

## **OHJ-3100 Ohjelmien ylläpito ja evoluutio**

Tentti 28.1.2008

1. Mitä tarkoitetaan seuraavilla termeillä? Anna myös esimerkki kuhunkin. (6 p.)
  - (a) Evoluutiolaki
  - (b) Takaisinmallinnusmalli
  - (c) Oliointi
2. Tarkastele tehtäväpaperin toisella puolella olevaa C-ohjelmaa ja sovelle siihen viipalointia seuraavasti: (6 p.)
  - (a) Staattinen takautuva viipale muuttujan `nw` ja rivin 26 perusteella. (Riittää, että ilmoitat viipaleeseen tulevien rivien numerot.)
  - (b) Staattinen takautuva viipale muuttujan `nl` ja rivin 26 perusteella. (Riittää, että ilmoitat viipaleeseen tulevien rivien numerot.)
  - (c) Mitä viipaleiden koodit tekevät?
3. Tehtävinäsi on mitata (6 p.)
  - (a) koodin kompleksisuutta
  - (b) ohjelman ylläpidettävyyttä.

Miten etenet tehtävissäsi? Miten tehtävät suhtautuvat toisiinsa?
4. Tehtävänäsi on tietojärjestelmän uudistaminen, jossa tarvitaan sekä koodin että datan uudistamista. Miten pystyt sovittamaan koodin ja datan uudistamisen toisiinsa? (6 p.)
5. Tarkastellaan uudelleenkäytettävien komponenttien muodostamaa komponenttikirjastoa. Millaisia ominaisuuksia komponenteilta vaaditaan ja miten komponenttikirjasto kannattaa rakentaa, jotta uudelleenkäyttö onnistuisi? Toisaalta, mitä esteitä voit tunnistaa uudelleenkäytölle, ja miten voit poistaa esteitä? (6 p.)

**Käännä**

Tehtävään 2 liittyvä koodi:

```
(1) #define YES 1
(2) #define NO 0
(3) main (
(4) {
(5)     int c;
(6)     int nl = 0;
(7)     int nw = 0;
(8)     int nc = 0;
(9)     int inword = NO;
(10)    c = getchar ( );
(11)    while ( c != EOF ) {
(12)        nc = nc + 1;
(13)        if ( c == '\n' )
(14)            nl = nl + 1;
(15)        if ( c == ' ' || c == '\n' || c == '\t' )
(16)            inword = NO;
(17)        else if ( inword == NO ) {
(18)            inword = YES;
(19)            nw = nw + 1;
(20)        }
(21)        c = getchar ( );
(22)    }
(23)    printf ( "%d \n", nl );
(24)    printf ( "%d \n", nw );
(25)    printf ( "%d \n", nc );
(26) }
```