

תכנות מתקדם 2: 89-211 – מועד א' תשע"ו

זמן המבחן: שעותיים, יש לענות על 6 מתוך 6 שאלות, בגוף השאלון בלבד. חומר סגור.

בקיאות

שאלה 1: (16 נק')

נתונה לנו מערכת לניהול אותות מצוקה (signals) עבור קבוצת רובוטים. במחלקה SignalManager קיימת המתודה sendAck() שפועלת באופן הבא: כל עוד אות המצוקה נקלט - המשך לשלוח Ack לרובוט שבמצוקה.

אליס העלתה שקיים חשש ש sendAck() תהיה מתודה חוסמת במקרה זה Ack לא נקלט אצל הרובוט שבמצוקה - הרי הוא ימשיך לשלוח את אות המצוקה כל עוד הוא לא קיבל Ack...

בוב טוען שלכן המחלקה SignalManager צריכה להיות Active Object. שכן, אם הגיעו מספר אותות מצוקה במקביל ועבור כל אחד הפעלנו את sendAck(), אז היות ו sendAck() הופכת להיות אסינכרונית אז לא תהיה בעיה לשלוח לשאר הרובוטים Ack, גם כאשר אחד מהם לא מצליח לקבל את האות.

אליס טוענת שבוב טועה, והמחלקה SignalManager צריכה לעשות שימוש ב Thread Pool.

נמקו מדוע האחד צודק והשני טועה.

(8 נק') מי צודק? _____, מדוע? _____

(8 נק') מי טועה? _____, מדוע? _____

שאלה 2: (12 נק')

הקיפו בעיגול את התשובות הנכונות:

- static design עלול במקרים מסוימים להוביל לקוד כפול
- נעדיף קוד שמתאפיין ב high coupling וב low cohesion
- ב REST נעשה שימוש ב SOAP אך לא ב WSDL
- Memcached היא טכנולוגיה המאפשרת linear scalability.

מיומנות עיצוב קוד (Design) וכתובת קוד

שאלה 3: (20 נק') נתון לנו הממשק `SignalReader` המגדיר את המתודה `read()` שמחזירה `double`. בנוסף, נתונות לנו המחלקות `RFSignalReader`, `HFSignalReader`, `UHFSignalReader` שמימשו את הממשק, כל אחת עבור סוג אחר של אות.

אולם, האותות המוחזרים ע"י מחלקות אלו הינם "רועשים". כלומר, בכל דגימה (הפעלה של `read`) הערך שמוחזר לנו נע מסביב לערך האמתי. לדוג' אם הערך האמתי הוא 88 אז חוזרים לנו ערכים מסביב ל 88.2, 88.3, 87.9 וכו'.

רעשים אלו צריך לסנן ולהחזיר את הערך המשוער כאמתי (לדוג' 88). המחלקות `MovingAverage` ו `KalmanFilter` מימשו כל אחת אלגוריתם שונה לסינון רעשים במתודה `double filter(double[] v)`.

עליכם **לשרטט** תרשים מחלקות (class diagram) ב **UML**, המציג עיצוב שמאפשר לסנן את הרעשים המוחזרים ע"י `RFSignalReader` ודומיה. בפרט:

- המחלקה `FilterReader` תהיה אחראית לסינון רעשים
- הקליינט צריך להכיר רק את `SignalReader` (4 נק')
 - כלומר אין זה משנה לו מהו הסוג הספציפי של `SignalReader`, ועדיין נרצה לאפשר לו לקרוא אותות מסוננים.
- נרצה יכולת לסנן כל סוג של אות (4 נק')
- ובאמצעות כל אלגוריתם סינון (6 נק')
- נרצה שהעיצוב ישמור על עיקרון ה `open / close` (6 נק')

תשובה: (על השרטוט להכיל את כל המחלקות הרלוונטיות + הערות ע"פ הצורך)

תשובה:

This image shows a blank sheet of white paper with horizontal ruling lines. The lines are evenly spaced and run across the width of the page. There are no margins, text, or other markings on the paper.

[illegible]

שאלה 5: (20 נקודות)

נתונה המחלקה `MySorter<T>` (כקוד סגור שלא ניתן לשינוי) המאפשרת מיון של אובייקטים מסוג פרמטרי `T` במתודה `sort(T[] arr)`. כדי לקבוע את הקריטריון ההשוואה בין כל שני `T`-ים המתודה `sort` מפעילה (בעת צורך) את המתודה האבסטרקטית `abstract int compare(T t1, T t2)`. לצערנו זהו `static design`.

עליכם **לשרטט** תרשים מחלקות (class diagram) ב **UML**, שעושה שימוש חכם ב `MySorter<T>` (5 נק') אך מציג `dynamic design` עבור בעיית המיון (15 נק').

תשובה:

בהצלחה!

[illegible]

[illegible]