



### תרגיל בית מס' 2 (חובה): מחלקות ואובייקטים

- המועד האחרון להגשה ומספר הסטודנטים המקסימלי בקבוצה מופיע באתר הגשת העבודות.
- ההגשה דרך אתר הגשת העבודות בלבד.
- מותר להתייעץ וללמוד יחד עם החברים, אבל על כל קבוצת מגישים לכתוב ולהריץ את הקוד, ולערוך את התרגיל להגשה לבד. בשום פנים ואופן אין להעביר קבצי קוד או קבצי הגשה בין הקבוצות. העתקה אפילו של חלק מהתרגיל עלולה לפסול את התרגיל כולו.
- הגשת התרגיל מהווה הצהרה שהמגישים לא העתיקו מהקבצים של האחרים בתהליך הכנת התרגיל.
- חשוב לוודא שכל קבצי ההגשה עוברים קומפילציה ב-Eclipse. הפתרון שלא עובר קומפילציה לא יקבל ציון.
- יש לארוז את כל הקבצים של קוד המקור לקובץ Zip אחד ולהגיש דרך הגשת העבודות. **חוץ מהקבצים המפורטים בדרישות התרגיל אין לכלול בהגשה שום תיקיה או קובץ נוסף.** הגשה בפורמט לא נכון עלולה לגרום לעיכוב בבדיקה ולהורדת נקודות.
- בתחילת כל קובץ עם קוד המקור אמור להופיע בלוק הערות עם מספר התרגיל, **שמות ותעודות זהות של המגישים** ושם הקובץ. אי-עמודה בדרישה זו עלולה לגרום להורדת הנקודות.
- עם שאלות ובידורים לגבי התרגיל נא לפנות דרך הפורום באתר הקורס. כך כל הסטודנטים של הכיתה יקבלו תועלת מהשאלות ומהתשובות. עם שאלות שלא קשורות לכל הכיתה נא לפנות דרך פניה למרצה באתר הקורס.

### רשימת קבצי ההגשה של התרגיל:

1. CheckingAccount.java – פתרון של שאלה מס' 1.
2. CheckingAccountTester.java – פתרון של שאלה מס' 2.
3. TestResults.txt – פלט של בדיקות היחידה בשאלה מס' 2.
4. Circle.java – פתרון של שאלה מס' 3.
5. Rectangle.java – פתרון של שאלה מס' 4.
6. ShapesTester.java – פתרון של שאלה מס' 5.
7. ShapesTesterOutput.txt – פלט של בדיקות היחידה בשאלה מס' 5.

## 1. מימוש של מחלקה CheckingAccount (25 נק')

כתבו מחלקה CheckingAccount לפי הדרישות הבאות.

**קבועים.** במחלקה CheckingAccount יש להגדיר את הקבועים הבאים:

א. עמלת פעולה ברירת מחדל. כל פעול בחשבון מחויבת בעמלה. קבוע `DEFAULT_TRANSACTION_FEE` מגדיר את עלות העמלה ברירת מחדל. עמלה ברירת מחדל היא 1.75 ש"ח לפעולה. הקבוע מוגדר כ- כיוון שיש רק עותק אחד של הקבוע.

```
static final double DEFAULT_TRANSACTION_FEE = 1.75;
```

ב. יתרת פתיחת חשבון ברירת מחדל. הקבוע `DEFAULT_INITIAL_BALANCE` מגדיר את היתרה ברירת מחדל בפתיחת החשבון. הערך של הקבוע הוא 0.

```
static final double DEFAULT_INITIAL_BALANCE = 0;
```

**משתנים.** יש להגדיר את המשתנים הבאים:

- ג. `accountNum` מספר חשבון (מחרוזת)
- ד. `customerName` שם לקוח (מחרוזת)
- ה. `balance` יתרה בחשבון (מספר עשרוני)
- ו. `transactionFee` עמלת פעולה (מספר עשרוני). כל פעולה בחשבון מחויבת בעמלה בסך `transactionFee`. למשל, אם העמלה עבור החשבון היא 1.5 ש"ח ובוצעה משיכה של 100 ש"ח, החשבון יחויב ב-101.5 ש"ח (משיכה + עמלת הפעולה).

**מתודות.**

- ז. בנאי שמקבל את מספר החשבון, שם הלקוח, יתרת הפתיחה ועמלת הפעולה.
- ח. בנאי נוסף שמקבל את מספר החשבון ואת שם הלקוח בלבד. עמלת הפעולה והיתרה בחשבון ייקבעו כברירת מחדל לפי הערכים של הקבועים `DEFAULT_TRANSACTION_FEE` ו-`DEFAULT_INITIAL_BALANCE`.
- ט. המתודה `withdraw(double amount)` שמבצעת משיכת סכום `amount` וחייב העמלה.
- י. מתודה `deposit(double amount)` שמבצעת הפקדת סכום `amount` וחייב העמלה.
- יא. מתודה `toString()` מחזירה מחרוזת עם כל פרטי החשבון (מספר החשבון, שם הלקוח, יתרה ועמלת הפעולה).

**קובץ ההגשה:**

`CheckingAccount.java` הקוד של המחלקה.

## 2. בדיקות יחידה עבור המחלקה CheckingAccount (25 נק')

כתבו תכנית `CheckingAccountTester` שמבצעת את הפעולות הבאות.

- א. מקבלת מהמשתמש את הפרטים של החשבון הראשון: מספר החשבון, שם הלקוח, יתרת הפתיחה ועמלת הפעולה.  
**הדרכה:** כדי לקבל נתון מסוג מחרוזת, ניתן להשתמש במתודה `next()` של `Scanner`.
- ב. יוצרת חשבון חדש לפי הפרטים הנ"ל.
- ג. מדפיסה הודעה על יצירת החשבון הראשון ואת פרטי החשבון.
- ד. מקבלת מהמשתמש את הפרטים של החשבון השני: מספר החשבון ושם הלקוח בלבד.
- ה. יוצרת חשבון שני עם הפרטים הנ"ל. עמלת הפעולה ויתרת הפתיחה נקבעים לפי ברירת המחדל.
- ו. מדפיסה הודעה על יצירת החשבון השני ואת פרטי החשבון.
- ז. מקבלת מהמשתמש את הסכום שיש להעביר מהחשבון הראשון לחשבון השני ומעבירה את הסכום בין החשבונות.
- ח. מדפיסה הודעה על העברת הסכום ואת הפרטים של שני החשבונות.

דוגמה לפלט של התכנית:

```
Creating first account.
Please, enter the account number: 1023/a1
Please, enter the customer name: Alice
Please, enter the initial balance: 1000
```

```
Please, enter the transaction fee: 5.5
First account is created. Account details:
CheckingAccount [accountNum=1023/al, customerName=Alice, balance=1000.0, transactionFee=5.5]

Creating secont account.
Please, enter the account number: 909/bo
Please, enter the customer name: Bob

Second account is created. Account details:
CheckingAccount [accountNum=909/bo, customerName=Bob, balance=0.0, transactionFee=1.75]

Please enter the amount to transfer: 1000

Amount of 1000.00 is transfered from the first to the second account.
Accounts details:
CheckingAccount [accountNum=1023/al, customerName=Alice, balance=-5.5, transactionFee=5.5]
CheckingAccount [accountNum=909/bo, customerName=Bob, balance=998.25, transactionFee=1.75]
```

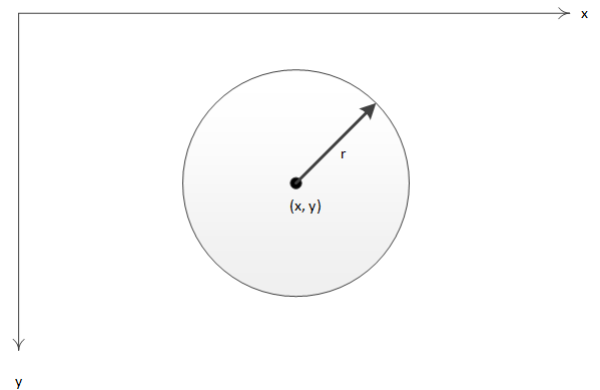
### קובצי ההגשה:

CheckingAccountTester.java הקוד של המחלקה.

TestResults.txt קובץ טקסט עם פלט של שתי הרצות שונות של התכנית. ניתן לעשות העתק\הדבק מחלון ה-Console של Eclipse לקובץ טקסט.

### 3. מחלקה Circle (20 נק')

כתבו באופן מלא מחלקה Circle שמתארת עגול. עגול מוגדר על ידי הקואורדינטות של המרכז, והרדיוס  $r$ , כפי שמופיע בציור להלן:



על המחלקה Circle לכלול את כל המשתנים הנדרשים ואת המתודות הבאות:

א. בנאי עם פרמטרים שמקבל את הקואורדינטות של המרכז והרדיוס של העגול.

ב. מתודה circumference שמחזירה את ההיקף של העגול.  
הדרכה: את היקף העגול ניתן לחשב לפי הנוסחה  $C = 2\pi R$ .

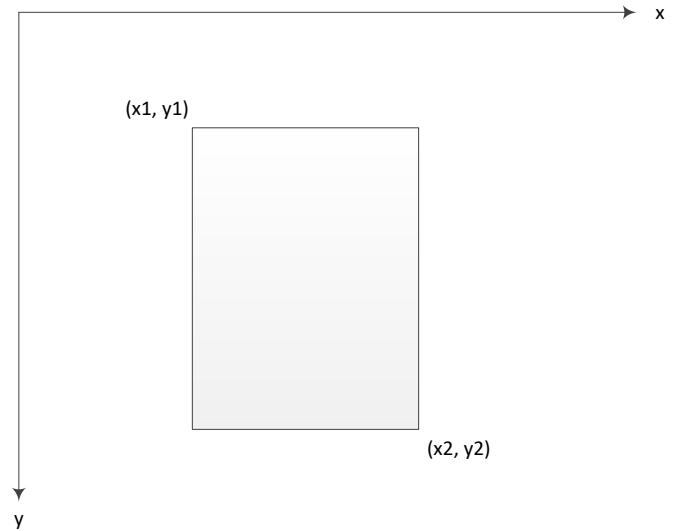
ג. (5 נק') מתודה area שמחזירה את השטח של העגול.  
הדרכה: שטח העגול ניתן לחשב לפי הנוסחה  $A = \pi R^2$ .

**קבצי ההגשה:**

Circle.java – הקוד של המחלקה.

**4. מחלקה Rectangle (20 נק')**

כתבו באופן מלא מחלקה `Rectangle` שמתארת מלבן. המלבן מוגדר על ידי הקואורדינטות  $(x1, y1)$  של הפינה השמאלית העליונה והקואורדינטות  $(x2, y2)$  של הפינה הימנית התחתונה, כפי שמופיע בציור להלן:

על המחלקה `Rectangle` לכלול את כל המשתנים הנדרשים.על המחלקה `Rectangle` לכלול ואת המתודות הבאות:

- נאי עם פרמטרים שמקבל את הקואורדינטות של שתי הפינות של המלבן.
- מתודה `area` שמחזירה את השטח של המלבן.
- מתודה `perimeter` שמחזירה את ההיקף של המלבן.
- מתודה `diagonal` שמחזירה את אורך האלכסון של המלבן.
- מתודה `toString()` שמחזירה את כל הפרטים של המלבן, כולל השטח, ההיקף ואורך האלכסון.

**קבצי ההגשה:**

Rectangle.java - הקוד של המחלקה.

**5. בדיקות יחידה של המחלקות Circle ו-Rectangle (10 נק')**כתבו תכנית `ShapesTester` שיוצרת שני עיגולים ומלבן אחד, ומדפיסה את כל הפרטים שלהם.

ניתן לקבוע את הערכים בתוך הקוד. לא נדרש לקבל קלט מהמשתמש.

לא נדרש פלט מעוצב.

דוגמה לפלט של התכנית:

```
Circle [x=0.0, y=0.0, r=10.0, circumference=62.83185307179586, area=314.1592653589793]
Circle [x=10.0, y=10.0, r=1.0, circumference=6.283185307179586, area=3.141592653589793]
```

Rectangle [x1=0.0, y1=0.0, x2=3.0, y2=4.0, area()=12.0, perimeter()=14.0, diagonal()=5.0]

### קבצי ההגשה:

ShapesTester.java – הקוד של התכנית.

ShapesTesterOutput.txt – פלט של התכנית.

**בהצלחה רבה !**