**תכנון קוד**

הקדמה

פתרון בעיה באמצעות קוד הוא דבר מורכב, שיש לתכנן אותו לפני הביצוע שלו. כשאנחנו ניגשים לפתור בעיה באמצעות קוד ישנם כמה שלבים לתהליך הזה. במהלך הקורס אתם תלמדו את כולם, אך כעת אנחנו נתמקד בשניים הראשונים, ונדבר בקצרה על השלישי:  
1) הבנת הבעיה  
2) תכנון הפתרון  
3) כתיבת הקוד

הבנת הבעיה

בשלב הזה עלינו לקרוא את תיאור הבעיה, ולוודא כי אנחנו מבינים מה מבקשים מאיתנו: נרצה להבין את הדרישה הכללית, כלומר מה המטרה שלנו, לאחר מכן נזהה את הקלט שלנו, את הפלט שאנו זקוקים לו ונזהה כל מיני מגבלות המוחלות עלינו או שיכולות לעלות בזמן כתיבת הקוד.

נשתמש בדוגמא לאורך המסמך -  
נניח והבעיה שאנו רוצים לפתור היא הבאה:   
**"בהינתן מספר שלם חיובי N, הדפס N שורות של כוכביות כאשר בשורה הראשונה כוכבית אחת, בשורה השנייה שתי כוכביות וכן הלאה."**

עכשיו ננסה להבין את הבעיה. מכיוון שזוהי בעיה פשוטה יחסית, קל לנו להבין את הדרישה הכללית ואת מטרת התוכנית מקריאה ראשונית. בבעיות מורכבות יותר, שלב זה דורש קריאה מעמיקה של הדרישות והבנה מלאה שלהן.

נעבור לשלב זיהוי הקלט והפלט שלנו.  
הקלט שלנו הוא מספר שלם וחיובי.  
הפלט שלנו הוא תבנית של כוכביות המודפסת למשתמש.  
מבחינת מגבלות, השאלה לא מציבה לנו מגבלות מסויימות ועל כן, המגבלה הראשונית היחידה שקיימת היא הידע שלנו. בבעיות מורכבות יותר, יכול להיות שהדרישה תציב לנו מגבלות כמו חומר ספציפי שאנחנו צריכים להשתמש בו, קוד קיים שצריך לעשות בו שימוש, דברים שאנחנו לא יכולים לעשות ואפילו מקרים שחייבים להתייחס אליהם.  
מבחינת מגבלות שיכולות לעלות לנו בזמן כתיבת הקוד, נצטרך לענות לעצמנו על שאלות כמו: "האם יש חלקים שעלינו לממש שתלויים אחד בשני וסדר המימוש שלהם משנה?", "האם אנחנו תלויים בקוד קיים ולכן צריכים להבין ולהכיר אותו לפני שנתחיל במימוש שלנו?" ועוד שאלות דומות אשר עוזרות לנו להבין האם יש דברים שעלינו להתכונן אליהם מראש כדי שלא יפתיעו אותנו/יפגעו בנו בזמן כתיבת הקוד. בשאלה שלנו, אין מגבלות כאלו, לכן, נמשיך לשלב הבא.

תכנון הפתרון

השלב הזה הוא שלב קריטי ביותר בפתרון בעיה באמצעות קוד, וגם הוא מורכב ממספר שלבים:

1. זיהוי החלקים המרכזיים בבעיה:

1. ריצה על המספר N  
2. הדפסת הכוכביות

1. פירוק כל חלק לתתי-בעיות קטנות יותר:
2. ריצה על המספר N

* נשתמש בלולאה על מנת לרוץ N פעמים

1. הדפסת הכוכביות
   * הדפסת שורה אחת של כוכביות
   * הדפסה של כמות כוכביות כמספר השורה
2. הדפסה של שורה אחת של כוכביות
   * נשתמש בלולאה פנימית על מנת לרוץ עד למספר הריצה הנוכחי של הלולאה החיצונית כדי להדפיס שורה אחת של כוכביות כמספר השורה הנוכחית
3. נזהה מקרי קצה:
   1. N שווה 0 – נדפיס כלום.
4. כעת נתחיל לממש את הפתרונות לתתי-בעיות, וכך לבסוף, נפתור את הבעיה הגדולה.

עברית מבנית (Pseudocode)

כלי נוסף שיכול לעזור לנו בתכנון הקוד ובהבנת הפתרון הוא עברית מבנית / Pseudocode.  
עברית מבנית היא תיאור ברמה גבוהה של תכנית מחשב שנכתבת בשפה טבעית ולא בשפת תכנות ספציפית. המטרה של עברית מבנית היא לספק דרך ברורה ומתומצתת לתאר תכנית מחשב בלי להתעכב על הסינטקס הספציפי של השפה.  
בדר"כ נכתוב עברית מבנית, כפי ששמה מעיד, בעברית. כמובן שניתן לכתוב גם בכל שפה.

עברית מבנית מהווה "גשר" בין המחשבות שלנו, והדרך שבה אנחנו רוצים לבצע את הדברים לבין הקוד עצמו. לחשוב כמו מחשב זה לא דבר שקורה ביום, ומטרת העברית מבנית לעזור לנו להתקרב לשפת התוכנה מבלי לעזוב את שפת האם שלנו.

עברית מבנית היא לא שפה רשמית ועל כן אין לה חוקים ספציפיים אבל כן קיימים מספר קווים מנחים:

* יש להשתמש בשפה פשוטה וברורה שתהיה קלה להבנה ע"י אחרים
* יש להשתמש בהזחה ובמרווחים על מנת להבהיר את מבנה התוכנית
* יש להשתמש במבני תכנות סטנדרטים כמו לולאות, תנאים ופונקציות כדי לתאר את לוגיקת התוכנית
* יש להשתמש במשתנים קבועים ומבני נתונים על מנת לייצג את המידע בתוכנית

נשתמש באותה הדוגמא שהתחלנו איתה:

**התחלה**

נקבל משתנה N כפרמטר

לכל מספר I (החל מ-1) עד למספר N (כולל):

הדפס שורת כוכביות

הדפס ירידת שורה

סיים לולאה

**סוף**

דוגמא יותר ספציפית:

**התחלה**

נקבל משתנה N כפרמטר

לכל מספר I (החל מ-1) עד למספר N (כולל):

לכל מספר J (החל מ-1) עד למספר I (כולל):

הדפס כוכבית

סיים לולאה

הדפס ירידת שורה

סיים לולאה

**סוף**

אותה הדוגמא רק ב-Pseudocode באנגלית:

//1st way, using descriptive words

Start

    get integer parameter N

    for each value from 1 until N

        print a line of "\*"

        go down a line

    end for

    return sum

End

//2nd way, using a syntax closer to a programming language

function printStarPattern(N) do

    for (set i = 1; i <= N; i++) do

        for (set j = 1; j <= i; j++) do

            print "\*"

        end

        println ""

    end

end

סיכום

אחרי שלב התכנון יגיע שלב המימוש. בשלב הזה נוציא את התוכנית שלנו לפועל ונתחיל ממש לממש את הקוד.

חשוב לציין, כי הרבה פעמים כשנגיע לשלב הזה, נגלה שלא משנה כמה תכננו מראש וכמה מחשבה השקענו, יש דברים שלא צפינו מראש, דרישות חדשות, מקרי קצה שלא חשבנו עליהם ואולי אפילו נמצא דרך יותר טובה לעשות את אותו הדבר.  
בגלל זה בדיוק חשוב לגלות גמישות ולא להיות נעולים על התכנון שלנו לחלוטין.

שלב התכנון הוא שלב **קריטי** שבאמצעותו נוכל להבין את המשימה יותר טוב, לצפות בעיות ולכתוב קוד נכון יותר, יעיל יותר בהספק טוב יותר.

שימו לב כי לאחר שתסיימו לקרוא את ה-ל.ע. הזה אתם לא תהפכו למומחי תכנון. שליטה בשלב התכנון מגיעה לאחר זמן וניסיון. הכי חשוב זה שלא תוותרו על השלב הזה ותשימו עליו דגש כדי שתוכלו להשתפר בו, בהתחלה זה ירגיש מוזר, לא יעיל ובזבוז של זמן אבל לטווח הארוך, היכולת לתכנן היא אחד מהדברים שמפרידים בין תוכניתנים ממוצעים/טובים לבין תוכניתנים מעולים.

נסו להשתמש בכלים ובטיפים שבמסמך זה כבר בתרגול המסכם ובתרגולים הבאים שיהיו לכם בקורס המקדים וגם במהלך הקורס.