תכנות מכוון עצמים ושפת Java - תשע"ג קיץ - דף הנחיות ותרגילים

**יש להתעדכן באופן רציף בהודעות במערכת ה- highlearn לגבי הבהרות בדף המטלה**

דרישות הקורס

בקורס אין בחינה אך חובה להגיש את כל 3 התרגילים (יש בוחן). **יש להגיש את התרגילים בזוגות**, אך לא ביחידים / שלישיות / רביעיות וכו'. יש להקפיד על מעורבות אקטיבית של שני השותפים, אך על שני החברים לשלוט באופן מלא בכל חלקי התרגיל.

התרגיל הינו "מתגלגל" כך שכל חלק מתבססת על החלק הקודם.

את המטלה השנייה תמשיכו על פתרון התרגיל הראשון של מישהו אחר מהכיתה, לכן הקפידו שהתוכנית שאתם כותבים תהייה מצויינת!

בשקלול הציון הסופי ציון רכיבי המטלות וציון הפרוייקט זהים ויהיו שקלול ציון 3 חלקי הפרוייקט.

הנחיות להגשת התרגילים

התרגילים יוגשו לתיקיית dropbox משותפת עד תאריך הגשת המטלה, ויוצגו באופן פרונטלי למרצה בסוף הסמסטר לפי טבלת זמנים שתפורסם מראש. ינתן ציון בעבור כל אחת מהמטלות בנפרד.

הערה כללית: חלק מהעבודה בתרגילים היא קבלת החלטות בנושאים שאינם מפורטים במדויק. המטרה היא לתרגל את הנושאים המרכזיים הנלמדים בקורס, ולא לתפור מוצר לפי דרישות של לקוח, ולכן בכל מקום שלא מופיעה דרישה מדויקת – מוטל עליכם **לבחור בדרך ההגיונית ביותר שנראית לכם, או לשאול את הלקוח**. כדאי מאוד שהיא תראה גם הגיונית בעיניי ;-)

הנחיות כלליות (לכל חלקי הפרויקט)

* יש להימנע בכל מקרה משכפול קוד, מפונקציות ארוכות מדי ומבחירת שמות גרועים למחלקות לפונקציות ולמשתנים.
* יש לוודא קלט תקין מהמשתמש בכל שלב ולהחזיר הודאות שגיאה קריאות במידה והקלט אינו תקין.
* יש להקפיד להשתמש ב-modifiers בצורה נבונה. מחלקה שלא אמורים לבנות אובייקטים שלה אמורה להיות מוגדרת כ-abstract,
* קבועים יש לסמן כ-static final public (אין להשתמש בקבועים מספריים בגוף התוכנית ללא הגדרה כנ"ל),
* משתנים של המחלקה רצוי כמעט תמיד להגדיר כ-private, וכו'.
* יש להקפיד על מוסכמות בסגנון הכתיבה –
* שמות מחלקות יתחילו באות גדולה,
* שמות חבילות, משתנים ופונקציות באות קטנה,
* שמות קבועים יהיו מורכבים רק מאותיות גדולות,
* יש להקפיד על אינדנטציה נכונה, וכו'.
* יש לדאוג שבאף מקרה לא ייזרק Exception שלא טופל ושייגרום לתעופה של התוכנית.
* יש להקפיד על קוד פשוט ונקי.
* ינתן דגש על תכנות מכוון עצמים, ולכן שימו לב מי האובייקט האחראי על פעולה מסויימת, כלומר, אם יהיה באג, לכותב של איזו מחלקה יפנו באופן אינטואיטיבי.
* ומניסיון העבר: זהו קורס בשפה מכוונת עצמים, לכן הקפידו להשתמש בכל מאפיניה כפי שלמדתם בעבר.

פרוייקט – "ניהול תחנת דלק"

מטרת התרגיל היא לממש מערכת המדמה מצב התנהלות של תחנת דלק.

בתחנת דלק יש 4 סוגי אובייקטים:

1. **מאגר דלק מרכזי**. מכיל את הדלק של כל התחנה. כאשר כמות הדלק בו יורדת מ- 20% מסה"כ תכולתו הפוטנציאלית, יוצאת התרעה. [עליכם להחליט למי וכיצד]
2. **משאבת דלק.** תפקידה לספק דלק למכוניות שמתחברות אליה.
   1. משאבה יכולה לשרת מכונית אחת בלבד בכל רגע נתון. MUTEX
   2. על מנת שהמשאבה תעבוד, מאגר הדלק המרכזי צריך להיות לא ריק. Event ברגע שהמאגר ריק משבית את כל המשאבות.
   3. כאשר ממלאים את מאגר הדלק המרכזי, התחנה מושבתת ולא ניתן למלא דלק במשאבות השונות. Event ברגע שהמאגר מתמלא משבית את כל המשאבות.
3. **בית קפה.** בבית הקפה יש כמות מסוימת של קופות שיכולות לתת שירות בו-זמנית ללקוחות.
   1. לקוח שנכנס או מסתובב ומחפש מה לקנות
   2. או עומד בתור לתשלום,
   3. או מקבל שירות מהקופאית (ממומש במחלקה קופה)
   4. או יושב ושותה את הקפה שלו.
4. **מכוניות/נהגים.**
   1. נהג יכול להיכנס לתחנת הדלק על מנת למלא דלק.
   2. או לשתות קפה.
   3. נהג אינו יכול לשתות קפה בזמן שמכוניתו מתמלאת בדלק (כלומר, אין הפרדה בין נהג למכונית). LATCH
   4. נהג יכול להיכנס לבית הקפה בזמן שהוא ממתין למשאבה,
      1. ויודיעו לו כאשר תורו יגיע. (להאזין לEVENT מהמשאבה שהוא מחכה בתורה )
      2. במקרה זה הנהג יוכל לבחור האם להישאר בבית הקפה ולוותר על תורו למשאבת הדלק, או לצאת מיד החוצה ולמלא את הדלק במכוניתו.
      3. יציאה מיד החוצה מתאפשרת אך ורק אם הלקוח אינו משלם ברגע זה.
      4. כאשר נהג מחכה בתור, הוא מחכה בתור למשאבה ספציפית.

לצורך הפשטות, אנו מנהלים תחנת דלק אחת, אך יש לכתוב תשתית שתאפשר הוספת תחנות נוספות בהמשך. *לא סטטית ולא סינגלטון*

בסיום יום עבודה (סגירת תחנת הדלק), יש לשמור את כל הנתונים ולהציג דו"ח על תזרים הכספים בתחנה: כמה כסף נכנס מתדלוקים וכמה כסף נכנס מפעילות בבית הקפה.(להוסיף פונקציה לכל אוביקט שתפקידה לדווח על הכנסות למחלקת האבא)

שימו לב: כל הפעולות אמורות לקחת זמן, ולכן יש לדמות זאת ע"י פקודת sleep שזמנה יוגרל ע"י המערכת.

*ללמוד XML ולקרוא*

**קונפיגורציה:**

מאחר ובתוכנית קונסול קשה להריץ תוכנית אינטראקטיבית מהירה, הדמיית חלק מפעולות התחנה תהייה באמצעות קובץ קונפיגורציה בפורמט XML, שיכיל מידע לגבי מיכל הדלק המרכזי, המשאבות, בית הקפה והנהגים.

דוגמא לקובץ הקונפיגורציה:

<?xml version=*"1.0"* encoding=*"UTF-8"*?>

<GasStation numOfPumps=*"4"* pricePerLiter=*"7.82"*>

<MainFuelPool maxCapacity=*"500"* currentCapacity=*"220"* />

<CoffeeHouse numOfCashier=*"3"* />

<Cars>

<Car id=*"111"*>

<WantsCofee locationInCoffeeHouse=*"around"* amountToPay=*"34"* />

<WantsFuel numOfLiters=*"47"* pumpNum=*"2"* />

</Car>

<Car id=*"333"*>

<WantsFuel numOfLiters=*"42"* pumpNum=*"1"* />

</Car>

<Car id=*"222"*>

<WantsCofee locationInCoffeeHouse=*"inLine"* amountToPay=*"52"* />

</Car>

<Car id=*"444"*>

<WantsCofee locationInCoffeeHouse=*"inLine"* amountToPay=*"34"* />

<WantsFuel numOfLiters=*"35"* pumpNum=*"1"* />

</Car>

</Cars>

</GasStation>

במידה וחסרים נתונים בקונפיגורציה, אתם רשאים להוסיפם לקובץ.

**לוגים:**

כל הודעת לוג תציין את הזמן בו נכתבה בפורמט dd/MM/yyyy HH:mm:ss

1. יש לייצר קובץ לוג לכל המערכת ובה יתועדו כל הפעולות שיתוארו בסעיפים הבאים
2. יש לייצר קובץ לוג נפרד לכל לקוח המתעד את הפעולות הקשורות אליו
3. יש לייצר קובץ לוג נפרד לכל משאבה המתארת את כל הפעולות הקשורות אליה
4. יש לייצר קובץ לוג נפרד לבית הקפה המתעד את כל הפעולות הקשורות אליו

שימו לב לעבוד עם logger אחד במערכת ולעבוד עם handler נפרד עבור כל קובץ לוג.

**פעולות המערכת:**

כתוב תפריט בקונסול המאפשר את הפעולות הבאות:

1. הוספת מכונית: למשאבה או לבית הקפה
2. העברת מכונית מסטטוס אחד בבית הקפה לאחר (למשל מחיפוש מוצרים לקופה)
3. מילוי דלק במשאבה המרכזית

חלק 1 – תוכנית console

**הגשה לתאריך 05.08.2013**

מטרת חלק זה היא לממש כפרוייקט קונסול את כל המתואר בתיאור הפרוייקט.

כל הנתונים יקראו מקובץ הקונפיגורציה (או יוגרלו במידה והמידע חסר בקונפיגורציה) במהלך התוכנית. כמו כן, יש להוסיף תפריט אינטאקטיבי לביצוע פעולות בזמן הריצה כפי שצוין לעיל.

**בונוס 2 נקודות לציון בקורס:** עבודה עם source control לאורך כל התרגיל.

חלק 2 – GUI ושמירת ל- DB – יש להמשיך מפתרון של מישהו אחר

**הגשה לתאריך 26.08.2013**

בתרגיל זה יש לממש GUI עבור המערכת שכתבתם בתרגיל 1, לשמור את המידע בבסיס הנתונים ולהציג דו"חות כפי שתוארו לעיל.

בתרגיל זה הנתונים יתקבלו באופן אינטראקטיבי דרך כפתורים, בנוסף למידע שיקרא מהקונפיגורציה כפי שביצעתם בחלק הראשון של התרגיל.

יש ליישם ולהציג באופן ויזואלי את כל פעולות המערכת שתוארו לעיל.

חלק 3: ממשק משתמש WEB'י, שרת-לקוח, SPRING ו- AOP ו- Web Services – ניתן להמשיך מהפתרון שלכם של תרגיל 2 או של מישהו אחר

**הגשה לתאריך 31.10.2013**

1. יש לכתוב WebService אשר דרכו ניתן לקבל מידע כמה מכוניות ממתינות למשאבה מסוימת. - טל
2. יש לכתוב ממשק WEB עבור הלקוח שיתמוך בפעולות הבאות: ירמי
   1. הוספת מכונית ושיבוצה למשאבה ו/או לבית הקפה.
   2. העברת מכונית בין מצבים שונים בבית הקפה.
   3. העברת מכונית מבית הקפה למשאבה.
3. יש לכתוב מנגנון Client-Server. דרך הקליינט נוכל להוסיף מכונית חדשה ולמקם במשאבה או בבית הקפה.
4. באמצעות מנגנון ה- Spring, יש לאפשר להגדרה בקונפיגורציה האם רוצים לשמור את המידע לבסיס הנתונים באמצעות מנגנון ה- JDBC (שמומש בתרגיל הקודם) לבין מנגנון ה- JPA.
5. את כל כתיבת הלוגים יש להמיר לעבודה עם AOP. –מי שמסיים ראשון

תזכורת: כדי שמשתנה יעבור ב- aspect הוא צריך להיוצר ע"י bean בקונפיגורציה של ה- aspect.

**עבודה פוריה, מלמדת ונעימה!**

**שאלות שלנו לקרן:**

1. numOfPumps=*"4"*– למה צריך להגדיר את מס' המשאבות? מימשנו כוקטור.
2. הוספנו מחלקה של קופה.
3. האם לנהל תור לכל התחנה או לכל משאבה
4. למה אני חייב להוסיף כתיבה ללוגר ב-INIT וגם לפני הFORMATTER. למה לא יורד לי שורה.
5. האם חייב שהמכונית תרוץ בפונקציה על מנת לדמות הסתובבות בחנות ואז להריץ כמה תהליכונים יחד עבור כל המכוניות?? צריך לדמות זמן של הסתובבות.
6. האם להוסיף תור של מסתובבים בבית הקפה בנוסף לתור הרגיל צריך לדמות זמן של הסתובבות
7. הנחה - יש תור אחד לקופות. האם זה בסדר? כן
8. האם לנעול על OBJECT אחד את כל התהליכים או שצריך לעשות נעילה פר מחלקה.? עשיתי אחד לכולם כי אני לא רוצה ששום תהליך שבשום מחלקה יפריע לי.
9. האם אני יכול לעשות SYNC על מכונית – ראה קופה. לנעול על המשאבה\קופה
10. האם להשתמש בPROTECTD (רק ליורשים) או בPUBLIC עבור הפונקציות?????.
11. אם עושים WAIT מישהו יכול להיכנס לתהליך