

מבוא לתכנות ב JAVA - סמסטר ב 2015

תרגיל מספר 4

תכנות מונחה עצמים

בעבודה זו נתרגל תכנות מונחה עצמים ורשימות. עליכם ליישם מספר מחלקות ובחלקם תשתמשו במבנה של רשימה. מבנה הרשימה נלמד בהרצאות ואתם יכולים להשתמש **אך ורק** בשיטות השונות שנלמדו בהרצאות. עליכם ליצור מחלקה בשם List ובמחלקה Node המשתמשת אותה כדי ליישם את מבנה הרשימה.

שימו לב שאתם כותבים רק חלק מתוכנית שלמה. המחלקות שתכתבו לא יכללו פונקציה main(). כדי לבדוק את העבודה שלכם כדאי לכתוב מחלקה נוספת הכוללת פונקציה main(). לרשותכם גם קובץ בדיקה לדוגמה, מחלקה הכוללת main() ואת הפלט הצפוי עבור הריצה שלה להשוואה.

במסעדת האחיינים החליטו להשתמש במערכת ממוחשבת לניהול הזמנות של הסועדים. המערכת כוללת רשימות שונות: תפריט אלקטרוני המכיל רשימה של המנות, רשימת הזמנות לכל שולחן, רשימת המנות שהשף צריך להכין וקופה רושמת עבור כל ההכנסות ושומרת נתונים שניתן לנתח לפי מספר שולחן והפריטים שבתפריט.

חלק א מחלקת המנות

כתבו מחלקה בשם MenuItem המתארת פריט אחד מתוך התפריט. שדות המחלקה כוללים את שם הפריט, מספר הפריט ומחיר. ניתן להניח שמחיר הפריטים הוא מספר שלם ולכן יש שימוש בטיפוס int. השדות במחלקה הם:

```
public class MenuItem{  
    private String name;  
    private int id;  
    private int price;
```

שיטות המחלקה:

```
public MenuItem(int id,String name,int price)
```

בנאי היוצר אובייקט וממלא את השדות name, id ו price.

```
public String toString()
```

שיטה המחזירה מחרוזת הכוללת את מספר הפריט שם הפריט והמחיר שלו. דוגמה למחרוזת:

“6 Hamburger Price: 57”

כאשר 6 הוא מספר המנה ו 57 הוא המחיר.

דגשים לחלק א

- אין להוסיף שדות נוספים למחלקה. ניתן להוסיף שיטות למחלקה כרצונכם, וגם שיטות המחזירות את ערכי השדות (getters).

חלק ב' מחלקת פריט מהזמנה

כתבו מחלקה בשם Order המכילה פריט הזמנה אחד. המחלקה מחזיקה פריט מתוך התפריט ומוסיפה לו שדה של כמות עבור הזמנה מרובה של אותו פריט. השדות במחלקה הם:

```
public class Order{  
    private int table;
```

```
private int ordered;
private int served;
private MenuItem item;
```

השדה table שומר את מספר השולחן, השדה ordered שומר את מספר המנות שהוזמנו עבור פריט זה, השדה served שומר את מספר המנות שהוגשו עבור פריט זה. שיטות המחלקה:

```
public Order(int table,int count,MenuItem item)
```

בנאי היוצר אובייקט וממלא את השדות served, item, table ו-ordered. שדה count מאופס בעת ההזמנה, ושדה ordered מאותחל לפי המשתנה count.

```
public void serve(int count)
```

שיטה שנועדה לעדכן הגשה עבור פריט זה. השיטה מקבלת ערך count שמתאר כמה מנות נוספות מהפריט הוגשו לשולחן. במידה והערך שהתקבל גבוה מכמות המנות שיש להגיש, מעודכן השדה served לפי הערך המקסימלי האפשרי, כלומר לערך של ordered.

```
public String toString()
```

שיטה המחזירה מחרוזת הכוללת את הכמות, שם הפריט, המחיר של הפריט והמחיר הכולל של הפריט. דוגמה למחרוזת:

“6 Hamburger Price: 57 Table: 5 Served: 2 Total: 114”

כאשר 6 הוא מספר הפריט, 57 הוא מחיר הפריט, 2 הוא מספר המנות שהוגשו, 114 הוא המחיר הכולל עבור הפריט כלומר 57×2 .

דגשים לחלק ב

- אין להוסיף שדות נוספים למחלקה. ניתן להוסיף שיטות למחלקה כרצונכם, וגם שיטות המחזירות את ערכי השדות (getters).

חלק ג' מחלקת הקופה

כתבו מחלקה בשם CashRegister המרכזת את הרשימות השונות שנחוצות במסעדה. השדות במחלקה הם:

```
public class CashRegister{
    private List<MenuItem> menu;
    private List<Order> chefsQue;
    private List<Order> pending;
    private List<Order> paid;
```

השדה menu מחזיק רשימה של התפריט. השדה chefsQue מחזיק רשימה של פריטים שעל השף להכין. נתונים מתווספים לרשימה זו רק בסופה כך שנשמר הסדר הכרונולוגי. השדה pending מחזיק רשימה עבור כל הפריטים שהוזמנו ובהמתנה בכל השולחנות. השדה paid שומר רשימה שאוספת את הנתונים לגבי כל מה שהוזמן ושולם. הרשימות pending מסודרת לפי מספר השולחן ואחר כך לפי מספר המנה. והרשימה paid מסודרת לפי מספר המנה ואחר כך לפי מספר השולחן. שיטות המחלקה:

```
public CashRegister()
```

בנאי היוצר אובייקט וממלא את השדות ויוצר רשימות ריקות עבורם.

```
public void addMenuItem(int id,String name,int price)
```

שיטה המוסיפה פריט לרשימה menu.

```
addOrder(int table,int count,int itemId)
```

שיטה המוסיפה הזמנה למערכת לפי הנתונים השונים. השיטה מוסיפה צומת חדשה לסוף הרשימה של השף. השיטה גם מוסיפה צומת לרשימת ההזמנות בהמתנה. במידה וקיים כבר צומת עם אותו מספר שולחן ואותו פריט, לא נייצר צומת חדשה, אלא פשוט נעדכן את השדה של כמות ההזמנות. במידה והפריט לא נמצא ברשימת התפריט לא קורה דבר.

```
public void printCheck(int table)
```

שיטה המדפיסה את החשבון של שולחן מסויים. החשבון כולל רק את המנות שהוגשו לשולחן ולא מציין כמה מנות לא הוגשו.

השיטה גם מוחקת את האיברים הרלוונטיים ברשימות של השף ורשימת ההמתנה. כלומר יש למחוק את כל האיברים ברשימות **pending** ורשימת השף עם אותו מספר שולחן.

השיטה גם מעדכנת איברים ברשימה **paid**. במקרה זה יש לשמור גם פריטים שהוגשו וגם פריטים שלא הוגשו, כדי להדפיס בעתיד את הסטטיסטיקה של המסעדה.

במידה וקיים איבר ברשימה **paid** עם אותו מספר שולחן ואותו מספר מנה יש לעדכן את השדות שלו.

במידה ולא קיימים האיברים המתאימים ברשימה **paid** יש ליצור אותם.

public void printChefsQue()

שיטה המדפיסה את רשימת השף ומציינת רק כמה מנות טרם הוגשו. ההדפסה מתבצעת לפי סדר כרונולוגי של כל ההזמנות שנמצאות ברשימה. שימו לב, ייתכנו חזרות של איברים עם אותו מספר שולחן ומספר מנה, כך שאחת ההזמנות בוצעה במועד מאוחר יותר.

public void serve(int table,int count,int itemId)

שיטה המעדכנת את הרשימות השונות כאשר **table** מתאר את מספר השולחן, **count** את מספר המנות שיוגשו, ו **itemId** את מספר המנה. השיטה מעדכנת את שתי הרשימות **pending** ו **chefsQue**.

עדכון הרשימה **pending** יתבצע תוך עדכון פריט אחד ברשימה.

עדכון הרשימה **chefsQue** יכול לעדכן מספר פריטים כאשר הרשומה הראשונה שנמצאה כוללת מספר מנות להגשה הנמוך ממספר המנות שהתקבל בשיטה. לכן למעט מחיקת איבר זה מהרשימה יש לחפש ברשימה האם יש עוד איברים עם אותה מנה ואותו מספר שולחן ולעדכן אותם לפי מידת הצורך.

public void printMenu()

שיטה המדפיסה את התפריט. התפריט מודפס בסדר עולה לפי ערכו של השדה **id**.

public void printReport()

שיטה המדפיסה דו"ח הכולל את כל הרשומות של הרשימה **paid**. הרשימה מדפיסה עבור כל פריט את מספר המנה, שם המנה, מספר המנות שהוגשו, המחיר הכולל, ומספר המנות שלא הוגשו. השיטה גם תדפיס בשורה האחרונה את סך הכל ההכנסות.

סדר ההדפסה של רשימה זו הוא לפי מספר הפריט ואחר כך לפי מספר השולחן. דוגמה לשורה מתוך הדו"ח:

9 Fish Patties Price: 21 Table: 5 Served: 1 Total: 21 Not Served: 1

ראו את התכנית לדוגמה לפלט הכולל מספר שורות ואת סך הכל ההכנסות.

דגשים לחלק ג

- שימו לב שיש לשמור על סדר מסויים בכל פונקציה של הדפסה. מומלץ (אך לא חובה) לשמור על סדר הרשימות השונות לפי סדר הדפסתן.
- אין להוסיף שדות נוספים למחלקה. ניתן להוסיף שיטות למחלקה כרצונכם, וגם שיטות המחזירות את ערכי השדות (getters).

הערות חשובות לגבי התרגיל:

- עליכם להגיש את הקבצים הבאים:

List.java
Node.java
MenuItem.java
Order.java
CashRegister.java

- שימו לב שבקבצים שאתם מגישים אין פונקציה בשם **main()**.

- על המחלקות List ו Node להכיל את השדות והשיטות שנלמדו בהרצאה.
- יש להגיש את הקבצים בתוך קובץ עם סיומת ".zip".
- ע"מ להקל על המשימה ולהבטיח שלא יהיו תקלות בשמות עליכם להוריד קבצי שלד של כל המחלקות הנ"ל.
- לקבצי השלד מצורפת מחלקה עם שיטת main לבדיקת המחלקות שאתם בונים, ומצורף גם הפלט שאמור להיות מוצג במהלך ריצת התכנית.
- הגשת התרגיל הינה חובה.
- תרגיל אותו לא ניתן להדר (לקמפל) או להריץ יקבל ציון נכשל.
- יש לדייק באופן כתיבת שמות השיטות וחתימותיהן כפי שתואר כאן. הבדיקה אוטומטית ומסתמכת על כך שהשמות הינם כפי שתוארו.
- פניות בנוגע להגשה באיחור, מילואים, מחלה וכד' יש להפנות למרצה הקורס.
- אנא הימנעו מהעתקות, אל תנסו אותנו זה לא נעים לשני הצדדים קראו על [Academic Dishonesty](#), לפני שאתם מתחילים את העבודה.

בהצלחה!