staattisella semantiikalla?
 - (b) Mitä eroa on (aliohjelman) kuormituksella (overloading) ja dynaamisella sidonnalla?
 - (c) Mikä on staattisen ja dynaamisen linkin tarkoitus aktivaatietietueessa?
-

1. Analyze the possibilities of static and dynamic checks in connection with pointers, assuming that the language is well designed
2. Explain the idea of loop invariants in the light of the following example for computing Fibonacci numbers $\text{fib}(n)$, $n \geq 0$:

```
k := 0;
a := 0;
b := 1;
while k < n loop
  b := a + b;
  a := b - a;
  k := k + 1;
end loop;
```

Attach a precondition, a loop invariant, and a postcondition to this loop. What needs to be proved to get convinced about the correctness of this loop?

3. Explain why the following features in Ada are safe:
 - (a) variant fields in record types,
 - (b) case statement,
 - (c) loop with a looping variable.
4.
 - (a) What is meant by the terms syntax and static semantics?
 - (b) What is the difference between overloading and dynamic binding (of a subroutine)?
 - (c) What is the purpose of static and dynamic links in activation records?