

## Harjoitukset

Dokumentti on nyt lopullinen.

Arvosanjakauma on, kuten lupasin, täysin sama kuin oli Java-jaksolla:

- tehtävistä kerättynä vähintään 8 pistettä → arvosana 1.
  - tehtävistä kerättynä vähintään 12 pistettä → arvosana 2.
  - tehtävistä kerättynä vähintään 16 pistettä → arvosana 3.
  - tehtävistä kerättynä vähintään 19 pistettä → arvosana 4.
  - tehtävistä kerättynä vähintään 23 pistettä → arvosana 5.
1. Toteuta pino (siis stack) kokonaislukujen tallentamista varten. Luokka käyttää elementtien tallennukseen ennalta määrätyn kokoista kokonaislukutaulukkoa. Luokassa tarvitaan ainoastaan push-, pop-, size- ja empty-operaatiot. Et saa siis käyttää Javan valmiita luokkia vaan luokka on luotava ihan omin pikku kätös. 2p.
  2. Sama kuin edellä, mutta luokkaan pitää voida tallentaa mitä tahansa olioita. Tarvitset siis geneerisen tyypin. Lisäksi pinon kokoa ei saa ennalta määrätä. 1p.
  3. Toteuta kaksipäinen jono kokonaislukujen tallentamista varten. Jonoon on voitava lisätä ja poistaa alkia sekä jonon alkuun että loppuun. Jos jono on täysi, lisäys ei saa onnistua ja jos jono on tyhjä, poisto ei voi onnistua. Lisäksi tarvitaan operaatiot empty ja size. Selvyyden vuoksi mainittakoon, että tässäkin kohtaa et saa käyttää Javan valmiita luokkia. Niiden sisältöä voi kyllä tutkia. 2p.
  4. Toteuta Stack-olion avulla merkkijonon kääntäminen (siis merkkijonosta ”sika” tulee ”akis”). 1p.
  5. Arvo 200 miljoonaa (tai niin monta kuin koneesi muisti antaa myöten) numeroa integer-arvoalueelta ja tallenna ne ArrayList- sekä LinkedList-rakenteisiin. Tee ohjelma, jolla testaat, kumpi rakenne on nopeampi ja kuinka paljon, kun noudat jokaisen numeron rakenteesta yksitellen. 1p.
  6. Kirjoita Employee-luokka, jossa ominaisuuksina id, nimi ja ammattinimike (mikään ominaisuus ei ole yksikäsitteinen). Toteuta luokkaan hashCode- ja equals-metodit. Testaa toiminta kattavasti. 1p.
  7. Toteuta *parhaaksi katsomallasi* tavalla 5000 Employee-luokan ilmentymää ja tallenna ne HashSet-rakenteeseen. Parhaaksi katsomasi tapa on oltava sellainen, että duplikaatteja syntyy. 2p.
  8. Kuten edellinen, mutta tallennus HashMap-rakenteeseen. HashMap sen takia, että tässä kohden ei saa mennä hukkaan yhtään ilmentymää. Tämä tarkoittaa siis sitä, että jokainen ilmentymä on saatava talteen, vaikka se olisi duplikaatti. 2p.
  9. Kloonaa tehtävässä 7 muodostunut HashSet niin, että syntyy todellinen kopio. Osoita, että kloonaa ja varmasti toimii oikein. 1p.
  10. Arvo sata lukua väliltä 1–100 ja talleta ne HashSetiin. Kuinka monta numeroa tallentuu keskimäärin ja miten testasit asian? Kuinka monta numeroa pitää arpoa, että settiin tulevat tallennetuiksi kaikki numerot väliltä 1-100? 1p.
  11. Arvo merkkijonoja, joiden pituus on 1–5 merkkiä ja joissa käytetään satunnaisesti aakkosia a–c. Talleta syntyneiden merkkijonojen hash code -arvot HashSetiin. Kuinka monta erilaista hash code-arvoa syntyy? Todista asia ohjelmallisesti sekä matematiikan avulla. 2p.

12. Jos sallitaan edelliskohdassa minkä mittainen merkkijono tahansa, niin mikä on se merkkijonon pituus, jonka tarkistaminen ohjelmallisesti ei ole enää kovin mielekästä? 1p.
13. Pidä merkkijonon mitta kolmena (siis merkit a-c), mutta kasvata merkkijonon pituutta. Talleta hash code -arvot edelleen HashSetiin. Talleta lisäksi itse merkkijonot TreeMappin siten, että pidät yllä tietoa siitä, kuinka monta kertaa merkkijono on esiintynyt duplikaattina. Yritä päivittää molempia rakenteita HashSetin arvojen perusteella. Kun merkkijonon mitta on jossain 13 kieppeillä, niin pitäisi alkaa tapahtua hassuja asioita. Mitä tapahtuu ja miksi? (Tästä saa pisteet, vaikka ei saisikaan viimeistä kysymystä selvitettyä). 2p.
14. Teamsista löytyy tiedosto Jobs.txt (lopun tehtävät tämän tiedoston pohjalta). Tee luokka, jonne tiedot voi tallettaa. Lisäksi tarvitaan lajittelijat viikonpäivää, työvuoron kestoa ja aloitusaikaa varten. 3p.
15. Toteuta tietorakenne, jossa kaikkien töiden eri kestojen lukumäärät. Siis kuinka monta kpl kutakin kestoa. Kestot pitää olla suurimmasta pienimpään. 1p.
16. Toteuta järjestetty tietorakenne, jossa avaimena on työvuoro lajiteltuna ensin viikonpäivän mukaan ja sitten aloituskellonajan mukaan. Arvona rakenteessa on joukko kaikista vuoroista, jotka täsmäävät avaimeen. 2p.
17. Lajittele kaikki työt ensin päivän mukaan, sitten keston mukaan ja lopuksi aloitusajan mukaan. Talleta jonnekin. 1p.