

שיעורי בית

1. מה עושה פעולת **Suspend**? מה עושה פעולת **Resume**? מה עשוי להיות היתרון בכך שנעביר את ה-Thread למצב של- **Suspend** ולאחר מכן נקרא ל- **Resume**? מדוע הפונקציות **Suspend** ו- **Resume** הם **deprecated**?
2. מהו עושה פעולת **Wait**? מה עושה פעולת **Pulse**? מה עושה פעולת **PulseAll**?
3. מהו **Race Condition**?
4. מהו **dead lock**?
5. אילו תנאים צריכים להתקיים כדי שאוכל להפעיל את- **Wait Pulse**?
6. מהם **design patterns**? מיהם **Gang of Four**?
7. מתי נשתמש ב- **Singleton design pattern**?
8. אילו "פעולות" יש לבצע במחלקה כדי שתהפוך ל- **Singleton**? אם נסווג את הבנאי במחלקת singleton ל- **protected** בימקום **private**. האם הסינגלטון עדיין יהיה בתוקף? מדוע? אם כן, מה היתרון בשימוש ב **protected**?
9. כתוב **סינגלטון** ובתוכו פונקציה בשם **GetTime** המחזירה את השעה (השעה חוזרת במחרוזת "\$Time in Isreal {time}")
10. *אתגר:
כתוב מחלקה בשם **AsyncPrinter** אשר מחזיקה תור (**Queue**) של string לטובת הודעות. צור פונקציה בשם **AddMessage** אשר מקבלת מחרוזת ומוסיפה אותה לתור. לאחר ההוספה, קרא ל **Pulse** (אל תשכח לעטוף את הקוד ב lock).
צור פונקציה בשם **CheckPrintMessage** אשר בודקת אם יש הודעה אחת (לפחות) בתור. אם כן, היא שולפת הודעה מהתור, מדפיסה את ההודעה ויוצאת מהפונקציה. אם לא, בצע- **Wait**. אחרי ה Wait - שלוף הודעה מהתור, הדפס את ההודעה וצא מהפונקציה. כמובן שפונקציה זו תהיה עם lock. כעת ב- **Main** צור 10 ת'ראדים ושלח אותם לפונקציה **CheckPrintMessage** (כמובן שהם יחכו ב- Wait... מכיוון שהתור ריק). כעת הוסף 10 הודעות לתור, באמצעות **AddMessage** (שלח לדגומא במחרוזת של ההודעה את הזמן הנוכחי). בין הוספה להוספה המתן 100 מילי שניות (שיהיה מתח...).
אם כתבת הכול נכון- אחרי כל הכנסה לתור, מיד תוצג לפניך ההדפסה... פשוט לא? 😊
** אתגר מכשלי: התוכנית לא "תעבד" את ההודעה ה- 11 מכיוון שיש רק 10 ת'ראדים. לכן דאג שתמיד יהיו לפחות 10 ת'ראדים פעילים (אולי בשלב ה- Enqueue?). נסה להכניס 100 הודעות ובדוק אם אכן היו 100 הדפסות... הצלחת? אם כן- הוכתרת כ- thread ninja!