

חזרה כללית לחלקכם, המשך לולאות ליתר

היום חזרנו על חומר באופן כללי, לטובת מספר תלמידים, חלקם הצטרף אלינו מאוחר וחלקם עדיין לא מרגיש שולט בחומר ע"מ להמשיך קדימה.

בינתיים, בחזרה על החומר, דיברנו על מהי תוכנית C#, ועל כך שיש עדיין דברים 'סתומים' למדי שאיננו מבינים לגמרי מדוע הם שם, כגון: משפטי ה- using שמופיעים בראש כל תוכנית חדשה שאנו רוצים לכתוב בסביבת העבודה של: visual C#.

עוד דברים שלא לגמרי מובנים למה התוכנית שלנו מתחילה דווקא לאחר השורה שבה כתוב:

```
public void Main(...)
```

ומה זה בכלל ה- Main הזה? מהי המילה class שחוזרת על עצמה בתדירות מפחידה? אל חשש, לאט לאט יתבררו יותר פרטים. אם תיזכרו, כבר אמרתי בעבר שפה ושם יהיו דברים לא מובנים, כדי שנוכל לכתוב משהו מעניין, אבל הכל יתבהר בסופו של דבר.

אז התחלנו לדבר על איך מגדירים משתנים שהם 'שטח' בזיכרון שיכול להכיל ערכים שמשתנים (כמו מספרים שלמים או מחרוזות) והמשכנו לדבר על פקודות קלט ופלט (ReadLine ו- WriteLine) וגם נוכחנו שאם לא מדפיסים דבר על המסך, לא יודעים אם התוכנית מתבצעת. לכן כדי לקלוט נתון, קודם הדפסנו הודעה למשתמש מה אנו מצפים לקלוט, ואח"כ בצענו את פקודת הקלט. בצורה כזו:

```
Console.WriteLine("Hey, user, please enter an integer number: ");
```

```
int input; //prepare an integer variable for input
```

```
input=int.Parse(Console.ReadLine());
```

גם בזה יש מוזרות מסוימת, אבל אמרנו שבסיס הפקודה הוא ה- ReadLine() שגם אותה נסביר בעתיד, וגם אמרנו שה- int.Parse אומר למחשב שאנו עומדים לקרוא מספר שלם. לאחר צמד הקווים הנטויים כתבתי הערה.

אמרנו שמחרוזת זה אוסף תוים שנכתבים בין צמד של גרשיים כפולים כך: "This is a string", ואם נשים מחרוזת זו לתוך משתנה מסוג string, רק מה שיש בין הגרשיים היא המחרוזת (ללא הגרשיים), כך שאם למשל נכתוב:

```
string str1 = "This is the first string";
```

```
Console.WriteLine(str1);
```

מה שיודפס זה: This is the first string

אמרנו שיש גם אפשרות ל'שרשר' מחרוזות, כלומר לחבר אותן אחת לשניה בעזרת פעולה החיבור + (כמו חיבור במתמטיקה), למשל:

```
string st1= "The class is.....";
string st2="Introduction to Computer Science";
string st3 = st1+ " " +st2; //concatenate 3 strings
Console.WriteLine(st3);
```

מה שיודפס זה: The class is..... Introduction to Computer Science - למעשה נעשה כאן שירשור של 3 מחרוזות, שתיים מצויות בתוך המשתנים: st1 ו-st2, וביניהן מחרוזת 'מפורשת' (זה ניקרא באנגלית: literal) והסדר משמאל לימין קודם st1, אח"כ " ", ולבסוף st2. כל השירשור הזה ניכנס כערך (השמה) למשתנה st3.

ניסינו גם את פקודת ה-if-else ודיברנו על תנאי של השוואת מספרים שנעשה בעזרת == ובאופן כללי תמיד אחרי פקודת ה-if יהיו סוגריים עם התנאי ולאחר מכן יהיה 'בלוק' של פקודות, אלה הן פקודות המוקפות בסוגריים מקושטות { } - המלצתי לכל הקבוצה תמיד לכתוב בלוק כזה אחרי ה-if וגם אחרי ה-else (אם יש else), למרות שאם רוצים לכתוב רק פקודה בודדת, אפשר להשמיט את הבלוק.

אמרנו גם שה == הוא תנאי, בניגוד ל = (אחד) שמסמל השמה של מה שכתוב מימינו אל מה שכתוב משמאלו של סימן השווה. ניסינו גם תנאי עם הסימנים <= שזה אומר: האם מה שמשמאל קטן או שווה למה שמימין.

כתבנו שוב את התוכנית שקולטת מספר מעלות בצלזיוס ומסבה את המספר למעלות פרנהייט בעזרת נוסחה ידועה: $F = C * 9/5 + 32$ - זה הראה לנו איך לייצג פעולות מתמטיות ב-C# (למשל * זה כפל / זה חילוק).

בשיעור הבא נמשיך לתרגל תנאים ונעבור גם על לולאות (loops). אני מעריך שתוך שבוע עד שבועיים כולכם תהיו ברמה שווה בערך.

לאלה שקצת יותר מתורגלים ניתן לכתוב תוכנית שמדפיסה את המספרים הראשוניים בין 1 ל-100. למי שאינו זוכר או שלא ידע, מספר ראשוני הוא מספר (טבעי) שגדול מאחד ושמחלק אך ורק ב-1 ובעצמו. או כמו שנכתב בויקיפדיה: שלא ניתן להציגו כמכפלה של שני מספרים טבעיים קטנים ממנו, לדוגמה:

2, 7, 13, 23

זו תוכנית יותר מורכבת ממה שכתבנו עד כה, ולכן יש 'לתכנן' אותה, או לחשוב על האלגוריתם (או דרך הפעולה) שבה מצד אחד נעבור על כל המספרים מ-1 עד 100, ומצד שני כיצד לבדוק עבור כל מספר אותו נעבור, אם הוא ראשוני אם לאו. מי שלא סיים את התוכנית, מוזמן להמשיכה כעבודת בית (לא חובה).