

לולאות ב-Scratch

מוטי בן-ארי

המחלקה להוראת המדעים

מכון ויצמן למדע

הזכויות שמורות למרדכי (מוטי) בן-ארי 2010. השימוש במסמך כפוף לרישיון "ייחוס-שימוש לא מסחרי-איסור יצירת נגזרות 2.5 ישראל" של Creative Commons. ניתן לעיין ברישיון באתר:

<http://creativecommons.org/licenses/by-nc-nd/2.5/il/>

1. מבוא

קיימים ב-Scratch ארבעה מבנים של לולאות. באופן מוזר, אין לאף אחד מהם הקבלה ישירה בשפות תכנות רגילות כגון Java. מסמך זה מסביר את הסמנטיקה של הלולאות ב-Scratch תוך השוואתם למבנים בשפת Java ובשפות אחרות. המסמך מיועד למורים בעלי ניסיון בתכנות המלמדים Scratch, וכן לתלמידים שלמדו Scratch וכעת מתחילים ללמוד שפות תכנות אחרות.

2. ביצוע חוזר אינסופי

הלולאה הפשוטה ביותר ב-Scratch היא ביצוע חוזר אינסופי, ולה המבנה הבא:



מבנה זה גורם לגוף הלולאה (סדרת ההוראות הנכללת בתוך ה-"פה" של הלבנה) להתבצע שוב ושוב. בשפת התכנות Ada קיים מבנה דומה:

```
loop ... end loop;
```

אין מבנה לביצוע חוזר אינסופי ברוב שפות התכנות, אולם, אין כלל קושי לכתוב הוראה לביצוע חוזר אינסופי. למשל, כך ניתן לעשות זאת ב-Java, אם כי בצורה מעט מסורבלת:

```
while (true) { ... }
```

:או

```
do { ... } while (true)
```

שוני נוסף בין Scratch לשפות אחרות הוא שב-Scratch לא ניתן "לעזוב" הוראה לביצוע חוזר אינסופי. ניתן לראות זאת בצורה פשוטה מהתחביר: ללבנה "לעולמים" קו תחתון חלק המונע מהלבנה הצמדת לבנים נוספות אחריה בתסריט. ב-Java קיימת הוראת break המאפשרת עזיבה של לולאה כלשהי.

3. ביצוע חוזר מוגבל מראש

הוראה של Scratch:



גורמת לביצוע של גוף הלולאה מספר פעמים קבוע. הוראה זו נוחה מאוד לשימוש כי אין צורך לתכנת את הטיפול במשתנה הבקרה בצורה מפורשת, כפי שנעשה ב-Java:

```
for (int i = 0; i < 10; i++) { ... }
```

מאידך, אם נזקקים למשתנה בקרה ב-Scratch, יש להצהיר עליו, לאתחל אותו, ולעדכן אותו בהוראות נפרדות, כפי שניתן לראות בתסריט שלהלן בו הדמות מציגה את המספרים מ-0 ל-9:



תסריט זה דומה למשפטים ב-Java:

```
for (int i = 0; i < 10; i++)  
    System.out.println(i);
```

4. ביצוע חוזר מותנה

בעת ביצוע ההוראה:



מתבצע גוף הלולאה שוב ושוב עד שהתנאי הופך ל-"אמת". מפתה להשוות הוראה זו להוראת `while` ב-Java, אבל משמעותו היא הפוכה. ההוראה:

```
while (condition) { ... }
```

מבצעת את גוף הלולאה כל עוד התנאי הוא "אמת", בעוד ההוראה ב-Scratch מבצעת את גוף הלולאה כל עוד התנאי הוא "שקר"!

הוראה דומה אחרת ב-Java היא ההוראה:

```
do { ... } while (condition)
```

גם הוראה זו מבצעת את גוף הלולאה כל עוד התנאי הוא "אמת", אבל בה מובטח שגוף הלולאה מתבצע לפחות פעם אחת.

המבנה ב-Java השקול להוראה ב-Scratch הוא:

```
while (!condition) { ... }
```

ההוראה ב-Scratch דומה להוראה ב-Pascal:

```
repeat ... until condition
```

אבל לא לחלוטין, כי בהוראה של Pascal גוף הלולאה מתבצע לפחות פעם אחת.

5. ביצוע חוזר אינסופי משולב בדיקה

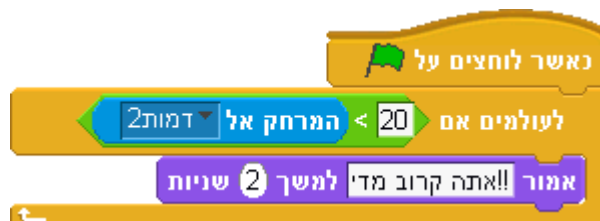
ההוראה ב-Scratch:



שונה לחלוטין ממבני לולאות בשפות אחרות. מבחינה תחבירית, ההוראה דומה להוראה לביצוע חוזר אינסופי "לעולמים", ולכן גם הוראה זו יכולה להופיע רק בסופו של תסריט. מבחינה סמנטית, הלולאה מתבצע בצורה **אינסופית**, אבל גוף הלולאה מתבצע רק כאשר התנאי הוא "אמת". ההוראה שקולה למבנה ב-Java:

```
while (true)
  if (condition) {
    ...
  }
```

ההוראה של Scratch שימושית מאוד בתכניות מקבילות כאשר תהליך אחד (דמות אחת) משנה משהו ותהליך אחר בודק אותו. בתסריט שלהלן, דמות אחת מציגה אזהרה אם "דמות2" מתקרבת אליה יותר מדי:



6. סיכום

ראינו שמבני הלולאות ב-Scratch שונים מאוד ממבני הלולאה בשפות רגילות. אין זאת אומרת שהמבנים טובים יותר או טובים פחות; הם פשוט שונים! מתכנתים מנוסים, וכן תלמידים העוברים מ-Scratch לשפות תכנות רגילות חייבים להיות ערים להבדלים ולשנות את סגנון התכנות בהתאם.