HAJAUTETTUJEN SOVELLUSTEN MUODOSTAMISTEKNIIKAT
TENTTI 13.2.2012

Tehtävistä saatava maksimipistemäärä on merkitty kunkin tehtävän loppuun. Tentin läpäisemiseksi pitää saada puolet pisteistä. Syntaksiin liittyvät pienet yksityiskohtavirheet eivät ole merkittäviä.

1. Mikä on RMI? Mihin sitä käytetään? Miten? Mitä osapuolia RMI:n käytössä on ja miten eri osapuolet toteutetaan? Miten RMI toimii? Miten lukkoja voi/tulisi käyttää erilaisten RMI:n olioiden yhteydessä? (8pts)

2. **(a)** Miten TCP-sokettien yhteydessä saadaan aikaan tilanne, jossa palvelinpuoli ajetaan hallitusti alas, jos mikään asiakassovellus ei ole ottanut yhteyttä palvelimeen 2 vuorokauteen? (2pts)

   (b) Miten TCP-pohjainen sokettiyhteys luodaan? Mitä huomionarvoisia seikkoja palvelinpuolen toteutukseen liittyy? (2pts)

3. Toteuta tuottaja-kuluttaja tilannetta varten kokonaislukujen FIFO-jono (First-In First-Out), jonka maksimi koko annetaan konstruktorissa. FIFO-jono tarkoittaa kokonaislukujen jonoa, jonka koko voi vaihdella, alkioita lisätään alkuun ja otetaan lopusta. Olkoon kyseisen luokan nimi Fifo. Tuottaja(t) – jotka ovat säikeitä – voivat lisätä yhden luvun kerrallaan puskurin alkuun, kunhan puskurissa vain on tilaa. Jos tilaa ei ole, operaation pitää odottaa kunnes FIFO-jonossa taas on tilaa ko. alkiolle. Vastaavasti kuluttaja(t) – niinikään säikeitä – voivat kuluttaa puskurin lopusta yhden luvun kerrallaan, kunhan puskurissa on jotain kulutettavaa. Jos kulutettavaa ei ole, operaation pitää odottaa kunnes FIFO-jonossa on kulutettavaa. Lukujen taltioimisen lisäksi halutaan pitää kirjaa puskurissa olevien lukujen summasta. Tee edellistä varten metodit, joilla voidaan luomisen lisäksi suorittaa 'puskurin nykyisen alkioiden määrän tiedustelu', 'luvun lukeminen', 'luvun kirjoittaminen', 'puskurissa olevien lukujen summan tiedustelu' ja 'puskuriin liittyvän kokorajoituksen tiedustelu'. Pidä huoli siitä, että puskurin kokoa ei ylitetä ja monisäikeisyys huomioidaan asianmukaisesti lukkoja käyttämällä. (Käytä toteutuksen osana kokonaislukutaulukkoa tai luokkaa Vector.) (6p)

Metodi	param. tyypit	tulos	merkitys
Konstruktorit			
Vector	–	Vector<E>	luo tyhjän vektorin
Olioiden metodit			
addElement	E a	–	lisää olion a vektorin loppuun
insertElementAt	E a, int i	–	**lisää** alkion a vektorin kohtaan i; ArrayIndexOutOfBoundsException
elementAt	int i	E	palauttaa alkion kohdasta i; ArrayIndexOutOfBoundsException
firstElement	–	E	palauttaa vektorin ensimmäisen alkion; NoSuchElementException, jos vektori tyhjä
lastElement	–	E	palauttaa vektorin viimeisen alkion; NoSuchElementException, jos vektori tyhjä
isEmpty	–	boolean	palauttaa true, joss vektori tyhjä
removeAll-Elements	–	–	tyhjentää vektorin alkioista
remove-ElementAt	int i	–	poistaa kohdassa i olevan alkion vektorista; ArrayIndexOutOfBoundsException
size	–	int	palauttaa vektorin koon

Taulukko 1: Luokan java.util.Vector joitakin hyödyllisiä metodeja.