פיתוח תוכנה מתקדם 2 – סמסטר א' מועד א' תשפ"ב

תזכורת: כתובת מערכת הבדיקות: https://cktest.cs.colman.ac.il/

שם הקורס, PTM2 מועד א'. לאחר הורדת המבחן ממערכת הבדיקות. העתיקו את כל קובצי ה Java לתוך הפרויקט ב package בשם test.
במבחן זה 3 שאלות, חובה לענות על כל 3 השאלות ולהגיש למערכת הבדיקות במוד הגשה סופית לפני סוף המבחן.

שאלה 1 – תכנות מקבילי באמצעות ת'רדים (35 נק')

תזכורת כללית: גישה למחלקה שירשנו נעשית ע"י ***super.***

נתונה לכם המחלקה TaskerList כסוג של LinkedList שמחזיק בתוכו BlockingQueues של Runnables.
עליכם לממש את המתודות הבאות:

- המתודה pollAll – עבור כל BlockingQueue היא תייצר ת'רד חדש נפרד, שיריץ בזה אחר זה את כל ה Runnables שבתור שלו.
- המתודה stopRunning – תוביל לסגירה מיידית ומוחלטת של כל הת'רדים שנפתחו.
- בנוסף עליכם לדרוס את המתודה addLast אשר תאפשר הכנסה של BlockingQueue חדש לסוף הרשימה אך ורק אם stopRunning **לא הופעלה**.

בבדיקה של MainTrain1 ננסה להכניס מספר מסוים של BlockingQueues ל TaskerList כאשר כל BlockingQueue מכיל מספר Runnables. הציפייה היא שיחזרו ערכים (אם בכלל) בהתאם להנחיות שנכתבו לעיל.

יש לציין כי ב"מוד הגשה" יהיו בדיקות עמוקות יותר עם פרמטרים שונים.

שאלה 2 – סנכרון (30 נק')

בקובץ של Count.java נמצאת מחלקה פשוטה שמחזיקה מערך מסוג int, יש לה שלוש מתודות:

- המתודה inc שמגדילה את כל הערכים של אברי המערך ב-1
- המתודה dec שמקטינה את כל הערכים של אברי המערך ב-1
- המתודה get שבהינתן index תחזיר איבר במערך במקום ה index

הביטו ב MainTrain2.java, אנו מפעילים שני ת'רדים שמריצים את אותו ה Runnable. ה Runnable קורא d פעמים השווה בין מיליון ל 2 מיליון פעמים באופן אקראי למתודה inc של אותו מופע של Count. הציפייה היא שהערך בכל איבר במערך של count יהיה שווה בדיוק ל d2 לאחר ששני הת'רדים סיימו את פעולתם. אולם, כפי שלמדנו ,בכל ריצה נקבל ערך שונה בגלל שהפעולה inc אינה פעולה אטומית. עליכם לערוך את Count כך שאכן יתקבל הערך 2d ב MainTrain2.

עבור המתודה dec, אותו הסיפור פשוט הפעולה ההפוכה מ inc.

הערה: **חל איסור לעשות שימוש ב synchronized.**

יש לציין כי ב"מוד הגשה" יהיו בדיקות עמוקות יותר עם פרמטרים שונים.

שאלה 3 – אופטימיזציות קוד (35 נק')

חישוב בשם Softmax מוגדר על ווקטור z של K מספרים ממשיים $(z_1, \ldots, z_K) \in \mathbb{R}^K$ כך:

$$\sigma(\mathbf{z})_i = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}} \qquad \text{for } i = 1, \ldots, K \text{ and } \mathbf{z} = (z_1, \ldots, z_K) \in \mathbb{R}^K$$

פונקציית ה Softmax לוקחת כקלט ווקטור z עם K מספרים ממשיים ומנרמלת אותו להתפלגות הסתברות המורכבת מ-K הסתברויות פרופורציונליות למעריכים של מספרי הקלט.

ל-softmax יש שתי מתודות עזר, הראשונה נקראת exp, בהינתן ערך z היא תחזיר את האקספוננט שלו. לפונקציית העזר השנייה קוראים expSum, בהינתן ווקטור היא תחזיר את סכום האקספוננט של כל איברי הווקטור.

דוגמא:

בהינתן הקלט: Vec = {1,2,3}

בווקטור החדש שיחזור מ softmax הערך במיקום ה 0 יחושב כך:

$$NewVec[0] = \frac{\exp\ (Vec[0])}{\exp(Vec[0]) + \exp(Vec[1]) + \exp\ (Vec[2])}$$

בקובץ BadCode.java תמצאו את softmax, המימוש הלא יעיל של הפונקציה הנ"ל.
עליכם לכתוב את הפונקציה softmaxOpt ב GoodCode.java כך שתהיה יעילה פי 1.5 מ softmax.

הערה: **אין להשתמש בפונקציות של BadCode.java.**

יש לציין כי ב"מוד הגשה" יהיו בדיקות עמוקות יותר עם פרמטרים שונים.

הגשה

עליכם להיכנס למערכת הבדיקות בכתובת: https://cktest.cs.colman.ac.il/ ולהגיש ל PTM2 מועד א' את הקבצים TaskerQueue.java, Count.java, GoodCode.java.

בכל הגשה יש להגיש את כל הקבצים (להתייחס לפלט רק של השאלות שעניתם עליהן).

ניתן להגיש במוד אימון ובמוד הגשה כמה פעמים שתרצו עד לסוף המבחן.

בסוף המבחן יש להגיש במוד הגשה ואז במוד הגשה סופית. אחריה תקבלו מס' אסמכתא בין ארבע ספרות.
לאחר הגשה במוד הגשה סופית לא תוכלו להגיש יותר.

בהצלחה!