

חומרים שהוכנו על-ידי מורי הניסוי תש"ע להוראת "יסודות מדעי המחשב"

ניתן להשתמש בחומרים לצורך הוראה בלבד.
לא ניתן לפרסם את החומרים או לעשות בהם כל שימוש מסחרי
ללא קבלת אישור מראש מצוות הפיתוח

סביבת קארל, פרק 1 הכרות מותאם להוראת הפרק הראשון (הכרות עם עצמים)

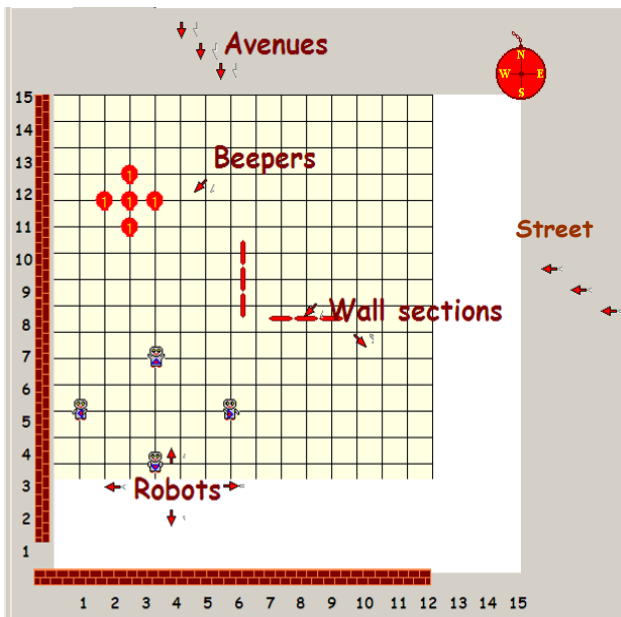
כתיבה ועריכה:

שירלי רוזנברג-כהן

מעובד על פי הספר "מבוא למדעי המחשב – סביבת קארל הרובוט"
שפותח ע"י צוות מגוון במחלקה להוראת הטכנולוגיה והמדעים, הטכניון

דפי העבודה מבוססים על התוכנה Object Karel for Karel++ מהדורת 2005
ניתן להוריד את התוכנה ללא תשלום בכתובת
<http://pcc.pace.edu/~bergin/temp/findkarel.html>

עולם הרובוטים

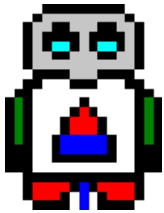


עולם הרובוטים מורכב מרשת מישורית של רחובות אופקיים ושדרות אנכיות. העולם ממוסגר בארבעה קירות קשיחים המונעים מהרובוטים ליפול לתהום שמעבר לעולם. בכל מפגש בין רחוב לשדרה ממוקמת צומת. השדרות והרחובות ממוספרים, כל שכל צומת מזוהה על-ידי מספר הרחוב והשדרה המצטלבים.

בעולם יש גם קטעי קירות וזמזמים.

קטעי קירות קשיחים באורכים שונים. ממוקמים בין צמתים עוקבים וחוסמים את דרכם של הרובוטים מצומת אחד לשני. ניתן לבנות בעזרת הקירות חדרים, מבוכים וכד'.

זמזמים (beepers) הנמצאים בצמתים מסוימים ומספקים צפצוף חלש. הרובוטים יכולים לשמוע זמזם רק אם הם עומדים בדיוק באותה צומת. הזמזם אינו מפריע לרובוט ללכת. רובוט יכול לאסוף את הזמזם, להניח זמזם חדש, או להתעלם ממנו.



הרובוט:

בעולם הרובוטים יש בית חרושת המייצר רובוטים. כרגע מייצר בית החרושת רובוט מסוג אחד בלבד. כמובן – ניתן לייצר כמה רובוטים מאותו סוג.

לכל רובוט המיוצר בבית החרושת יש תכונות ופעולות.

תכונות הרובוט הם: שמו, טיפוס-האב, המקום בו יעמוד בתחילת דרכו(מספר הרחוב ומספר השדרה), הכיוון שאליו יפנה (צפון, דרום, מערב או מזרח) ומספר הזמזמים ההתחלתי שהרובוט נושא עימו בשק על גבו.

כאשר יוצרים מופע של רובוט נותנים ערכים ראשוניים לתכונות אלה. התכונות תשתננה במהלך ביצוע המשימה.

למשל: Primitive_Robot Karel (1, 2, East, 0);

טיפוס	שם (מספר זמזמים בשק, כיוון, שדרה, רחוב)
-------	---

או: Robot Bob (4, 7, north, 10);

הרובוט מסוגל לבצע משימות שונות בעולמו, באמצעות מספר פעולות אותן הוא יודע לבצע. לשם כך עליו לקבל סדרת הוראות: תכנית.

הפעולות הבסיסיות:

- **הוראת תזוזה – move**
בעת ביצוע ההוראה הרובוט זז קדימה לצומת הקרוב, בלי לשנות את כיוון פניו. כדי למנוע נזק לעצמו – הרובוט לא ינוע קדימה אם הוא רואה מחסום לפניו. במקרה זה, הרובוט יודיע על שגיאה ויפסיק את ביצוע התכנית.
- **הוראת סיבוב שמאלה – turnLeft**
בעת ביצוע ההוראה קארל פונה ב-90 מעלות שמאלה בלי לזוז ממקומו. מאחר ואין קירות שיפריעו בביצוע הפעולה, אין אפשרות שתיווצר שגיאה בזמן הביצוע.
- **הוראה להרמת זמזם – pickBeeper**
בעת ביצוע ההוראה הרובוט מרים זמזם מהצומת בו הוא נמצא ומאכסן אותו בשק. במקרה שיש מספר זמזמים בצומת - הרובוט ירים רק אחד. במקרה שאין זמזם בצומת – הרובוט יודיע על שגיאה ויפסיק את ביצוע התכנית.
- **הוראה להנחת זמזם – putBeeper**
בעת ביצוע ההוראה הרובוט לוקח זמזם אחד מהשק ומניח אותו בצומת בו הוא עומד. במידה והשק ריק בעת ביצוע ההוראה – הרובוט יודיע על שגיאה ויפסיק את ביצוע התכנית.

לרובוט יש גם חושים:

- **חוש כיוון** – הרובוט יודע לאן הוא פונה.
- **חוש שמיעה** – הרובוט יכול לשמוע זמזם כאשר הוא נמצא באותה צומת.
- **חוש ראייה** – הרובוט יכול לראות קירות הנמצאים לפניו או בשני צדדיו.

נעסוק בפעולות המנצלות חושים אלה רק בשלב מאוחר יותר.

חוקי תחביר לתכנית: ¹

הרובוט מבין ומבצע רק הוראות שנכתבו בדיוק כפי שהוא מכיר.

1. כל משימה מתחילה במילה task (משימה) ולאחריה בלוק של הוראות התחום בסוגריים מסולסלים.
2. בתחילת כל משימה יש להורות על ייצור רובוט אחד לפחות.

למשל: Robot Karel (1, 2, East,10);

3. מכיון שיכולים להיות כמה רובוטים בו זמנית בעולם, יש לציין לפני כל הוראה מיהו הרובוט שצריך לבצע אותה ואח"כ תבוא נקודה וההוראה עצמה. כל הוראה מסתיימת בצמד סוגריים, ולאחריהן נקודה-פסיק.
למשל: karel.move();

```
task
{
    Robot Karel ( 3, 2, east, 2);
    Karel.turnLeft();
    Karel.move();
    karel.putBeeper();
    Karel.turnOff();
}
```

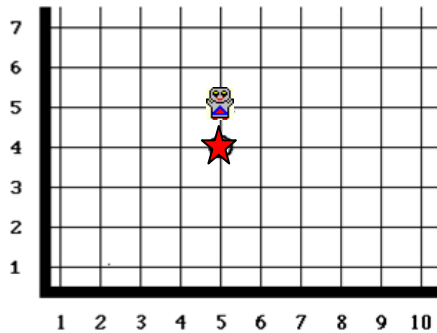
דוגמה לתכנית:

¹ התוכנית כאן לפי תוכנה מהדורת 2005 הקרויה: Object Karel for Karel++
ניתן להוריד אותה בכתובת: <http://pcc.pace.edu/~bergin/temp/findkarel.html>

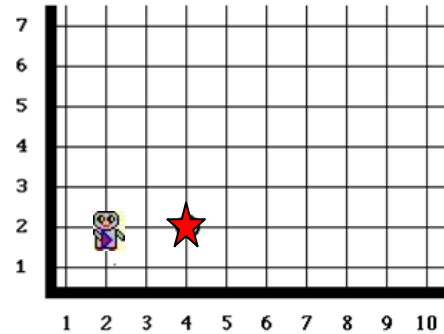
משימה לדוגמא:

ניצור רובוט בשם Rob. במצב ההתחלתי Rob עומד בצומת 2,2 כשפניו מזרחה. המטרה היא לגרום לRob לאסוף את הזמזם מצומת 2,4 ולהניחו בצומת 4,5 ולעצור בצומת 5,5 כאשר פניו צפונה.

מצב סופי



מצב התחלתי



```
task
{
    Robot Rob(2,2,east,0);
    Rob.move();
    Rob.move();
    Rob.pickBeeper();
    Rob.move();
    Rob.turnLeft();
    Rob.move();
    Rob.move();
    Rob.putBeeper();
    Rob.move();
    Rob.turnOff();
}
```

התכנית המתאימה תהיה:

שלי הביצוע:

בשלב הראשון Rob קורא את התכנית ובודק אם הוא מכיר את המילים. אם משהו לא מוכר לו (למשל חסר הסימן נקודה-פסיק בסוף הוראה) הוא לא מוכן לבצע. שלב זה של בדיקה נקרא **הידור** או **קומפילציה** (compile). רק אם הכל עבר בשלום קארל מתחיל ב**ביצוע**. הביצוע יעשה לפי הסדר כל זמן שלא ניתנה הוראת סיום או נגרמה שגיאה.

סוגי שגיאות:

שגיאה פירושה שהמחשב (או הרובוט) אינו מנחש את כוונתנו. בכל שפת תכנות קיימים שלושה סוגי שגיאות:

- שגיאות הקשורות בכללי ה**תחביר** של השפה, כתיבת הוראה שאינה נמצאת באוצר המילים של המחשב, או סימני פיסוק לא מדויקים. שגיאות אלה גורמות לאי הרצת התכנית.
- שגיאות המתגלות בזמן ה**ביצוע** של התכנית וגורמות לעצירת התכנית (למשל חילוק ב-0 במחשב, או הוראת פניה לקארל למקום שחסום על ידי קיר).
- שגיאות **לוגיות** – שגיאות כוונה. שגיאות של המתכנת שגרמו למחשב לעבוד לפי התכנית שקיבל אך לא זאת היתה המטלה או כוונת המתכנת.

במדעי המחשב כל סוגי השגיאות נקראות bugs ושלב חיפוש השגיאות ותיקונן נקרא **ניפוי שגיאות** (debugging).

תרגיל 1:

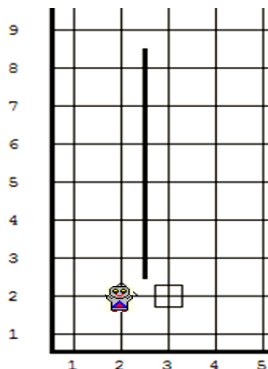
- לפניך סדרות של הוראות לביצוע. תרגם כל סדרה כזו לתכנית שלמה בשפת הרובוטים.
 עבור כל אחת מהתוכניות אותחלו תכונותיו של הרובוט קארל כך: Robot Karel(2,2,east,2)
 עבור כל אחת מהתוכניות: מה יהיה מצבו (ערכי תכונותיו) של הרובוט לאחר ביצוע המטלה?
 א. התקדם 4 פעמים, הסתובב שמאלה, התקדם והנח זמזם.
 ב. הרם זמזם, התקדם, הנח זמזם, התקדם, הסתובב שמאלה, והתקדם.
 ג. התקדם, התקדם, הסתובב ימינה, הנח זמזם, התקדם, הנח זמזם, התקדם.
 ד. עליך להגיע לצומת 5,4.

הוראת חזרה:

לפעמים, בעת ביצוע התכנית, יש לחזור על הוראה מספר קבוע של פעמים. והתכנית הופכת להיות ארוכה ומסורבלת. הדרך לומר לרובוט לחזור על פעולה כמה פעמים (לולאה) היא באמצעות הפעולה: loop ().
 ובתוך סוגריים מסולסלים ההוראות שעליהם נרצה לחזור.

```
loop (3)
{
    karel.turnLeft();
}
```

למשל: הפעולה "הסתובב ימינה" בעזרת הפעולה .turnleft.



דוגמא: משימתו של קארל היא "לטפס" על העמוד ולהדליק שני "פנסים" (זמזמים) בראש המשוכה ולחזור למטה מצידה השני.

```
ההוראות תהיינה :
התקדם 7 פעמים.
הנח זמזם.
הסתובב ימינה.
התקדם.
הנח זמזם.
הסתובב ימינה.
התקדם 7 פעמים.
```

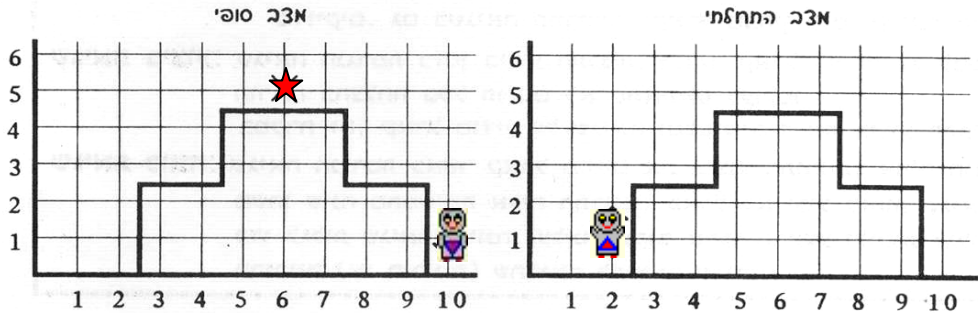
```
התכנית:
task
{
    Robot Karel(2,2,north,2);
    loop (7)
    {
        Karel.move();
    }
    Karel.putBeeper();
    loop (3)
    {
        Karel.turnLeft();
    }
    Karel.putBeeper();
    loop(7)
    {
        Karel.move();
    }
}
```

תרגיל 2:

השלם את הקטעים החסרים בתכנית – המסומנים בקו. הרץ את התכנית.

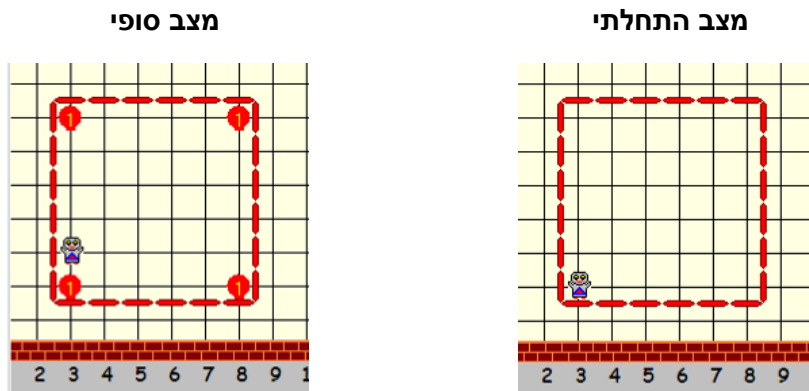
תרגיל 3:

כתוב תכנית שבה הרובוט צריך לטפס על הר להניח דגל (מיוצג על-ידי זמזם) בראש ההר ולרדת למטה.
שים לב: כתוב תכנית שבה יהיה צורך לשנות רק שתי שורות אם יהיו 20 מדרגות ולא רק שתיים.

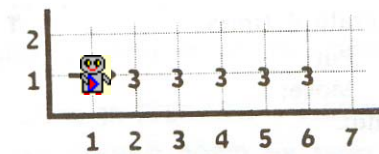


תרגיל 4:

קארל התבקש לסייע בקישוט החדר לקראת מסיבה. משימתו לתלות בלון (זמזם) בכל אחת מפינות החדר (המתואר בתרשים). כתוב תכנית לביצוע המשימה.

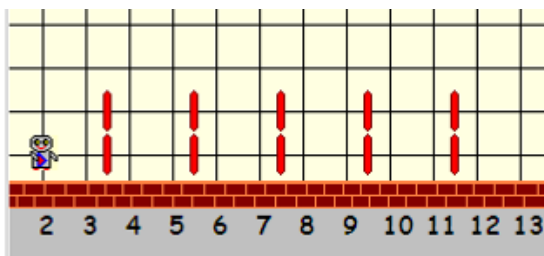


תרגיל 5:



בדרך לביה"ס קארל מתבקש לאסוף תלמידים מ-5 תחנות ההסעה. התחנות מסודרות ברצף ובכל תחנה ממתנים 3 תלמידים. בסוף הדרך על קארל לפרוק את כל התלמידים באותו מקום.

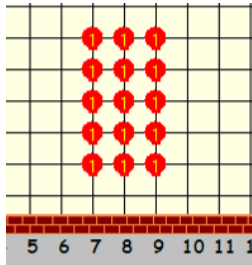
תרגיל 6:



בריצת משוכות על הרובוט לרוץ ולעבור את כל המשוכות.

עבודה עם מספר רובוטים:

לפעמים יש עבודה קשה שכדאי לחלק אותה בין מספר רובוטים. נוכל להכניס לעולם מספר **מופעים** של הרובוט, וכך יעבדו כמה רובוטים בעולם. מכיון שהפעולה מתיחסת למופע מסוים של הרובוט (שם שונה לכל רובוט) כל רובוט ידע מה לעשות. מכיון שכך – כל רובוט יכול לעשות עבודה אחרת.



דוגמא:

בשדה התותים נשכרו לעבוד 3 רובוטים (אבי, בני וגילה) על מנת לאסוף את התותים.

כל אחד מהרובוטים האלה מיוצר באותו בית החרושת שממנו הגיע קארל ולכולם יש את אותן התכונות: שם, טיפוס-האב, מיקום התחלתי, כיוון התחלתי ומספר הזמזמים ההתחלתי בשק.

שימו לב שהערכים של התכונות שונים בין הרובוטים. למשל, המיקום ההתחלתי של אבי וגילה הוא שונה אבל לשניהם יש מיקום התחלתי.

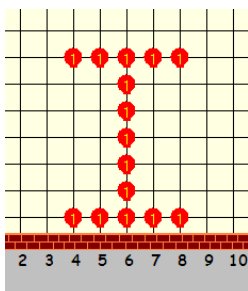
```
Robot Avi(1,7,North,0);
Robot Beni(1,8,North,0);
Robot Gila(1,9,North,0);
```

במונחים של עולם התכנות, נאמר שאבי, בני וגילה הם **מופעים** שונים של הרובוט. הואיל והם יוצרו באותו בית חרושת, הם מכירים את אותן הפעולות. רק לפני שאנחנו מביקשים ביצוע פעולה, נצטרך לציין ממי אנחנו מביקשים לבצע אותה. למשל: `Avi.move()` או `Gila.pickBeeper()`;

התכנית המתאימה יכולה להיות:

```
task
{
    Robot Avi(1,7,North,0);
    Robot Beni(1,8,North,0);
    Robot Gila(1,9,North,0);
    loop(5)
    {
        Avi.move();
        Avi.pickBeeper();
        Beni.move();
        Beni.pickBeeper();
        Gila.move();
        Gila.pickBeeper();
    }
}
```

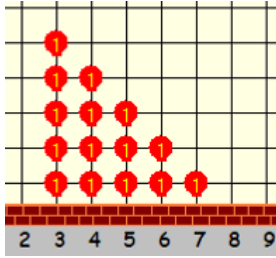
תרגיל 6:



א. שנה מינימום הוראות בתכנית הקודמת על מנת לאסוף את שדה התותים בעולם:

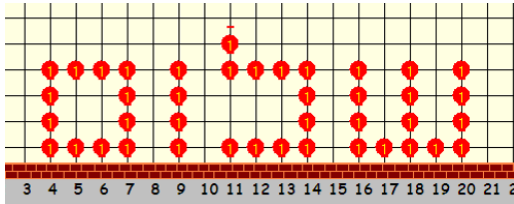
ב. מה תצטרך לשנות בתכנית החדשה כך שהרובוטים יעבדו בזה אחר זה, ולא בו זמנית.

תרגיל 7:



שדה התותים הזה נשתל בצורה מיוחדת. שים לב שהרובוטים לא חייבים לבצע כולם אותה העבודה. כלומר, אחד יכול לאסוף שורה של 5 זמזמים, שני יאסוף 4 זמזמים, והשלישי ירקוד ויסתובב במעגלים ויעשה להם שמח בזמן שהם עובדים...

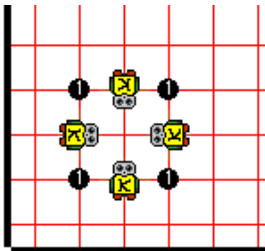
תרגיל 8:



יש לצייר את המילה "שלום" בעזרת זמזמים.

כתוב תכנית שתצייר את המילה בעזרת כמה רובוטים. לשיקולך: האם תעדיף שכל רובוט יצייר אות אחרת? או אולי שיהיה רובוט אחד "מומחה" לעמודות ואחר "מומחה" לשורות.

תרגיל 9:



כתוב תכנית היוצרת ארבעה רובוטים הממוקמים לפי התמונה.

כל רובוט יאסוף את הזמזם אשר מימינו ויפרוק אותו בנקודת האמצע, ולאחר מכן יחזור למקומו המקורי.

תרגיל 10:

כתוב תכנית המדמה מרוץ שליחים על-ידי ארבעה רצים. כל רץ מתקדם 5 צעדים ומעביר לרץ הבא אחריו את המקל (המיוצג על-ידי זמזם).

תרגול משעשע!