

OHJ-2050 Ohjelmointikielten periaatteet

Tentti 17.12.2012

Ei laskimia, ei lunttaushuijauksia, palauta tehtäväpaperi t: Bitti

1 Määrittele

Määrittele, mitä seuraavat käsitteet tarkoittavat. Esimerkit on ok.

- a) L- ja R-arvo
- b) pinodynaaminen muuttuja
- c) lambda-funktio
- d) aktivaatietietue (kuva ois kiva :)
- e) tyyppiekvivalenssi
- f) paradigma (ohjelmistohommissa)

2 Käännösvaihehommia

Tässä on koodi:

```
1. void f( std::string* p, int i )
2. {
3.     *p = "Ota rennosti, kyllä se tästä!";
4.     std::cout << --i-p--->length() // WTF ?!
5. }
```

- a) Jaa ohjelma leksikaalialkioihin vaikka pystyviivoilla.
- b) Tee ohjelmaan leksikaalivaiheessa havaittava virhe. Perustele, miksi se on siinä vaiheessa.
- c) Näytä ohjelmasta kolme eri syntaktista rakennetta ja kerro, mitä ne ovat. Kerro selvästi, mistä mikäkin alkaa ja mihin mikäkin loppuu.
- d) Tee ohjelmaan syntaksivirhe. Perustele, miksi se on siinä vaiheessa.
- e) Tee ohjelmaan semanttinen virhe. Perustele, miksi se on siinä vaiheessa.
- f) Mitä muita vaiheita on käännöksessä? Mitä niissä tehdään?

3 Funktionaalisuudesta ja sidonnasta

- a) Mitä merkittäviä eroja funktionaalisissa kielissä on imperatiivisiin verrattuna? Mitä etuja siinä on?
- b) Mitä on laiska laskenta ja miten se sopii funktionaalsiin kieliin?
- c) Mitä eroa dynaamisella ja staattisella tyyppityksellä on? Mitä etuja kummassakin on?

4 Parametrinvälitys

- a) Mitä parametrinvälitysmekanismeja on kurssilla käsitelty ja miten ne toimivat? Selitä kustakin niiden toiminta.
- b) Ohjeisessa ohjelmassa % on jokin kurssilla käsitellyistä parametrinvälitysmekanismeista. Leikitään, että C++:ssa on kukin toteutettuna. Kirjoita ohjelman tulosteet kullakin mekanismilla.

```
1. void f( int% x, int% y )
2. {
3.     y = y + 1;
4.     cout << x << y << endl;
5.     x = 3;
6.     if( y == 2 )
7.         throw( "Poikkeus!" );
8. }
9. int main()
10. {
11.     int a = 0;
12.     vector< int > b;
13.     b.push_back( 1 ); b.push_back( 4 );
14.
15.     try
16.     {
17.         cout << a << b[0] << b[1];
18.         f( b.at( a ), a );
19.         cout << a << b[0] << b[1];
20.         f( b.at( a ), a );
21.     }
22.     catch(...) { cout << "!";
23.     }
24.     cout << a << b[0] << b[1];
25. }
```