

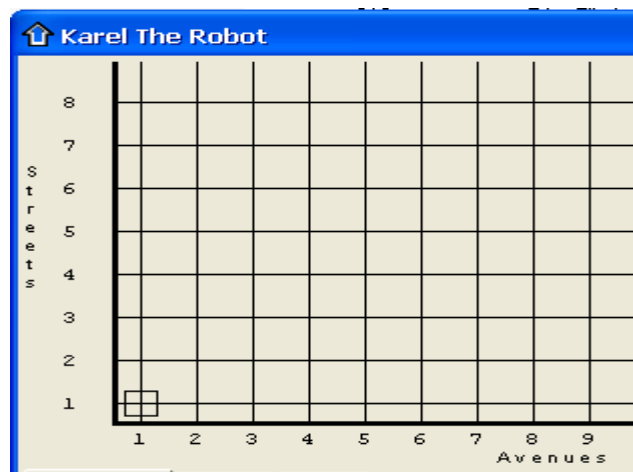
קארל הרובוט

רובוט

המלה "רובוט" הומצאה בשנת 1920 על-ידי הסופר והמחזאי הצ'כי **קארל צ'אפק** במחזהו R.U.R (הרובוט האוניברסלי של רוסום). למעשה, "רובוט" הוא מילה בשפה הצ'כית אשר פירושה "עבודה" (בדומה למילה בשפה הרוסית - "רבוטה").

קרל הרובוט

סביבת קארל הרובוט משלבת עולם ויזואלי עם שפת תכנות. קארל הוא רובוט הנע על מערכת צירים (כמו נהג מונית הנוסע בין צמתים של כבישים). תוך כדי נסיעתו הוא יכול לזהות מכשולים (קירות), להניח או לאסוף זמזמים (beepers) הנמצאים בצמתים.



עולמו של קארל מורכב מ**רשת** מישורית של רחובות. העולם ממוסגר בארבעה קירות קשיחים המונעים מקארל ליפול לתהום שמעבר לעולמו. בעולמו של קארל מצטלבים רחובות אופקיים ושדרות אנכיות. בכל מפגש בין רחוב לשדרה ממוקם **ס צומת**. השדרות והרחובות ממוספרים, כל צומת מזוהה על-ידי מספר הרחוב והשדרה המצטלבים.

קארל יכול לנוע בעולמו מצומת לצומת, רק **צפונה**, **דרומה**, **מזרחה** ו**מערבה** (לא באלכסון). פניו פונות לאחד מכיווני המצפן.

מלבד קארל יש בעולם גם **קטעי קירות וזמזמים**:

- **קטעי קירות** קשיחים באורכים שונים. ממוקמים בין צמתים עוקבים וחוסמים את מסלולו של הרובוט מצומת אחד לשני.
- **זמזמים** (beepers) הנמצאים בצמתים מסוימים ומספקים צפצוף חלש. ניתן לשמוע את הזמזם רק אם הרובוט באותה צומת. הזמזם אינו מפריע לרובוט ללכת. הרובוט יכול לאסוף את הזמזם, להניח זמזם חדש, או להתעלם ממנו.

לרובוט יש מספר **חושים**: חוש כיוון (יודע לאיזה כיוון הוא פונה), חוש שמיעה (יכול לשמוע זמזם אם הם שניהם באותו צומת) וחוש ראייה (יכול לראות קירות הנמצאים לפניו או בשני צדדיו). לרובוט יש **שק** על גבו שיכול להכיל זמזמים. יתכן שהוא ריק ויתכן שיש בו זמזמים. הוא יכול להרים זמזם ולהניח אותו בשק, או להוציא זמזם מהשק ולהניח אותו בצומת.

יצירת רובוט:

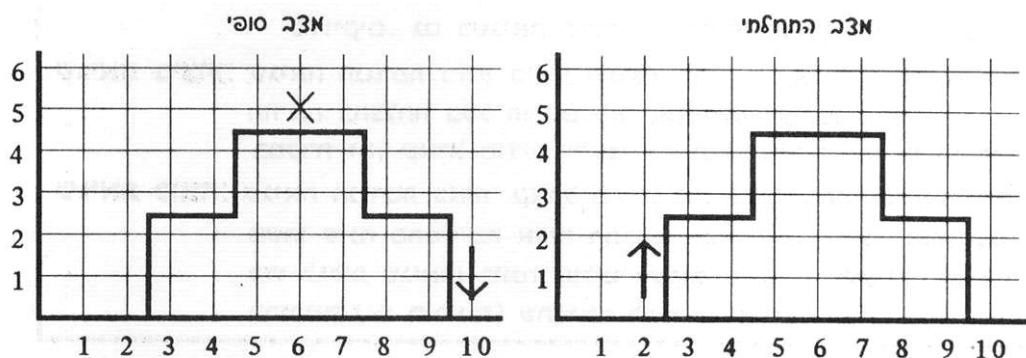
```
BasicRobot rob = new BasicRobot (3,2,North,5);
```

יצרנו עצם רובוט בשם rob אשר מיקומו על הלוח ציר : $x=3, y=2$ כיוון צפון ויש לו בשק 5 זמזמים.

הפעולות האפשריות לעצם זה (בהמשך נוסיף עוד פעולות):

- void move() – התקדם צעד אחד קדימה
- void turnLeft() – הסתובב 90 מעלות שמאלה
- void pickBeeper() – אוסף זמזם ושם בתיקו
- void putBeeper() – מוציא זמזם מהתיק ומניח בפינה בה הוא נמצא
- void turnOff() – מכבה את עצמו

משימה ראשונה:



המטרה: קארל צריך לטפס על הר להניח דגל (מיוצג על-ידי זמזם) בראש ההר ולרדת למטה. כתבו תכנית לביצוע המשימה.

```
void main() { // הצעת פתרון
    BasicRobot rob = new BasicRobot (2 ,2 ,North ,1);
    rob.move();
    rob.move();
    rob.move();
    rob.turnLeft();
    rob.turnLeft();
    rob.turnLeft();
    rob.move();
    rob.move();
    rob.turnLeft();
    rob.move();
    rob.move();
    rob.turnLeft();
    rob.turnLeft();
    rob.turnLeft();
    rob.move();
    rob.move();
    rob.putBeeper();
    rob.move();
    rob.move();
    rob.turnLeft();
    rob.turnLeft();
    rob.turnLeft(); // מצביע דרומה
    rob.move();
    rob.move();
    rob.turnLeft();
    rob.move();
    rob.move();
    rob.turnLeft();
    rob.turnLeft();
    rob.turnLeft();
    rob.move();
    rob.move();
    rob.turnOff();
}
```

אנו רואים כי ישנם פעולות החוזרות על עצמן ולכן כדאי לחלק את המשימה הראשית לתת משימות/מתודות תוך שימוש בפעולות הקיימות בקארל הרובוט ובמתודות שכבר יצרנו. כאן אנו נעזרים בפעולת ההורשה עליה נרחיב בהמשך הקורס.

```
class AdditionClass extends BasicRobot {
    void turnRight ( ) {
        turnLeft( );
        turnLeft( );
        turnLeft( );
    }
    void moveTwice( ) {
        move();
        move();
    }
    void moveThrice ( ) { // לשם הדוגמא בלבד
        moveTwice( );
        move();
    }
}
```

לאחר יצירת מחלקה זו ניתן יהיה להגדיר את התכנית בצורה הבאה:

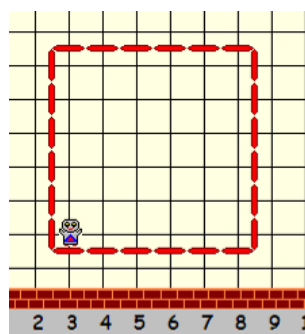
```
main {
    AdditionClass rob = new AdditionClass (2 ,2 ,North ,1);
    rob. moveThrice( );
    rob.ternRight();
    rob.moveTwice();
    rob.ternLeft();
    rob. moveTwice ();
    rob. ternRight ();
    rob. moveTwice ();
    rob. putBeeper();
    rob. moveTwice ();
    rob. ternRight (); // מצביע דרומה
    rob. moveTwice ();
    rob.ternLeft();
    rob. moveTwice ();
    rob. ternRight ();
    rob. moveTwice ();
    rob.turnOff();
}
```

בנוסף לפעולות הבסיס לעצם קארל הרובוט יש מתודות נוספות :

boolean anyBeepersInBag()	מחזיר "אמת" אם יש זמזם בתיק
boolean anyBeepersInCorner()	מחזיר "אמת" אם יש זמזם בפינה
boolean frontIsBlocked()	מחזיר "אמת" אם יש קיר קדימה
boolean frontIsClear()	מחזיר "אמת" אם אין קיר קדימה
boolean leftIsClear()	מחזיר "אמת" אם אין קיר משמאל
boolean rightIsClear()	מחזיר "אמת" אם אין קיר מימין
int getBeeperNumber()	מחזיר את מספר הזמזמים בתיק
int getCornerBeeperNumber()	מחזיר את מספר הזמזמים בפינה
int getOrientation()	מחזיר את הכיוון אליו פונה הרובוט (הכיוון מוחזר כמספר בשונה ממה שהוגדר בהרצאה

לפניך קטע תכנית :

```
AdditionClass robi = new AdditionClass (2 , 2,North ,4);
while ( robi.anyBeepersInBag() )
{
    while ( robi.frontIsClear() )
        robi.move();
    rubi.putBeeper();
    robi.moveThrice();
}
```



הסבר בקצרה מה מבצע קטע התכנית עבור הלוח הבא :

לפניך פתרון נוסף לאותה מטרה . הקטע המכיל מספר טעויות (לוגיות/כוונה וקומפילציה). הקף כל טעות והסבר את משמעותה.

```
Additionclass robi = new Additionclass (3,3,North,2);
while ( robi.anyBeepersInBag() == true )
{
    while ( robi.frontIsBlocked() == false ) // או while ( ! robi.frontIsBlocked() )
        robi.move();
    robi.putBeeper();
    robi.moveTriple();
}
```