

הוראות הגשה אבן דרך 1

הורידו את המטלה מ PTM_project2 ו ex1.

עליכם להגיש שלושה דברים:

1. את הפרויקט שלכם ארוז כ PTM_projectTest.jar
2. את הקובץ TestSetter.java שעושה התאמות במידה וצריך
3. את הקובץ American.java העונה על 5 שאלות אמריקאיות

PTM_projectTest.jar

אבן הדרך הראשונה כוללת תשתית של מספר מחלקות וממשקים. עליכם לארוז את הקוד לקובץ בשם PTM_projectTest.jar. הקפידו לכלול את קוד המקור. כדי לוודא זאת פיתחו את הקובץ כ ZIP ותראו האם הקבצים עם הסיומת java נמצאים שם.

TestSetter.java

כמובן, עליכם להקפיד על העיצוב הנדרש באופן מלא, כולל שמות הטיפוסים והמשתנים. וודאו שהכל נמצא ב package בשם server_side. אם עשיתם זאת כהלכה לא תצטרכו לשנות דבר ב TestSetter.java. אם יש לכם שינוי כלשהו, תצטרכו למלא שם את ההתאמות.

לדוגמא, אם קראתם לממשק Server בשם Server1, מאיזושהי סיבה מיותרת, ולמחלקה MySerialServer בשם MyServer אז עליכם להזין במתודות המתאימות את משתנה ה class של המחלקה המתאימה:

```
dt.setServerInterface(Server1.class);  
dt.setServerClass(MyServer.class);
```

American.java

עליכם לענות על השאלות האמריקאיות הבאות באמצעות המחלקה American.java שם תמצאו מתודות q1 עד q5 המתאימות ל 5 השאלות הבאות. בכל מתודה עליכם להחזיר תו 'a','b','c' או 'd' בהתאם לסעיף שבחרתם לתשובה.

1. מדוע המשתנה cm במחלקה MyTestClientHandler הוא מסוג CacheManager ולא מסוג FileCacheManager?

- a. זו שמירה על עקרון Dependency Inversion
- b. נוכל להחליף בזמן ריצה בין סוגים שונים של CacheManager מבלי לשנות את הקוד ב MyTestClient
- c. תשובות a ו b נכונות
- d. אף תשובה אינה נכונה

2. רציתי ש Server תהיה מחלקה אבסטרקטית ולא ממשק. כך המשתנה ClientHandler יוכל להיות data member ולא פרמטר של המתודה open. מדוע בחרנו דווקא אחרת?

- a. החלטה שרירותית ותו לא.
- b. Server מתאר פונקציונליות של טיפוס ולכן צריך להיות ממשק.
- c. כדי להכריח הזרקה של ClientHandler הכנסנו אותו למתודה open, כך שלא ניתן לפתוח את השרת בלעדיו. נשים לב שזו המתודה היחידה שצריכה אותו, ולכן מספיק לקבל אותו כפרמטר ולא לקבוע אותו כ data member
- d. תשובות b ו c נכונות

3. מדוע Solver ו CacheManager הם משתנים של MyTestClientHandler ולא פרמטרים של המתודה ב ClientHandler?

- a. כי לא כל ClientHandler בהכרח צריך Solver ו CacheManager.
- b. זו טעות, צריך לשים אותם כפרמטרים של מתודה בממשק.
- c. אולי יותר ממתודה אחת תשתמש בהם ולכן הם צריכים להיות data members ולא פרמטרים, ובממשק לא ניתן להגדיר data members.
- d. תשובות a ו c נכונות.

4. כיצד באה ידי ביטוי השמירה על העיקרון של Open/Close?

- a. המשתנה c במתודה open הוא מהסוג של ClientHandler ולא מחלקה ספציפית.
- b. הוספה של שרת חדש, מנהל שיחה עם לקוח, פותר בעיות או מנהל cache חדשים דורשת אך ורק מימוש של ממשק ולא דריסה של קוד קיים.
- c. ניתן לראות שלכל ממשק או מחלקה יש תחום אחריות אחד ויחיד.
- d. לא ביצענו down casting באף מקום.

5. ממשק שיוורש ממשק אחר יוצר סכנה להפרה של איזה עיקרון של SOLID?

- a. Single Responsibility
- b. Interface Segregation
- c. Liskov
- d. Dependency Inversion
- e. Open Close

בהצלחה!!!