

סיכום שיעור

חזרה על האלגוריתם של אוקלידס והפעלת: Visual C# management system

- הדגשנו בשיעור שהאלגוריתם של אוקלידס עובד על כל זוג מספרים שלמים וחיוביים.
- זוהי בעצם תכונה כללית של אלגוריתם נכון ומדויק, כלומר בכותבנו אלגוריתם, אנחנו צריכים להיות בטוחים שהוא פותר כל בעיה מהסוג שניסינו לפתור.
- זהו גם הכוח של תכנות מחשבים, כותבים תוכנית פעם אחת ומשתמשים בה פעמים רבות והיא אמורה לתת בכל שימוש את התוצאות המקוות ממנה.
- הזכרנו את הנושא של קלט ופלט, ונתנו דוגמה שבאלגוריתם הנ"ל, הקלט הוא שני מספרים טבעיים (שלמים וחיוביים), והפלט הוא מספר טבעי שהוא המחלק המשותף הגדול ביותר של מספרי הקלט.
- דיברנו על כך שבעצם התהליך של תכנות יכול להיות מחולק באופן הבא:

קלט –> עיבוד –> פלט

- העיבוד הוא פקודות התוכנית שמעבדות את הקלט ויוצרות את הפלט הרצוי. זהו הפורמט הכללי.
- יכולים להיות מקרים בהם אין קלט ויש פלט או מקרים בהם יש קלט ואין פלט ובהמשך ניראה דוגמאות לכך.
- בחלק השני של השיעור הפעלנו את התוכנה שניקראת: Visual C# management system
- דיברנו על סביבת הפיתוח באופן כללי (IDE או Integrated Development Environment) ואמרנו שעבורנו הייתה מספיקה תוכנה כמו Notepad או מעבד תמלילים כלשהו כדי לכתוב תוכניות ב C#, אבל כיוון שהתוכנה של ה-IDE הנ"ל היא חנימית, אנחנו בכל זאת נישתמש בה.
- לאחר הפעלת התוכנה של: Visual C# management system בחרנו פרוייקט חדש וציינו שהוא יהיה מסוג Terminal application שזה אומר תוכנית שעומדת בפני עצמה ויכולה להיות מופעלת (מורצת) על ידי איזכור שמה בלבד.
- כל סוגי הפרוייקטים האחרים האפשריים מיועדים לעבודה עם אתרים או מערכת חלונות והינם נושאים מתקדמים יותר שבד"כ נילמדים לאחר שכבר יודעים לתכנת באופן בסיסי כמו שאנחנו לומדים כאן.
- לאחר בחירת הפרוייקט מסוג: Terminal application ניפתח לפנינו מסך עם מספר שורות. שורות אלה מהוות למעשה תוכנית פשוטה וזיהינו בין היתר כמה מילות מפתח של C#

- מילות מפתח, כפי שכבר הזכרנו קודם, הן מילים של השפה עצמה. ל C# יש אוצר מילים של כ-50, ואנחנו נלמד מה מילים אלה (או פקודות אלה) יכולות לעשות, איך ומתי להשתמש בהן.
- שמנו לב על המסך של התוכנית שיש מספר מילים שמופיעות בכחול. סביבת העבודה שלנו שניקראת כאמור: Visual C# management system צובעת בכחול מילות מפתח של C# לנוחותינו. זה לא עוזר למחשב להבין, רק לנו.
- מילות המפתח שלמדנו עד כה היו: Namespace שזו הגדרה של מכלול תוכניות, או מרחב שלנו שמזהה משפחה של תוכניות (כמו שם משפחה), וגם את המילה: Using שעוזרת לנו להביא הגדרות שונות של C# שבהן נשתמש בתוכנית.
- בין היתר גם הזכרנו את המושג: משתנה ואמרנו שזהו מקום בזיכרון שיכול להכיל ערך. עד כה דיברנו על ערכים שהם מספרים, אבל בהמשך נראה גם סוגי ערכים אחרים.
- את המשתנים עליהם דיברנו באלגוריתם של אוקלידס, כינינו בשמות: תאים A, B, C זה כדי לחשוב בצורה ויזואלית כמו שמכניסים משהו לתוך תא דואר (פתק עם המספר שרשום עליו לדוגמה).
- בהמשך נגדיר משתנים בתוכנית ואלה יוכלו להכיל ערכים שיכולים להשתנות. בגלל זה הם ניקראים 'משתנים'.
- עוד נושא שהזכרנו זה על התפתחות העיסוק של תכנות מחשבים ושינוי שיטות התכנות מתכנות שניקרא פרוצדוראלי (שבו מבצעים פקודות אחת אחרי השנייה בתוכנית מלמעלה למטה) לתכנות שניקרא: "תכנות מונחה עצמים" שבו מגדירים מושגים בתוכנית שמקבילים לעצמים בעולם המציאותי. נראה בהמשך דוגמאות שימחישו את ההבדל.
- תכנות מונחה עצמים הוא בד"כ תפיסה לא פשוטה עבור מתכנת שעבד שנים על תכנות פרוצדוראלי ולכן משרד החינוך החליט שכדאי ללמד כל מי שמתחיל בתכנות, תכנות מונחה עצמים כדי שזה יהיה יותר טבעי.
- אנחנו ננסה ללמוד במקביל את שתי התפיסות, כי לדעתי אחת משלימה את השנייה. תכנות פרוצדוראלי מתאים לאפליקציות קטנות ובינוניות ותכנות מונחה עצמים הופך ליעיל יותר כאשר ישנם פרויקטים גדולים מאד.
- בשיעור הבא נחזור שוב על מילת המפתח: Using ונדבר גם על תכנות מונחה עצמים ולמה הוא בכלל התפתח.