תרגילים

threads

1. פתור-
   a. ייצר פרוייקט חדש לטובת הרצה ב- multi threaded, קרא לו לבחירתך
   b. ייצר 10 threads ולכל אחד תן name בסדר עולה. "1" , "2" ... "10"
   c. דאג שכל thread יפעיל פונקצייה בשם *print_order*
   d. הרץ את כל ה – threads (בסדר אקראי)
   e. בתוך פונקציית *print_order* הדפס את שם ה- thread הנוכחי, **אבל-** עליך לדאוג שהמספרים (שהם שמות הת'ראדים) יודפסו בסדר עולה : "1", "2", ...
      יש להיעזר ב- synchronized, wait, notify,notifyAll וכו'
   f. כעת, שנה את הקוד כך שכל thread ייקרא לפונקציה בשם *print_ten10*
   g. בתוך פונקציית *print_ten10* הדפס את שמו של ה- thread הנוכחי, **אבל-** עליך לדאוג ש- "10" יודפס אחרון ושאר ה- threads יודפסו בסדר שהגיעו.
      יש להיעזר ב- synchronized, wait, notify,notifyAll וכו'

2. מדוע כדאי להשתמש ב resource pool pattern ?

3. קח את הקוד שכתבנו בכיתה הפונה ל- sqlite דרך JDBC
   https://github.com/fullstack180822/23.03.2023/tree/main/sqlitepj
   וודא שהוא עובד כראוי (שהדרייבר עובד, והקובץ DB נמצא אצלך במחשב וכו').

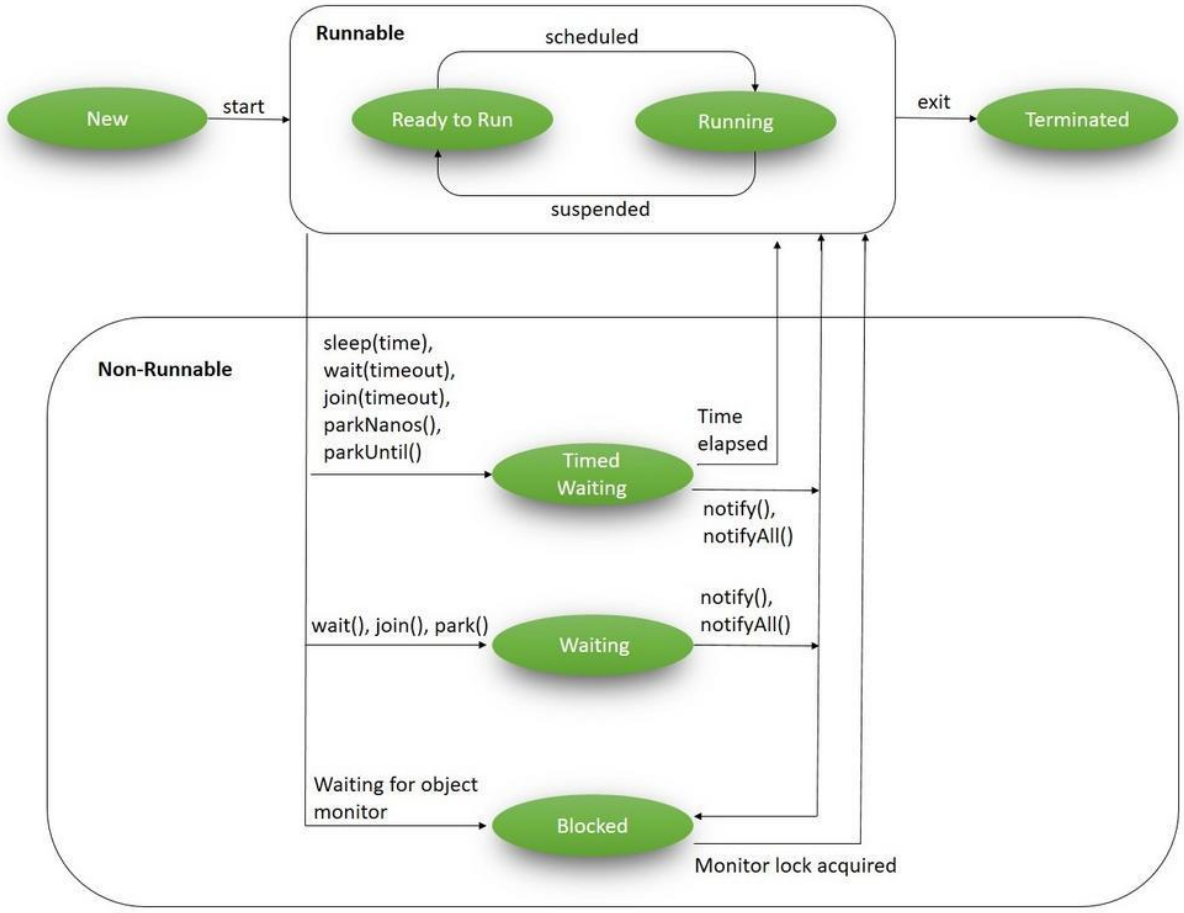
   קח את הקוד שכתבנו בכיתה של ה- connection pool.,
   https://github.com/fullstack180822/23.03.2023/tree/main/DbPoolPattern
   הוסף שורות הדפסה (println) לכל אחד מהמצבים (כמו logger). לדוגמא- בכל פעם שת'ראד עובר ל- wait, בכל פעם שנלקח חיבור, בכל פעם שמוחזר חיבור, בכל פעם שת'ראד עושה notify וכו'

   כעת במקום להחזיר אובייקט MyDbConnection ב connection pool, החזר אובייקט **אמיתי** של Connection לדאטא בייס. כלומר- במקום להחזיק רשימה של MyDbConnection, החזק רשימה של java.sql.Connection (פעילים ומחוברים). וזה האובייקט שיוחזר בפונקצית- getConnection

   הגבל את הגודל המקסימאלי של ה connection pool ל- 3
   כעת הפעל 8 ת'ראדים (threads) במקביל ובקוד שלהם
   א. בקש חיבור מה connection pool
   ב. בצע פעולה לתוך הדאטא בייס (לדוגמא הוספת רשומה אקראית), תוך שימוש ב- connection שקיבלת. בדוק בהדפסות שהכול התנהל כראוי

4. **אתגר: קח את הקוד שכתבנו בכיתה הפונה ל- sqlite דרך JDBC, ושנה אותו שיפעל מול Postgresql
5. מדוע לא כדאי לעשות stop לת'ראד? מה האלטרנטיבה המומלצת?
6. האם ניתן לעשות start לת'ראד שהסתיים ולהתחילו מחדש?
7. כיצד ניתן לראות את התהליכים הרצים במחשב (ווינדוס)? ואת הת'ראדים הרצים?
   האם JAVA מנהל את הת'ראדים? או מערכת ההפעלה?
8. על מי עושים wait ועל מי עושים notify ?
9. מהו race condition? מהו context switch ?

המשך בעמוד הבא...



10. בדיאגרמה מעליך-
   a. מה ההבדל בין ready to run לבין running
   b. מה ההבדל בין wait לבין wait עם פרמטר timeout
   c. לאיזה מצב נכנס ת'ראד כאשר מייצרים אותו?
   d. לאיזה מצב נכנס ת'ראד אחרי start ?
   e. לאיזה מצב נכנס ת'ראד כאשר הוא נכנס למתודה שהיא synchronized ולא הצליח לקבל מפתח?
   f. הם notifyall ישחרר את כל הת'ראדים הממתינים? או רק את אלה הממתינים על אותו המפתח?

האם המפתח הנועל יכול להיות primitive ? או חייב להיות אובייקט? מדוע?

בהצלחה