Teemad

Lõimede loomine ja käivitamine

Andmete jagamine mitme lõime vahel

Ligipääsu jadastamine e. sünkroniseerimine

Ülesanne - Shakespeare

1. On antud klass Book, mis tähistab tähtedest koosnevat raamatut. Tähtede salvestamiseks kasutatakse tavalist ArrayList'i. Meetod write(letter) lisab etteantud tähe raamatu lõppu. Meetod getContents() tagastab raamatu sisu tekstina.
2. Kirjanik (Writer) on lõim, kes kirjutab raamatusse ühe tähe ja pärast kirjutamist puhkab ühe sekundi. Pärast puhkamist kirjutab uue raamatusse tähe, puhkab jne... Writeri realiseerimisel laiendage Thread klassi. Lõim võiks käivituda kohe, kui Writer objekt on loodud (st. konstruktoris). Kirjutaja võiks igal töötaktil printida oma tegevuse ka konsoolile:

Kirjutasin: A

...

Kirjutasin: U

1. Lugeja (Reader) on lõim, kes loeb raamatust terve selle sisu, trükib selle konsoolile ning puhkab pool sekundit. Näide:

Lugesin: A

...

Lugesin: AU

Sarnaselt kirjutajaga kordab lugeja oma tegevust väsimatult.

1. Looge testprogrammis üks raamat ning kummastki lõimest (Writer, Reader) üks objekt. Veenduge, et mõlemad lõimed oleksid seotud ikka sama raamatuga.
2. Täiendage lugejat selliselt, et konsoolile prinditaks lisaks raamatust loetud tekstile ka selle pikkus. Nt:

Lugesin: 1 A

...

Lugesin: 2 AU

1. Looge testprogrammis 2 lugejat ja 10 kirjutajat ning laske programmil paar minutit käia. Mõne aja pärast hakkavad ilmnema anomaaliad:

a. Kuigi programmi alguses lõime 2 lugejat, on kaovad nad kuhugi ära.

b. Kui alguses lisanduvad tähed raamatusse kümnekaupa (nt. "Lugesin: 10

EEHLAJCKPU" "Lugesin: 20 EEHLAJCKPUURIPVESQQO"), siis mõne aja

pärast lisandub tähti kõigest 9- või isegi 8-kaupa.

Peatage programm ning üritage leida vea tekkepõhjus (vihje: konsooliväljundil peaks olema näha tekkinud erind).

NB! Kui vigu ei ilmne (sõltub kuufaasidest), on Teil vedanud – koodi testimiseks tehke järgmist:

a. Kommenteerige ajutiselt maha lõimede töötsüklis tehtud väljaprindid.

Põhjus: konsoolile trükkimine on jadastatud tegevus (st. üks

sümbol/tekstirida korraga), misvõttab ka mõnevõrra aega - see peidab

ühiskasutuse vead ära.

b. Lisage ajutiselt (testimise ajaks) iga lõime töötsüklisse

Thread.yield().

Põhjus: paljude korraga töötavate lõimede korral ei pruugi Thread

scheduler anda igale lõimele tööaega (mingil põhjusel saab

protsessoriaja endale ainult 1. kirjutaja ning seetõttu ei ilmne

vigu).

Pärast katsetamist tuleks Thread.yield() laused uuesti maha võtta.

Vea parandamiseks jadastage kõik ligipääsud jagatud andmeobjektile e.

raamatu sisuks olevale andmestruktuurile. Vihje: synchronized

Kui viga on parandatud, laske programmil uuesti mõnda aega töötada, et

veenduda selle korrektsuses.