תרגיל מס' 3

בתרגיל זה נתרגל שימוש במשתנים, בתנאים ובשילובים מורכבים שלהם.

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.
שימו לב לשמור את התרגילים שלכם, ולא למחוק בסוף כל משימה!

Gun cooling rate:

שמתם כבר לב, שאם אתם יורים כמה פעמים ברצף, חלק מהפעמים מתפספסות. זה נובע מזה שהמשחק מדמה מצב בו הנשק צריך זמן כדי להתקרר.

משימה מס' 1

כתבו רובוט שיורה שני כדורים ברצף.

שימו לב שאם פשוט תכתבו את הפקודה fire פעמיים, היריה השנייה לא תקרה בגלל gun cooling rate.

איך תעשו זאת?

יצרנו משתנה חדש מסוג double בשם currentGunHeat. כדי להשתמש בו, תצטרכו להוריד מחדש את ScannerRobot (הגרסה המעודכנת שמופיעה במייל ביחד עם התרגיל הזה), ולעשות את השינויים שהוסברו בתרגיל הקודם כדי להשתמש ב robotSeen.
הרובה שלכם יירה רק אם החום הנוכחי של הרובה הוא 0. האפשרות הטבעית כדי לחכות, היא להשתמש בלולאה.

משימה מס' 2

במשימה זו, צריך להשתמש ברובוט מיוחד כ"רובוט אב" של הרובוט שתכתבו. בהמשך הקורס נלמד עוד על ירושה ו"רובוטי 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 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.

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