

תרגיל מס' 2

בתרגיל זה נתרגל שימוש במשתנים ובתנאים, ואפילו נלמד להתבייט על מטרה.

כמו בתרגיל הקודם, את כל הפקודות אנחנו נשים בתוך המתודה `run()`:

```
public void run() {  
    // הקוד שלכם  
}
```

עבור כל משימה, חשוב להריץ את הרובוט, ולוודא שאכן קיבלנו את התוצאה הרצויה.

משימה מס' 1

כתבו רובוט שנוסע במסלול ריבועי, עשר פעמים. שימו לב – אל תכתבו את פעולת הנסיעה בריבוע יותר מפעם אחת. השתמשו במשתנים, לולאות ותנאים, כפי שנלמד בשיעור.

משימה מס' 2

כתבו רובוט שנוסע במסלול ריבועי, במשך 8 שניות. השתמשו בפעולה `getTime()` שמחזירה את מספר התורות שעברו מתחילת הסבב. שימו לב שתורות אינם מתרגמים ישירות לשניות – יש יותר מתור אחד בכל שניה. תזמנו את נסיעת הרובוט בעזרת ניסוי וטעיה.

משימה מס' 3

במשימה זו, צריך להשתמש ברובוט מיוחד כ"רובוט אב" של הרובוט שתכתבו. בהמשך הקורס נלמד עוד על ירושה ו"רובוט אב". בינתיים, כדי לעבוד עם משימה זו, בצעו תחילה את השלבים הבאים:

1. העתיקו לתיקייה `C:\robocode\robots\def` את הקובץ `ScannerRobot.class` שצורף לתרגיל.

2. ברובוט שאתם יוצרים, שנו את השורה הבאה (מדובר בשורה מס' 10):

```
public class <שם הרובוט> extends Robot
```

לשורה הבאה:

```
public class <שם הרובוט> extends ScannerRobot
```

(הגדרה זו גורמת ל-`ScannerRobot` להיות ה"אבא" של הרובוט שתכתבו)

3. ברובוט שאתם יוצרים, הוסיפו אחרי השורה (מדובר בשורה מס' 2):

```
import robocode.*;
```

את השורה:

```
import import def.ScannerRobot;
```

כעת תוכלו לכתוב את הקוד של הרובוט שלכם כפי שעשינו עד כה, בתוך `run()`.

החידוש ברובוט האב `ScannerRobot`, הוא שתוכלו להשתמש במשתנה `robotSeen`, שיהיה `true` אם אתם מכוונים על רובוט אחר, ו-`false` אם אתם לא מכוונים.

המשימה: כתבו רובוט שנכנס לזירה יחד עם `SittingDuck`, מוצא אותו, ויורה בו עד ש-`SittingDuck` מת (פשוט תמשיכו לירות לנצח, כשהוא ימות יתחיל סבב חדש).