# צד שרת - חלק ב

## שאלה 1

## שאלה 1

1. הסיבה ששני המחרוזות שוות זה בגלל מנגנון שנקרא "String interning". מנגנון זה הוא תהליך שבו ה-JVM מאחסן רק עותק אחד של כל ערך מחרוזת ייחודי במאגר של מחרוזות הנקרא String pool. כאשר מחרוזת חדשה נוצרת, ה-JVM בודק קודם את מאגר המחרוזות כדי לראות אם הערך כבר קיים במאגר. אם כן, ה-JVM מחזיר את ההפניה למחרוזת הקיימת ב-pool במקום ליצור אובייקט String חדש. אם הערך לא קיים במאגר, ה-JVM יוצר אובייקט String חדש ומוסיף אותו למאגר, ולכן במקרה שלנו לשני המחרוזות יש אותו שם ולכן זה מחזיר אמת כי שניהם אכן מפנים לאותו מקום בheap.
2. שיטת hashCode() מחזירה ערך של מספר שלם המבוסס על תוכן האובייקט. במקרה שלנו שרשרנו לs1 את המילה "Deo" , ומכיוון שמחרוזת לא ניתנת לשינוי בjava, המחרוזת המקורית לא משתנה אלא נוצרת מחרוזת חדשה שניתן להתייחס אליו באמצעות s1. לכן, ערך קוד ה-hash של המחרוזת החדשה שנוצר לאחר השרשור שונה מערך קוד ה-hash של המחרוזת המקורית.
3. כאשר מנסים להשוות שתי מחרוזות באמצעות האופרטור "==" זה משמש רק להשוואת **ההפניות** של אובייקטי המחרוזות, ולכן name1==name2 יחזיר false. אולם, ההשוואה השנייה תחזיר true כי הפעולה Intern() מחזירה הפנייה למחרוזת הנמצאת ב String pool. במקרה שלנו המחרוזת "Alex" כבר קיימת ב String pool (באמצעות name1) ולכן ההשוואה תחזיר true.  
   אם נפעיל שוב את אותה פקודה התוצאה עדיין תהיה true, כי ברגע שמחרוזת נכנסת לString pool , היא נמצאת שם למשך כל חיי היישום.
4. Thread pool עם thread בודד יעיל יותר ביחס לthread בודד במשאבי המערכת ושליטה טובה יותר בניהול שרשורים (במיוחד במצבים עם הרבה thread קצרים). הסיבה לכך היא ש thread בודד מוקדש למשימה מסוימת ומסתיים לאחר השלמת המשימה. אם נרצה ליצור משימה חדשה נצטרך ליצור thread חדש. לעומת זאת, אם משתמשים בthread pool הthread אינו נהרס לאחר השלמת המשימה, אלא הופך לזמין לטיפול במשימות אחרות.

## שאלה 2 / שאלה 3

בהנחה כללית שהמספרים שהמשתמש מכניס / המשתנה result לא עולים על MaxInt -

1. הערך המינימלי באופן תיאורטי של המשתנה result הוא 0, מכיוון שזהו הערך ההתחלתי של המשתנה (שורה 2 בקוד). הפעולה function מעלה את result כל פעם באחד בהתאם למשתנה המתקבל (numOfIterations), ואם אנחנו שולחים לפונקציה את המספר "0" (המספר המינימלי שמשתמש יכול להעביר), הפעולה לא תעלה בכלל את result ולכן הערך המינימלי התיאורטי הוא 0.
2. הערך המקסימלי באופן תיאורטי של המשתנה הוא numOfIterations בריבוע, מכיוון שהתוכנית מייצרת numOfIterations של threads ועל כל יצירה של thread היא עושה את הפעולה "function" שמעלה את המשתנה הסטטי result באחד במשך numOfIterations פעמים.
3. Thread safety - תוכנית עשויה להפעיל קוד במספר threads בו זמנית במרחב כתובות משותף שבו לכל אחד מאותם threads יש גישה למעשה לכל הזיכרון של כל thread אחר.