

סיכום נושאים: תנאי מורכב לולאת while, לולאת for, מספר אקראי

תנאי מורכב

ביטוי תנאי פשוט:

if (a > 20)

ביטוי תנאי מורכב: (חיבור תנאים בעזרת 'או', 'וגם')

if ((a > 20 && b==30) || (c != 30))

פירושו: אם a גדול מ 20 וגם b שווה ל- 30 או c שונה מ- 30

|| פירושו 'או'
&& פירושו 'וגם'

הסימנים הלוגיים:

a > b	a גדול מ b
a >= b	a גדול או שווה ל b
a < b	a קטן מ b
a <= b	a קטן או שווה ל b
a == b	a שווה ל b
a != b	a שונה מ b

דוגמאות: a=10 b=20 c=30

(a > 10) && (b==20) ==> שקר
(a==10) || (c != 30) ==> אמת
(c >= 20) || ((a==b) && (c==b)) ==> אמת
(a>100) || (a < 10) || (a > 8) ==> אמת

מהי לולאה בתכנות?

חזרה על אותו קטע של פקודות יותר מפעם אחת. מדוע זה נחוץ?

נניח שאמרתי לכם להדפיס את המספרים מ-1 עד 5 - לא נורא. מה אם ביקשתי להדפיס את המספרים מ-1 ועד 5 מליון - כמה פקודות נצטרך?

ישנו מבנה בשפת סי שרפ שנקרא לולאת while - כלומר 'כל עוד' משהו נכון. כל עוד איזה תנאי מתקיים, בצע את ההוראות הבאות.

מבנה הפקודה:

```
while ( condition )  
{  
    command1;  
    command2;  
    .  
    .  
}
```

שימו לב שאין נקודה פסיק לאחר הסוגריים של התנאי, התנאי תמיד בסוגריים ומיד מופיע בלוק של פקודות לאחר כותרת ה- while. אם לא תהיינה סוגריים מסולסלות, הלולאה תכלול פקודה אחת בלבד, כמו במשפט if.

דוגמה:

```
using System;  
class WhileLoop  
{  
    public static void Main()  
    {  
        int myInt =1;  
        while (myInt <= 10)  
        {  
            Console.Write("{0} ", myInt);  
            myInt++;  
        }  
        Console.WriteLine();  
    }  
}
```

לולאת for

לדוגמה:

```
for (int j=2; j <10; ++j)
{
    Console.WriteLine( j+ " " + j*j);
}
```

הכותרת (של משפט ה- for) היא בעלת 3 חלקים בתוך הסוגרים שמופרדים ביניהם על ידי התו נקודה-פסיק

1. השמת ערך למשתנה (והגדרתו - אפשר גם להגדירו קודם), בדוגמה: j (אפשר להגדיר יותר מאחד)

2. בדיקת התנאי (באמצע), בדוגמה: "כל עוד j קטן מעשר)

3. הגדלת משתנה הלולאה (j++) - זהו עידכון כלשהו

קודם מתבצעת פעולה מספר 1. לאחר מכן פעולה מספר 2. אם התנאי של פעולה 2 נכון, גוף הלולאה מתבצע מתחילתו עד סופו (בדוגמה זו רק הדפסה). אם התנאי אינו נכון, מסתיימת הלולאה.

הפעולה השלישית מתבצעת לאחר הפקודות שבתוך גוף הלולאה, בדיקת התנאי נעשית (כאמור) לפני ביצוע גוף הלולאה, ולכן יכול להיות מצב שהלולאה לא תתבצע כלל.

לאחר ביצוע העידכון בפעולה השלישית, שוב נעשית בדיקת התנאי (פעולה 2) וחוזר חלילה.

בדרך כלל לא משנים את משתנה הלולאה (j) בתוך גוף הלולאה על ידי פקודה. הוא נישלט עלידי משפט ה- for

הלולאה הנ"ל תדפיס את המספרים מ-2 ועד 9, והריבוע של כל אחד עם רווח ביניהם.

מספרים אקראיים

הזכרנו את המשחק שבו המחשב צריך 'לחשוב' על מספר בטווח מסוים (למשל מ-1 עד 1000), והתפקיד של המשתמש זה לנחש מה המספר, כאשר על כל ניחוש המחשב משיב אם הוא גבוה, נמוך או שווה למספר שבחר. המשחק מסתיים ברגע שהמשתמש ניחש נכונה.

עבור תוכנית כזו נצטרך מנגנון שמייצר מספרים אקראיים בגבול שנקבע מראש (כך שהתוכנית כל פעם תבחר מספר אקראי אחר בטווח). בנוסף נצטרך לולאה שתבצע עד אשר הניחוש יהיה נכון (אותו מספר שבחר המחשב). איננו יודעים מראש לכמה ניחושים השחקן יזדקק לכן זו תהיה לולאת `while`. בתוך הלולאה נצטרך להפעיל פקודת תנאי שתשווה בין המספר שהמחשב בחר והניחוש העכשווי ואז נודיע למשתמש איך הניחוש נמצא ביחס למספר שבחר המחשב (גדול, קטן או שווה)

כיצד התוכנית יכולה לייצר מספר אקראי נניח בין 1 ל-1000? לשם כך ישנה מחלקה שניקראת: `Random`, וכדי להפעיל אותה בתוכנית, נגדיר אובייקט "מסוג" `Random`, (או מופע של `Random`) בצורה הבאה:

```
Random r1 = new Random();
```

וכעת על מנת לייצר מספר אקראי בין 1 ל-1000, נפעיל פעולה שנימצאת במחלקה: `Random` שניקראת: `Next`, על האובייקט שיצרנו `r1`. וניתן לפעולה זו 2 פרמטרים, גבול תחתון וגבול עליון בצורה הבאה:

```
int computer_selection = r1.Next(1,1001);
```

הגבול התחתון הוא 1 והעליון 1001 כיוון שהפעולה `Next` מייצרת מספרים אקראיים שכוללים את הגבול התחתון ועד לגבול העליון פחות 1.