

..... Tehtävät 3. & 4. omalle paperilleen! Nimi paperiin! .....

3. Vastaa seuraaviin ranskalaisilla viivoilla ja lyhyillä lauseilla. Pohdi kysymystä ja piidä huoli siitä että kaikki kysymykseen liittyvät asiat ja avainsanat tulevat mainittua!
- a) Miten *yleiskäyttöisen ja geneerisen* ohjelmakomponentin suunnittelu eroaa tavallisen tiettyyn tarkoitukseen tarkoitettujen komponenttien suunnittelusta. (4 p)
  - b) Dynaaminen sitominen on tärkeä ominaisuus periytyemisessä. Mitkä ovat dynaamisen sitomisen hyvät puolet (=käyttökohteet) luokkahierarkiassa, mitkä taas sen haittapuolet? (4 p)
4. Seuraavassa on joukko väittämiä olio-ohjelmoinnista ja C++:sta. Mitkä väittämät ovat oikein, mitkä väärin? Perustele mielestäsi vääristä väittämistä n. 3–6 rivillä, *miksi/miten* väittäjä on väärin ja miten asia todellisuudessa on. (6 p)
- a) C++:n toteutusmalleissa (*template*) tyyppiparametriksi käyvät mitkä tahansa tyytit, joilla ko. toteutusmallin koodi kääntyy.
  - b) Abstraktit kantaluokat (*abstract base classes*) ovat luokkia, joista ei enää voi periyttää aliluokkia.
  - c) Kun funktion paluutyyppi on luokka, välitetään funktion kutsujalle kopiorakentajalla tehty väliaikaiskopio return-lauseessa olevasta oliosta.
  - d) C++:ssa periytetyn luokan purkajan rungossa täytyy kutsua kantaluokan purkajaa, jotta myös kantaluokkaosa siivotaan oikein.
  - e) Julistamalla itsensä toisen luokan ystäväksi (*friend*) luokka saa pääsyn tämän toisen luokan private-osaan.
  - f) Moniperiytyemisessä luokka periytyy kahdesta tai usemmasta keskenään vaihtoehtoisesta kantaluokasta.