מעבדה 2

אלי חדאד 207931536

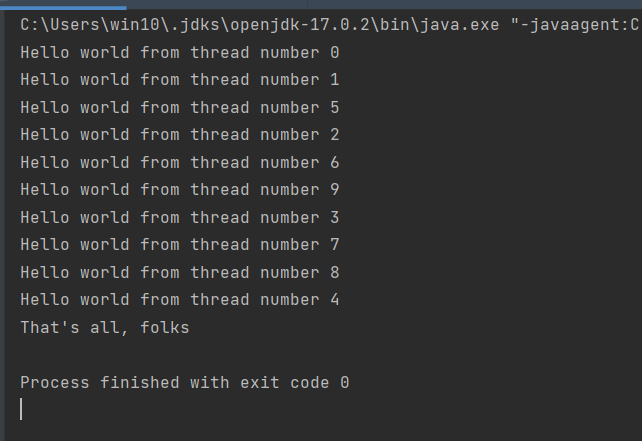
פייסל סעדיה 208336321

1)thread.start() יוצרת חוט חדש ואז החוט הזה קורה למתודה run אבל thread.run() לא יוצרת חוטים חדשים, רק מריצה חוט שכבר קיים .

2)

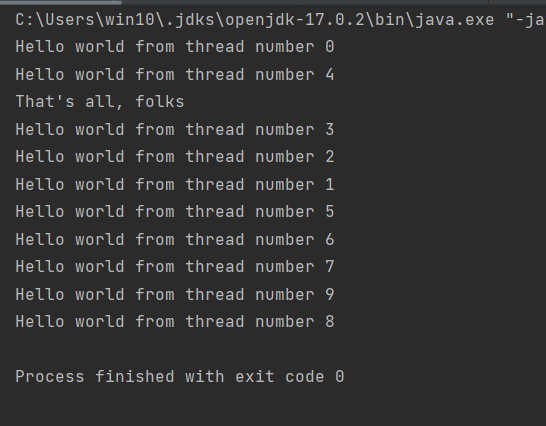
2.1)

התוכנית הדפיסה את מספרי החוטים בסדר שונה כל פעם כי כל החוטים עובדים במקביל, ובסוף בגלל פקודת join התוכנית מחכה עד שכל החוטים יסיימו ואז מדפיסה “that’s all , folks” .



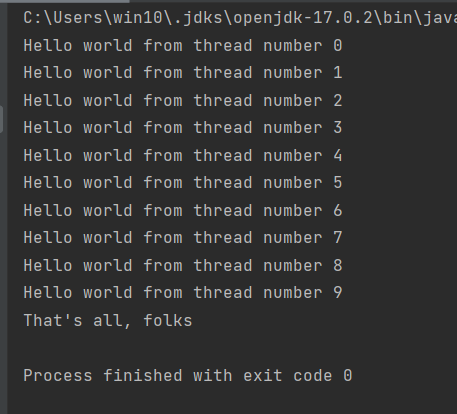
2.2)

הלולאה השלישית המטרה שלה היא לחכות עד שכל החוטים סיימו עבודתם ואז ממשיכה התוכנית לכן אם מוחקים את הלולאה התוכנית עלולה להדפיס את השורה האחרונה לפני שמדפיסה את כל מספרי החוטים .



2.3) אם מבצעים פקודת join אחרי start באותה לולאה אז התוכנית תמתין עד שהחוט הנוכחי ימות לפני שתייצר חוט חדש

ולכן התוכנית תדפיס את המספרים בסדר הנכון.

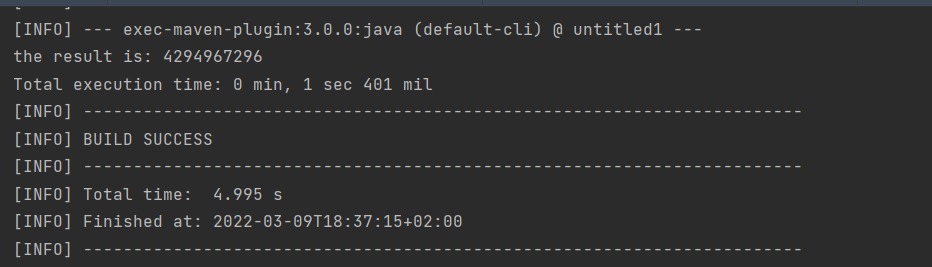


2.4) מצב של deadlock , כי אנחנו מבקשים לחסום החוט עד ש אותו חוט מסיים עבודתו, שזה לעולם לא יקרה.

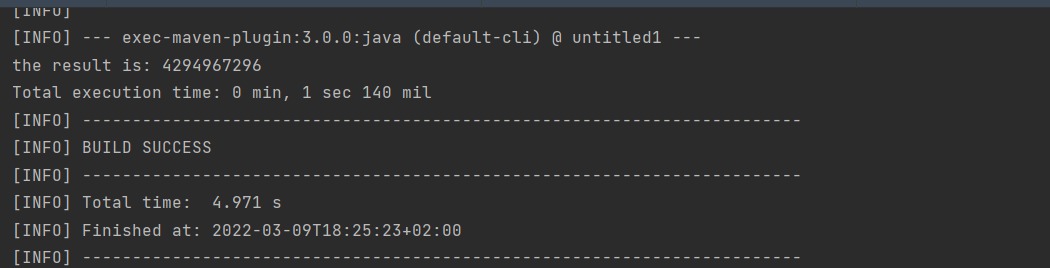
3)

3.1) כן, עם חוטים הקוד יהיה יותר מהיר ויעיל, כי החוטים רצים במקביל וזה חוסך זמן ומאמץ.

Sum :



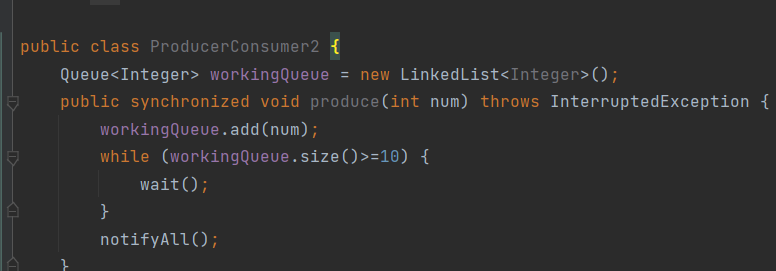
SumThreads :



3.2) לא יצא לנו הבדל של כמה milliseconds כל פעם, כי החוטים רצים במקביל ויכול להיות שכל פעם הם שצים בצורה שונה לכן ישפיע קצת על זמן הריצה, אבל ההבדל קטן מאוד.

3.3) צירפנו את קובץ SumThreads.java לקובץ ה zip.

4) הוספנו לולאת while שבודקת אם יש 10 איברים בתור אם כן אז קוראים ל wait .



גם הוספנו קובץ producerConsumer2.java לקובץ ה zip .