

OHJ-1151 Ohjelmointi IIe, tentti

16.5.2007, Essi Lahtinen

Kirjoita jokaiseen vastauspaperiisi selkeästi ylläolevat otsikkotiedot, nimesi, opiskelijanumerosi, koulutusohjelmasi, vuosikurssisi, vastauspaperin järjestysnumero ja jättämiesi vastauspapereiden kokonaismäärä. Jos olet korottamassa aikasempaa suoritustasi, kerro myös, miltä vuodelta suoritukseksi ovat. *Jätä vastauspapereihisi vähintään 3cm ulkoreunamarginaalit. Jätä ensimmäiseen paperiin vähintään 10 cm yläreunamarginaali tai piirrä valmis taulukko tehtävien pistemäärille.* Kirjoita selkeällä käsialalla. Tentin tarkastaja ei ole hieroglyfien erikoisasantuntija. Onnea tenttiin!

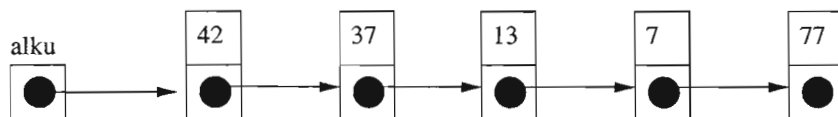
Tehtävä 1

Selitä *lyhyesti*, mitä seuraavat termit tarkoittavat:

1. olio (*object*) (1p.)
2. oletusrakentaja (*default constructor*) (1p.)
3. oletusparametri (*default parameter*)
4. moduuli (*module*) (1p.)
5. osoitin (*pointer*) (1p.)
6. Makefile (1p.)

Tehtävä 2

Oletetaan, että määriteltynä on tietuetyyppi `Alkio`, joka sisältää mm. `Alkio*`-tyyppisen kentän seuraava ja `int`-tyyppisen kentän data. Lisäksi on koottuna seuraavanlainen tietorakenne:



1. Oletetaan, että tarvittavat `include`-direktiivit jne. ovat käytössä. Minkä toimenpiteen seuraava ohjelmapätkä suorittaa kuvatulle tietorakenteelle? (2p.)

```
Alkio* osoitin = alku;
```

```
while( osoitin != 0 && osoitin->seuraava != 0 ) {  
    osoitin->seuraava = osoitin->seuraava->seuraava;  
    osoitin = osoitin->seuraava;  
}
```

2. Edellinen ohjelmapätkä menee virheittä läpi kääntäjästä, mutta mikä virhe siinä siitä huolimatta on? (1p.)
3. Miten virhe ilmenee, kun ko. ohjelmaa ajetaan? (1p.)
4. Miten korjaat ko. virheen? (2p.)

Tehtävä 3

Vastaa *lyhyesti*, esimerkiksi ranskalaisia viivoja käyttäen.

1. Tekeillä on ohjelma, jolle annetaan käynnistettäessä seuraavantyyllisiä komentorivi-parametreja: `./ohjelma -v -k <luku> -l <luku1> <luku2>`
eli parametrille `-v` ei anneta mitään lisätietoja, parametrille `-k` annetaan yksi luku ja parametrille `-l` kaksi lukua. Lisäksi parametrit saa antaa missä järjestyksessä tahansa, kunhan niihin liittyvät luvut annetaan välittömästi niiden perässä. Parametria `-v` ei ole pakko antaa, vaan se on valinnainen. Ohjelman komentoriviparametrien tarkistus halutaan tehdä tieto-ohjattua ohjelmointia (*data-driven programming*) käyttäen. Kirjoita C++:na sopiva tietorakenne komentoriviparametrit tarkistavan funktion käyttöön. (2p.)
2. Määriteltynä on class `Luokka`, jolla on jäsenmuuttuja `string nimi_` ja jonka julkiset jäsenfunktiot on toteutettu seuraavasti:

```
Luokka::Luokka( string nimi ): nimi_( nimi ) {  
    cout << "1 " << nimi_ << endl;  
}
```

```
Luokka::~Luokka() {  
    cout << "2 " << nimi_ << endl;  
}
```

```
Luokka::Luokka( const Luokka& l ): nimi_( l.nimi_ + " jr." ) {  
    cout << "3 " << nimi_ << endl;  
}
```

Oletetaan, että tarvittavat include-direktiivit yms. ovat kunnossa. Mitä seuraava ohjelma tulostaa? (3p.)

```
void funktio( Luokka l, Luokka* o ) {  
    cout << "Nan nan naa na" << endl;  
}
```

```
int main() {  
    Luokka olio( "Jonna" );  
    Luokka* osoitin = new Luokka( "Erin" );  
    funktio( olio, osoitin );  
    return EXIT_SUCCESS;  
}
```

3. C++, C, Ada, Python ja Perl ovat esimerkkejä imperatiivisista ohjelmointikielistä. Kurssilla käsiteltiin erittäin lyhyesti myös Haskellia, joka on funktionaalinen ohjelmointikieli, ja Schemeä, joka on hyvin paljon funktionaalisenkaltainen ohjelmointikieli. Kerro jokin ero imperatiivisten ja funktionaalisten kielten välillä? (1p.)

Tehtävä 4

Oletetaan, että määriteltynä on tietuetyyppi `Henkilo`, joka sisältää mm. kentät `int kenganNro` ja `set< string > lapset`, joista jälkimmäinen sisältää henkilön lasten nimet. Tietorakenteeseen `map< string, Henkilo >` on talletettuna tietoja monista henkilöistä siten, että mapin avaimena on henkilön nimi. Kirjoita rekursiivinen C++-funktio, joka selvittää, kuinka suuri kengännumero on henkilön X isokenkäisimmällä jälkeläisellä. Voit olettaa, että henkilöiden nimet ovat yksikäsitteisiä ja että jokaisen henkilön tiedot löytyvät tietorakenteesta. (6p.)

Tehtävä 5

Korttipelien ohjelmoimisesta innostunut ystävämme on huomannut, että olisi näppärää, jos käytettävissä olisi yleinen korttipakka-moduuli, jota voisi sitten käyttää apuna yhtälailla pokeri- kuin pasianssipelinkin toteuttamisessa.

Ystävämme päättää toteuttaa korttipakan abstraktina tietotyypinä. Suunnittele ja kirjoita C++:na alustavat versiot tietotyypin rajapinnoista. Kirjoita tarpeeksi kommentteja, jotta tärkeimmät suunnitteluratkaisut käyvät ilmi. (6p.)