

# אודות המבחן – תכנות מתקדם 2

## פרטים טכניים:

- זמן המבחן שעותיים
- 6 שאלות
- אין בחירה
- חומר סגור

ככלל, המבחן אודות חומר ההרצאות.

למבחן יש שלושה חלקים, בכל חלק 2 שאלות, ניקוד משתנה.

1. חלק הבודק בקיאות בחומר
2. חלק הבודק מיומנות עיצוב קוד
3. חלק הבודק מיומנות כתיבת קוד

## חלק הבקיאות

נועד לגרום לכם לחזור על כל החומר, שכן יש סיכוי לשאול על כל אחד מהנושאים שלמדנו.

שאלות פתוחות בסגנון:

- תארו מהי המטרה של consistent hashing, כיצד הוא פועל, באיזה הקשר למדנו עליו, האם הפונקציה f עונה על הדרישות?
- מהי המוטיבציה להשתמש ב Hadoop? מה לא מספיק טוב במסדי נתונים?
- מהו ההבדל המהותי בין MVC ל MVP? על איזו סוג של לוגיקה אחראית השכבה המקשרת?
- כיצד היא לא free loader? מה מיוחד ב MVVM?
- מהי סקלביליות לינארית? האם הפתרון שהציע בוב תורם לסקלביליות?
- תארו את ארכיטקטורת SOA על כל מרכיביה, במה שונה REST?
- מה ההבדל בין Active Object ל Thread Pool? עבור הנקודות שהעלתה אליס, האם כדאי להשתמש ב Active Object או ב Thread Pool?
- מדוע בגרסאות קודמות פיתחו באנדרואיד JVM חדש? נמך?
- ייתכנו גם שאלות פחות פתוחות (או תת סעיפים) בסגנון של נכון \ לא נכון או בחירה מרובה:
- שימוש ב web services תורם לסקלביליות של המיזם שלנו \ נכון \ לא נכון
- סמנו את החסרונות של Service locator
  - ☒ השירותים הפוכים להיות קופסאות שחורות
  - ☐ הוא לא מאפשר אופטימיזציה של קוד בצורה אוטומטית
  - ☒ הוא צוואר בקבוק
  - ☒ סכנת אבטחה
  - ☐ יכולת ההפרדה בין דורש השירות למספק השירות היא נמוכה
  - ☒ שגיאות יתגלו בזמן ריצה במקום בזמן קומפילציה
  - ☒ קשה יותר לתחזוק
- שבצו את הרכיבים הבאים על תרשים הארכיטקטורה של המיזם שלנו

## חלק מיומנות עיצוב קוד

חלק זה יתמקד בעקרונות העיצוב (שיעור 2), ה architectural patterns וה concurrency patterns. שתי שאלות יתארו בעיה עיצובית כלשהי, ועליכם יהיה לספק פתרון עיצובי לבעיה ע"י השרטוט שלו ב UML.

שאלות לדוגמא:

- האובייקט A אחראי להחזיר את תוצאות החיפוש. ייתכן ונרצה לסנן חלק מהתוצאות בדרכים שונות, שאף עלולות להשתנות בעתיד. נקבו בשם ה design ושרטטו אותו.
- כיצד אפריד בין האלגוריתם לבין ההיוריסטיקות שהאלגוריתם עושה בהן שימוש? בכל ניסוי אני רוצה להריץ את אותו האלגוריתם אבל להזין לו היוריסטיקה אחרת.

## חלק מיומנות כתיבת הקוד

בחלק זה תצטרכו לממש קוד (ב C# או Java לבחירתכם) עבור מימוש העיצוב של החלק הקודם.

דוגמאות:

- לממש מחלקה בסגנון future עם מתודת get שמיישמת garded suspenssion
- לממש את הממשקים שעוזרים לי להפריד בין האלגוריתמים לבין הבעיות שאותן הם פותרים
- השלימו את הקוד הבא כדי לחבר בין ה View ל ViewModel

לא צריך לזכור API של הספריות השונות שלמדנו.

נהניתי מאד ללמד אתכם הסמסטר, תודה רבה על ההקשבה. ☺

עלו והצליחו!

אלי.