

## Ohjelmoinnin edistyneet piirteet – OSA II – Harjoitukset 2

### 1. Tehtävä – Luokan rakentajat eli konstruktorit (constructor) – 2p

Lisää viime harjoituksissa luoduille Henkilo ja Paivays –luokille kaksi rakentajaa, jossa olion tiedot voidaan alustaa olioita luotaessa:

- Parametriton rakentaja (oletusrakentaja)
- Parametrillinen rakentaja, jossa oliolle voidaan välittää olion tietojäsenten arvot

Kirjoita rakentajien toteutuksiin alustusten lisäksi myös cout –tulostus tyyliin ”Henkilo –luokan parametriton rakentaja” ja ”Henkilo-luokan 2 parametrinen rakentaja”.

Testaa molempien luokkien rakentajien toimivuus pääohjelmassa, esim.

```
Henkilo pekka ("Pekka", 20);  
pekka.tulostaHenkilonTiedot();
```

Kokeile luoda taulukollinen taulukollinen olioita tyyliin Henkilo henkilot[10]. Mitä rakentajaa tällöin kutsutaan?

### 2. Tehtävä – olioiden luonti ja tuhoaminen sekä purkaja (1p)

Toteuta edellisiin luokkiin myös purkaja (destructor). Tee pieni testiohjelma (pääohjelma), jossa luot olion/olioita seuraavasti:

- Heti pääohjelman alussa
- If –lauseen lohkon sisällä
- While tai for –lauseen lohkon sisällä
- Aliohjelman sisällä (tee tätä varten yksikertainen aliohjelma)

Milloin purkajaa (destruktori) kutsutaan kunkin olion tapauksessa?

Kokeile myös dynaamista (kekoon) olion luomista new –operaattorilla (tästä lisää tulevilla kerroilla) joko pääohjelmassa, if –lohkon tai aliohjelman sisällä. Olio luodaan tällöin seuraavasti:

```
Henkilo* pekka = new Henkilo("Pekka", 20);
```

Milloin dynaamisesti luotu olio tuhotaan vai tuhotaanko ollenkaan? Mitä sinun tulee tehdä, jotta olio saadaan tuhottua?

### 3. Tehtävä – Noppa -luokan toteuttaminen (1p)

Toteuta luokka Noppa, jonka avulla voidaan simuloida nopan heittoa. Nopalla on tietojäsenenään nopanheiton viimeisin lukema (int).

Toteuta Noppa -luokalle ainakin seuraavat metodit:

- Rakentaja (alusta nopan lukema joksikin välillä 1-6)
- getViimeisinLukema() : int – palauttaa viimeisimmän heiton lukeman
- heitäNoppaa() : void – metodin sisällä arpoo rand() -funktiota käyttäen nopalle uuden lukeman
- kerroViimeisenHeitonLukema() : void – metodi, joka kertoo (cout) nopan viimeisimmän lukeman näytölle tyyliin. Viimeisin heitto on 6!

### 4. Tehtävä – Noppa -luokan laajentaminen (1p) – hieman haastavampi

Joissain peleissä (esim. Kimble) käytetään vain yhtä noppaa, joissain kaksi (esim. Monopoli), ja joissain jopa viittä noppaa (esim. Yazty). Laajenna nyt edellisen tehtävän Noppa -luokkaa siten, luokka voi käsitellä useamman nopan heiton. Noppia voi nyt olla 1-5 kpl, joten lisää luokalle uusi tietojäsen noppienLkm (int) sekä luokan sisälle 5 int muuttujaa noppia varten.

Lisää luokalle ainakin seuraavat metodit ja muuta muutamaa muuta metodia:

- Lisää luokalle oletusrakentaja, jossa alustat noppien lukumääräksi oletuksena 1.
- Lisää luokalle parametrillinen rakentaja, jossa käyttäjä voi syöttää noppien lukumäärän (1-5).
- Lisää getteri ja setteri noppien lukumäärälle. Getteri palauttaa noppien lukumäärän ja setterin avulla käyttäjä voi muuttaa käytössä olevien noppien lukumäärää.
- Muuta heitäNoppaa() metodia nyt siten, että se arpoo niin monelle nopalle lukeman, kuin käyttäjä on asettanut.
- kerroViimeisenHeitonLukumäärä() kertoo nyt viimeisimmän heiton noppien lukemien summan sekä käytössä olevien noppien lukumäärän tyyliin ”viimeisin heitto 21. Heitetty 4 nopalla”

Luokan käyttöesimerkki:

```
// Luodaan noppapari
Noppa monopolI(2);

// käyttää sisäisesti nyt kahta noppaa, joille arpoo lukemat
monopolI.heitaNoppaa();

// tulostaa arvotun heiton lukeman esim. tyyliin:
// Noppa1: 4, noppa2: 2, yhteensä 6
monopolI.kerroViimeisenHeitonLukema();
```

