

Tehtävä: Jono on abstrakti tietotyyppi¹. Jonoon ensiksi lisätty elementti poistetaan jonosta ensimmäisenä (FI-FO, First-In-First-Out).

Suunnittele ja toteuta rajapinta **JonoInterface**, jolla on metodit *lisaaElementti(E e)* ja *E palautaElementti* sekä metodit, *boolean onkoJonoTyhja* ja *boolean onkoJonoOlemassa* eli onko tallennusrakenteelle varattu tilaa. (E on joku tietotyyppi)

Tee ohjelma, jossa simuloidaan kaupan asiakkaiden kassajonoa. Tee kassajonosta oma luokka, joka toteuttaa **JonoInterface**:n metodit. Toteuta kassajono taulukkoa käyttäen. Jos taulukko tulee täyteen, luodaan uusi isompi taulukko, jonne vanhan taulukon arvot kopioidaan.

Asiakkaalla on attribuutit *asiakasnumero* ja *maksutapa*. Asiakas voi olla joko käteisasiakas tai luottokorttiasiakas. Käteisasiakkaaseen liitetään **KateisMaksu**-olio ja luottokorttiasiakkaaseen liitetään **LuottokorttiMaksu**-olio.

Kun asiakas poistuu jonosta suorittaa hän maksun. Maksutapahtuma simuloidaan tulostamalla maksun määrä ja maksutapa ja asiakasnumero. Yhden asiakkaan maksun kesto voi simuloida "nukuttamalla" ohjelma **This.sleep(aika_ms)** -metodilla, katso: <https://docs.oracle.com/javase/tutorial/essential/concurrency/sleep.html>

Suunnittele ja toteuta asiakkaiden jonoon lisääminen ja jonosta poistaminen main-metodissa käyttäen satunnaislukuja (0, 1). Jos satunnaisluku on nolla lisätään uusi asiakas jonoon ja arvolla yksi asiakas poistuu jonosta.

Simuloinnissa voi arpoa montako kertaa ohjelma toistaa jonon käsittelyä (jonoon lisäämistä, jonosta poistamista). Samoin maksutapahtumien tiedot (maksun määrä, luottokortin numero, asiakkaan asiakasnumero) voidaan tuottaa satunnaisluvuilla, joten käyttäjän ei ole pakko syöttää ohjelmalle mitään tietoja.

Talleta rajapinta ja luokat sekä main-metodi omiin tiedostoihin.

Ohjelman demoversion **.class** -tiedostot voi ladata viikkotehtävien sivulta ja suorittaa omalla koneella. Ohjelma lähtee käyntiin komennolla **java JonoMain**

Muista palautetekstitiedosto. Muista pakata tiedostot yhdeksi pakkaukseksi ennen palauttamista.

¹ **Abstrakti tietotyyppi** ([engl.](#) *abstract data type, ADT*) on tyyppin määrittely, joka määrittelee vain tyyppin tietosisällön ja tyyppiin kuuluvat operaatiot, ei tyyppin toteutustapaa. Toisin sanoen ADT määrittelee tarkalleen ja täydellisesti tyyppin julkisen [rajapinnan](#) eikä mitään muuta (lähde:wikipedia.org)