השיטה toString

כשאנו מדפיסים משתנה מטיפוס פשוט כלשהו, מה שמודפס הוא ערכו של המשתנה.

לדוגמא, אם i הוא משתנה מטיפוס int וערכו הוא 5, `int i = 5;`

כאשר נכתוב את הפקודה `System.out.println (i);`

יודפס הערך 5.

מה יקרה אם נרצה להדפיס אובייקט? מה יודפס?

בדרך כלל, כשאנו רוצים להדפיס אובייקט, אנו רוצים להדפיס תיאור שלו . הדרך הטובה ביותר להדפיס תיאור של אובייקט היא להדפיס את תיאורי משתני המופע (instance variable) שלו ואת ערכיהם.

כך למשל, אם המחלקה Person מכילה שני משתני מופע (שדות, תכונות) שהם שמו וגילו של האדם שמיוצג על-ידי האובייקט, כשנדפיס את האובייקט, נרצה שיופיעו בהדפסה השם והגיל של האדם. להלן הגדרת המחלקה:

```
public class Person
{
      String _name;
      int _age;
      // הבנאים והשיטות הושמטו
}
```

כשנרצה להדפיס אובייקט מהמחלקה Person נעשה זאת כך:

```
System.out.println ("Name = " + _name + "\t age = " + _age);
```

(שימו לב שצמד התווים \t מסמן מעבר של tabulator כמו שצמד התווים \n מסמן מעבר לשורה חדשה).

ב- Java קיימת שיטה בשם toString שזו חתימתה: `public String toString()`

כפי שאפשר להבין מהחתימה, השיטה היא שיטה ציבורית, ללא פרמטרים והיא מחזירה אובייקט מטיפוס String, כלומר מחרוזת תווים. המטרה של השיטה toString היא ליצור ייצוג של אובייקט כמחרוזת תווים (string). הדבר שימושי מאד כאשר ברצוננו להדפיס על הפלט אובייקט כלשהו.

נכון שאנו יכולים גם לכתוב שיטה בשם printPerson() שתדפיס את התיאור של אדם בעזרת הפקודה System.out.println כמו שעשינו לעיל, אבל קריאה לשיטה זו תגביל אותנו. לדוגמא, השיטה שלנו מבצעת println (מעבר שורה בסוף ההדפסה). מה יקרה אם נרצה להדפיס פעמיים בשורה אחת את פרטי האדם? נצטרך שיטה אחרת עם print. זה בהחלט לא יעיל וממש נוגד את חוקי השימוש החוזר בשיטה ואת חוקי התכנות מונחה העצמים ליצור שיטות שונות להדפסה.

יותר מכך, השיטה ()printPerson מגבילה אותנו שההדפסה תיעשה דווקא על המסך. ומה אם נרצה להדפיס לתוך קובץ? הפתרון הוא שיטת ה- ()toString.

מסיבות שנלמד מאוחר יותר ביחידות 11-12, השיטה מוגדרת בכל מחלקה, אבל כדי להשתמש בה אנו בכל זאת צריכים להגדיר אותה מחדש. בהגדרה שלנו אנו נחליט איך אנחנו רוצים לייצג את האובייקט של המחלקה, מה בדיוק לכתוב ואלו שדות להציג. אנו ניצור את המחרוזת שתייצג לדעתנו בצורה הטובה ביותר את האובייקט, ונחזיר אותה בעזרת הפקודה return.

כך למשל תיראה השיטה ()toString של המחלקה Person שתיארנו לעיל:

```
public String toString()
{
      String s = new String();
      s = "Name = " + _name + "\t age = " + _age;
      return s;
}
```

אפשר גם לכתוב את השיטה כך:

```
public String toString()
{
      return "Name = " + _name + "\t age = " + _age;
}
```

עכשיו, הפעלת השיטה על האובייקט מחזירה לנו ייצוג בעזרת מחרוזת של האובייקט. אם אנו רוצים להדפיס את האובייקט (נניח ששמו הוא p), מה שנותר לנו לעשות הוא לכתוב כך:

```
System.out.println (p.toString());
```

אבל, מתברר שגם זה לא נחוץ . מספיק שנכתוב את הפקודה `System.out.println (p);` ותודפס המחרוזת המייצגת את האובייקט. זה קורה כי הפקודה של ההדפסה מקבלת כפרמטר אובייקט מטיפוס String. כאשר הקומפיילר (מהדר) מחכה לאובייקט מטיפוס String ואין לו כזה, הוא אוטומטית קורא לשיטה ()toString המוגדרת במחלקה.

אבל שימו לב, אם **לא** מוגדרת השיטה toString במחלקה שלכם, ותכתבו את פקודת ההדפסה באחת משתי הצורות שלעיל, לא יודפס האובייקט בצורה מחרוזתית, אלא יודפס ערך אחר (שלמעשה מציין את הכתובת של האובייקט בזיכרון). שוב, הסיבות לכך יילמדו בהמשך הקורס.