

Лабораторна робота №14

1. Розробити всі програми багатопоточні програми за наступними критеріями.
 - Використовувати методи, що знаходяться в пакеті `java.util.concurrent`.
 - Не застосовувати препарат слово `synchronized`.
 - Все об'єкти, що отримують доступ до ресурсу, повинні бути потоками.
 - Використовувати принципи ООП.
 - Можна не використовувати графічний інтерфейс, програма може бути консольною.
2. Написати клієнт для роботи бібліотеки. У бібліотеці доступні для читання кілька книг. Однакових книг в бібліотеці немає. Деякі книги можна видавати на руки, деякі – тільки в читальний зал. Читач може брати на руки і в читальний зал декілька книг одночасно.
3. Написати клієнт для роботи порту. Кораблі можуть заходити в порт для розвантаження/завантаження контейнерів. Кількість контейнерів, що знаходяться в поточний момент в порту і на кораблі, має бути невід'ємною і не перевищувати задану вантажопідйомність судна і місткість порту. У порту працює кілька причалів. У одному причалі може стояти лише один корабель. Корабель може завантажуватися біля причалу, розвантажуватися або виконувати обидві дії.
4. Розробіть багатопоточну програму, що виконує обчислення добутку матриць $A_{m \times n}$ і $B_{n \times k}$. Елементи c_{ij} матриці добутку $C = A \times B$ обчислюються паралельно p однаковими потоками. Якщо в деякому потоці вже відбувається обчислення елемента c_{ij} матриці C , то у наступному потоці, що починає виконання, буде обчислюватись елемент c_{ij+1} , якщо $j < i$, або c_{i+11} , якщо $j = k$. Після завершення обчислення елемента матриці-добутку, у потоці відбувається перевірка, чи немає елемента, який ще не обчислено. Якщо такий елемент є, то в потоці почнеться його обчислення. В іншому випадку відправляється (призначене для користувача) повідомлення про завершення роботи даного потоку і призупиняється виконання. У головному потоці, отримавши повідомлення про завершення обчислень від всіх потоків, виводиться результат на екран і запускається потік, для запису результату у файл. У кожному потоці зробіть затримку виконання обчислень (щоб працювали всі потоки). Синхронізацію потоків між собою організуйте через критичну секцію або м'ютекс.
5. Є програма-лічильник `Counter`, для обчислення кількості викликів. Дайте відповіді на наступні запитання:
 - Чому значення лічильника відрізняється від очікуваного?
 - Як це можна виправити не використовуючи `synchronized`? Спробуйте закоментувати звернення до `yield()`.
 - Чи зміниться значення? Внесіть зміни в код програми, щоб отриманий результат відповідав очікуваному.