



Powershell


במהלך הקורס תלמדו על השפה Powershell, לצורך היכרות בסיסית עם השפה ועם עקרונות קוד עליכם לקרוא את החומר בחוברת זו ובקישורים שיופיעו בהמשך, שימו לב את לימודי השפה בקורס נמשיך מנקודת הידע הבסיסית אליה תגיעו במסמך זה ולכן יש ללמוד היטב חומר זה על מנת להימנע מיצירת פער ביכולת ובידע שלכם בתחילת הקורס.

שפת תכנות (ובמקרה כאן - Powershell) היא שפה שעוצבה במיוחד כדי לתקשר, לשלוח בקשות ולשלוח הוראות למכונה כזו או אחרת (בדרך כלל מחשב). באמצעות שפות תכנות ניתן לכתוב תכנית שלמה שתבצע סדרה של פעולות במכונה. כלומר, באמצעות שפת התכנות נממש את האלגוריתם (סדר פעולות קבוע עם תוצר סופי) במחשב. כמו כל שפה גם לשפות תכנות יש חוקי תחביר שצריך לעקוב אחריהם ולכן חשוב ללמוד את החוקים והצורה בה שפת תכנות עובדת על מנת ליישם פעולות ואוטומציות שאדם רגיל לא היה יכול לבצע/היה מבצע בהרבה יותר זמן.

[בקישור הבא עליכם ללמוד את עמודים 1-12](#)

(יש ללחוץ ctrl ואז ללחוץ על הקישור בו זמנית עם העכבר)

שימו לב

על מנת להיכנס ל-Powershell במחשב שלכם אין צורך להוריד שום דבר (אלא אם אתם משתמשים במערכת הפעלה שאינה Windows) אלא רק ללחוץ על מקש  ה שבעזרתו תוכלו לחפש את Powershell במחשב שלכם בכך שפשוט תכתבו powershell ותוכלו להיכנס לתוצאת החיפוש המתקבלת. במידה ותרצו לכתוב סקריפט (רצף/אוסף של פקודות המקושרות אחת לשנייה) ולא להשתמש רק בפקודות בודדות עליכם להיכנס ל-Powershell ISE.

[קראו את המידע באתר הבא ובצעו את התרגולים הנמצאים בו](#)



Pseudo code

לפני שאנחנו ניגשים להתחיל לכתוב את הקוד שלנו, מומלץ לכתוב אותו קודם במה שנקרא Pseudo Code.

Pseudo code הוא קוד הכתוב בשפת אדם ולא בשפת תכנות, שבה לתאר איך הקוד הולך להיראות, אבל ללא חוקי כתיבה של שפת תכנות מסוימת.

שימו לב שכל עניין ה pseudo code הוא אינו חובה כלל אך הוא עוזר מאד בעיקר בהתחלה, מומלת להשתמש בו למשימות ותרגולים ארוכים.

בכתיבת קוד יש צורך גם בתכנון מקדים של מה אני רוצה שהקוד יעשה ואיך הוא יעשה את זה, לאחר היכרות מעמיקה עם שפה מסוימת התחביר שלה פחות מהווה מכשול אך הוא עדיין עלול להגביל את צורת החשיבה שלנו. במידה ונרצה לגשת לבעיה המציגים בפנינו ולמצוא פתרון יעיל הרבה פעמים יהיה יעיל יותר לעשות תוכנית עבודה במילים שלנו בה נתאר מה עושה כל שלב בקוד שלנו ולאחר תוכנית העבודה נראה איך ניישם אותה בקוד ובשפה שלנו בצורה יותר קלה, במקום לנסות לעשות את שני הדברים במקביל (תכנון הקוד + כתיבת הקוד).

ברגע שסיימתי לכתוב את ה Pseudo code שלי אני כבר יודע כל שלב ושלב בקוד, אפשר לחשוב עליו כמעין סקיצה, וכאשר הוא גמור כבר יש לנו את השלד של הקוד, עכשיו צריך פשוט להשתמש בסינטקס שפת התכנות עמה אנחנו עובדים.
לדוגמא:

המשימה - כתוב תוכנית אשר תדפיס תפריט למשתמש עם 4 אופציות, ולאחר בחירת המשתמש תבצע פעולה מסוימת ותחזור לתפריט.

Pseudo coden:

הדפסת התפריט:

אופציה 1

אופציה 2

אופציה 3

אופציה 4

קבלת קלט מהמשתמש.

אם המשתמש בחר באופציה 1:



מסלול בסמ"ח אלפא

בצע את פעולה מספר 1

אם המשתמש בחר באופציה 2:

בצע את פעולה מספר 2

אם המשתמש בחר באופציה 3:

בצע את פעולה מספר 3

אם המשתמש בחר באופציה 4:

בצע את פעולה מספר 4

חזור לתפריט

המשימה כמובן תהיה מפורטת יותר, ניקח את אופציה 1 לדוגמא:
אם המשתמש בחר אופציה 1 יש לקלוט מהמשתמש מספר, יש להדפיס אם הוא זוגי או אי זוגי.

Pseudo coden (של אופציה 1):

קבלת קלט מהמשתמש

אם המספר מתחלק ב2:

להדפיס שהמספר זוגי

אחרת:

להדפיס שהמספר לא זוגי



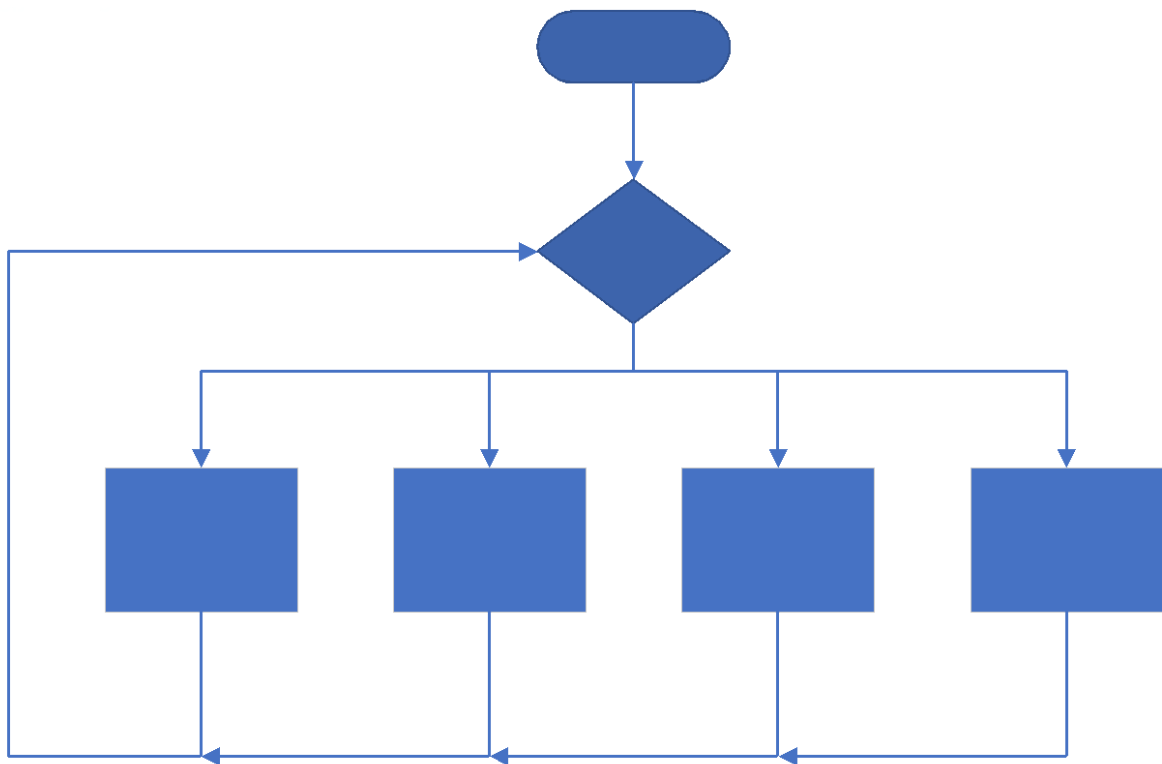
תרשימי UML

UML (Universal Modeling Language) היא שפה לאפיון לעיצוב של מערכות ומידע. השפה נמצאת בשימוש רחב בעולם התוכנה אבל ניתנת לניצול גם במערכות ומטלות בקנה מידה קטן.

תרשימי UML הם תרשימים אשר משתמשים באותן עקרונות ששפת UML מנחה כדי לייצג ויזואלית מערכות ותוכנות.

כאשר אנחנו ניגש למטלה מסוימת נרצה לקחת את הבקשות/שאלות שניתנות לנו ולהפוך אותן לתרחיש (מופשט או מורכב) אשר ינחה אותנו כמו מפה בתהליך הבנייה של הקוד שלנו.

ניקח למשל את המטלה הבאה (התרשימים נוצרו באמצעות תוכנה בשם visio) -
"כתוב תכנית אשר תציג למשתמש תפריט עם 4 אופציות ועבור כל אחת מן האופציות תבצע מטלה אשר אחריה היא תחזור לתחילת התפריט."



עיצוב זה נותן לנו איזושהי תמונה כללית ורחבה של התוצר הסופי שאנחנו רוצים להגיע אליו וכעת אחרי שיש לנו את זה אנחנו יכולים להיכנס ללוגיקה הפנימית של כל מקטע כדי להבין מה קורה בו (כמו שנעשה עם pseudo code).

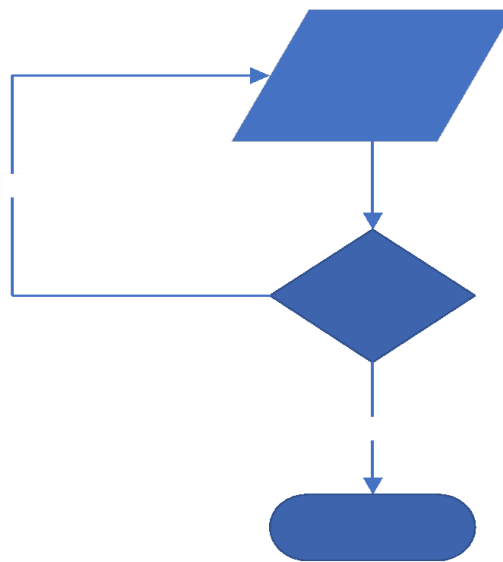


מסלול בסמ"ח אלפא

נגיד והפירוט של אופציה 1 היא "יש לבצע קליטת קלט מהמשתמש עד שהוא שווה לערך מסוים הנקבע מראש".

אנחנו נבין כי אנחנו נייצר איזושהי לולאה שבשלב מסוים תישבר ועל כן התרשים שלנו יראה בתצורה הבאה:

דוגמה לתצורה:



לסיכום-

על ידי שימוש בתרשימי UML אנחנו נוכל לסכם לעצמנו ולמדל את התוכן אותו נרצה לפתח בתצורה כזאת שיהיה לנו אמצעי ויזואלי להמשך התקדמות ומעקב אחרי מה שעשינו.

מתכנן לביצוע



מסלול בסמ"ח אלפא

לאחר שייצרנו לעצמנו איזושהי תכנית בין אם זה עם UML או עם pseudo code אנחנו צריכים לקחת את התכנון שלנו ולהמיר אותו לקוד בפועל.

השיטה הכי יעילה לעשות זאת היא לפעול בשלבים- לעקוב אחרי התבנית שבנינו שלב אחרי שלב כדי לוודא שיש לנו את עיקרי הלוגיקה שלנו. כאשר נבוא לממש את הדוגמה שבנינו לעיל אנחנו ניגש שלב אחרי שלב, תרשים אחרי תרשים ונניצר את התוצר שלנו.

DISCLAIMER

כלים אלו נועדו כדי לעזור לכם לבנות את הקוד בצורה טובה יותר, אך זכרו שאין סיבה לבזבז עליהם יותר מידי זמן מתוך כתיבת הקוד, כל התהליך הזה לא צריך לקחת לכם יותר מ15 דקות, אמנם לתכנן את הקוד זה חשוב ויכול מאוד לעזור אבל צריך לוודא שעדיין נשאר לכם זמן לכתוב את הקוד עצמו.