

2. טיפוס נתונים מופשט

טיפוס נתונים מופשט - **טנ"מ** - הוא טיפוס נתונים המורכב **מתכונות ופעולות**. התכונות הם ערכים - קבועים או משתנים, המתארים אותו, והפעולות הן הפעולות אותן ניתן לבצע עליו. הצירוף של תכונות ופעולות הוא זה המייחד טיפוס נתונים מופשט אחד מטיפוסי נתונים מופשטים אחרים.

דוגמא לטיפוס נתונים מופשט: **בקבוק יין**.

לבקבוק יין יש **תכונות** כגון: סוג היין (אדום/לבן), מידת היובש (מתוק/יבש/חצי יבש), סוג הענבים, שנת הבציר, היצרן, כמות היין בבקבוק, אחוז האלכוהול, האם הבקבוק פקוק (ערך בוליאני) וכד'. אלו הן **תכונות** הטיפוס.

חלק מה**פעולות** האפשריות על בקבוק יין שכזה הן: מילוי הבקבוק ביין, מזיגת יין מהבקבוק, וברורים שונים כגון: שם היצרן, שנת הבציר, סוג היין בבקבוק, מידת היובש, האם הבקבוק פקוק ועוד.

הפעולות על טיפוס הנתונים המופשט נקראות **ממשק הטנ"מ**. על מנת שהשימוש בטנ"מ יהיה קל וברור, חשוב להגדיר את הפעולות בצורה מלאה ומדויקת. ההתייחסות אל הטנ"מ נעשית אך ורק באמצעות פעולות הממשק, והן האמצעי היחיד להשתמש בו.

בעיצוב תכנה אנו עוסקים בשני סוגי ממשקים – **ממשק עברי וממשק בסביבת העבודה**. הממשקים הם שקולים: הממשק העברי משמש באלגוריתם מילולי, ואילו הממשק בסביבת העבודה משמש בתכנית מחשב.

ממשק עברי הינו הממשק ברמת לקוח - מנתח מערכות, דהיינו כאשר מגיע לקוח, למשל מנהל ספריית ספרים, הרי סביר להניח שהוא ידרוש שהתוכנה שתעמוד לרשותו תכיל את הפעולה: הוספת-ספר-חדש-למאגר (ספר, מאגר-ספרים).

הממשק בסביבת העבודה לעומת זאת הינו התאמה של הממשק העברי לשפת התכנות.

לעיתים יש צורך לבצע התאמות בממשק בסביבת העבודה כדי להתאימו למגבלות שפת התכנות.

הממשק העברי

הממשק העברי כולל את החלקים הבאים :

- א. **קבועים** – פירוט כל הקבועים, אם קיימים כאלו, השייכים להגדרת הטנ"מ. הפירוט כולל: שם הקבוע, ערכו ומשמעותו בהגדרת הטיפוס.
- טיפוסים** - שם הטיפוס, ושמות כל הטיפוסים המרכיבים אותו. יש לציין את תפקידו של כל טיפוס.
- ב. **פעולות** – הפעולות המוגדרות על הטיפוס.

לדוגמא: נגדיר ממשק עברי לטיפוס **פחית-משקה** אשר תכונותיו: סוג המשקה, שם היצרן, כמות המשקה בפחית והמחיר לצרכן. נפח הפחית הוא קבוע – 330 מ"ל.

קבועים

נפח הפחית: can_volume = 330

טיפוסים

פחית משקה: can_type
 שם יצרן: maker_type
 סוג משקה: drink_type
 כמות משקה: מספר שלם
 מחיר משקה: מספר ממשי

פעולות

הפעולה מחזירה פחית-משקה ריקה מתוצרת maker מסוג drink.	אתחל-פחית-משקה (maker, drink)
פעולה המחזירה את שם היצרן של הפחית can. <u>הנחה</u> : הפחית מאותחלת.	אחזר-שם-יצרן (can)
פעולה המחזירה את סוג המשקה הנמצא בפחית can. <u>הנחה</u> : הפחית מאותחלת.	אחזר-סוג-משקה (can)
פעולה המחזירה את כמות המשקה הנמצא בפחית can. <u>הנחה</u> : הפחית מאותחלת.	אחזר-כמות-משקה (can)
פעולה המחזירה את מחיר פחית-המשקה can. <u>הנחה</u> : הפחית מאותחלת.	אחזר-מחיר-פחית (can)
פעולה המוסיפה/גורעת מכמות המשקה בפחית can, בהתאם לסימנו של vol. (אם vol חיובי והמקום <u>הפנוי</u> בפחית קטן מ-vol, אז הפחית תמולא עד סופה. אם vol שלילי, וכמות המשקה בפחית קטנה מהערך המוחלט של vol, הפחית תרוקן ממשקה.) <u>הנחות</u> : הפחית מאותחלת. vol ערך תקין.	עדכן-כמות-משקה (can, vol)
הפעולה מעדכנת את מחירה של הפחית can להיות price. <u>הנחה</u> : הפחית מאותחלת.	עדכן-מחיר (can, price)

- **הפרמטרים של הפעולות:** נקבע ב"עיצוב תכנה" כי בכותרת הפעולה יופיעו רק פרמטרים מאותחלים, ואילו ערכים המחושבים או מאותחלים בתוך הפעולה לא יופיעו בכותרת.
- פעולת האתחול קובעת את שדות הטני"מ להיות הערכים המועברים אל הפעולה. הפעולה **אתחל-פחית-משקה** מחזירה פחית משקה מאותחלת, כלומר, קובעת את שם-היצרן להיות Maker ואת סוג המשקה להיות Drink, אולם הפחית עצמה אינה אחד הפרמטרים של הפעולה, כיוון שבשעת הזימון עדיין אינה מאותחלת.
- עקרון הסתרת המידע קובע כי לתכניתן המשלב את טיפוס הנתונים בתכניתו אין מידע הנוגע לאופן המימוש של טיפוס הנתונים. הממשק מהווה עבורו "מדריך למשתמש".
- התיעוד לפעולה יכלול מידע לגבי הפרמטרים המועברים לפעולה, תפקידם, מה מבצעת הפעולה או מה היא מחזירה, ומהן ההנחות שמניחים לגבי הפרמטרים.
- בממשק לטני"מ יופיעו פעולות משלושה סוגים:
 - **פעולות אתחול (initialization)** - פעולות המאתחלות כל אחת מהתכונות. יש להבחין בין פעולת קלט ופעולת אתחול. הראשונה מקבלת ערך אותו נרצה לתת עבור תכונה מסויימת, השנייה גורמת לתכונה לקבל ערך זה.
 - **פעולות אחזור (get)** - פעולות המאחזרות מידע מתוך הטני"מ. כיוון שלמשתמש בטני"מ אין מידע לגבי אופן ייצוגו של טיפוס הנתונים, הדרך היחידה לאחזור מידע תעשה באמצעות פעולות האחזור.
 - **פעולות עדכון (set/update)** - פעולות המחליפות את המידע הקיים במידע חדש.
- עקרון הסתרת המידע קובע שפעולות עזר הנחוצות לביצוע פעולות הממשק לא ירשמו בממשק שהוא החלק הגלוי למשתמש ביחידה, אלא בחלק המוסתר בלבד - חלק המימוש. (ראה דוגמא "יתאריך" בהמשך).

ממשק בסביבת העבודה

ממשק בסביבת העבודה הינו ממשק הנכתב בשפת התכנות. בממשק זה יש לפרט את הכותרת של כל פעולה, בצירוף תיעוד מתאים. אין לפרט את אופן ההגדרה המדויק של הטיפוסים השונים המרכיבים את טיפוס הנתונים המופשט. הממשק אינו מכיל את מימוש הפעולות.

להלן ממשק בסביבת העבודה של טיפוס הנתונים המופשט **פחית-משקה** :

הממשק בשפת פסקל :

<pre>procedure can_init (var can : can_type; maker : maker_type; drink : drink_type);</pre> <p>פעולה המחזירה פחית משקה מאותחלת וריקה עם שם-יצרן maker , סוג-משקה drink , הכמות ומחיר המשקה 0. <u>הנחות</u> : maker ו- drink תקינים.</p>
<pre>procedure can_retrieve_maker (can : can_type; var maker : maker_type);</pre> <p>הפעולה מחזירה את שם היצרן של פחית המשקה can. <u>הנחה</u> : הפחית מאותחלת.</p>
<pre>procedure can_retrieve_drink (can : can_type; var drink : drink_type);</pre> <p>הפעולה מחזירה את סוג המשקה בפחית can. <u>הנחה</u> : הפחית מאותחלת.</p>
<pre>function can_retrieve_capacity (can : can_type) : integer;</pre> <p>הפעולה מחזירה את כמות המשקה הנוכחית בפחית can. <u>הנחה</u> : הפחית מאותחלת.</p>
<pre>function can_retrieve_price (can : can_type) : real;</pre> <p>הפעולה מחזירה את מחירה של הפחית can. <u>הנחה</u> : הפחית מאותחלת.</p>
<pre>procedure can_update_drink_capacity (var can : can_type; vol : integer);</pre> <p>פעולה המוסיפה/גורעת מכמות המשקה בפחית can, בהתאם לסימנו של vol. (אם vol חיובי והמקום <u>הפנוי</u> בפחית קטן מ- vol, אז הפחית תמולא עד סופה. אם vol שלילי, וכמות המשקה בפחית קטנה מהערך המוחלט של vol, הפחית תרוקן ממשקה.) <u>הנחות</u> : הפחית מאותחלת. vol ערך תקין.</p>
<pre>procedure can_update_price (var can : can_type; price : real);</pre> <p>הפעולה מעדכנת את מחירה של הפחית can להיות price. <u>הנחה</u> : הפחית מאותחלת.</p>

הערה : ניתן היה לקבוע את מחיר המשקה ואת הכמות ההתחלתית כבר בשלב האתחול, אולם בחרנו בדרך זו על מנת להמחיש את פעולות העדכון.

<p><code>can_type can_init (maker_type maker, drink_type drink);</code></p> <p>פעולה המחזירה פחית מאותחלת וריקה עם שם-יצרן maker , סוג-משקה drink, הכמות ומחיר המשקה 0. <u>הנחות</u> : maker ו- drink תקינים.</p>
<p><code>maker_type can_retrieve_maker (can_type can);</code></p> <p>הפעולה מחזירה את שם היצרן של פחית המשקה can. <u>הנחה</u> : הפחית מאותחלת.</p>
<p><code>drink_type can_retrieve_drink (can_type can);</code></p> <p>הפעולה מחזירה את סוג המשקה בפחית can. <u>הנחה</u> : הפחית מאותחלת.</p>
<p><code>int can_retrieve_capacity (can_type can);</code></p> <p>הפעולה מחזירה את כמות המשקה הנוכחית בפחית can. <u>הנחה</u> : הפחית מאותחלת.</p>
<p><code>float can_retrieve_price (can_type can);</code></p> <p>הפעולה מחזירה את מחירה של הפחית can. <u>הנחה</u> : הפחית מאותחלת.</p>
<p><code>void can_update_drink_capacity (can_type *can, int vol);</code></p> <p>פעולה המוסיפה/גורעת מכמות המשקה בפחית can, בהתאם לסימנו של vol. (אם חיובי והמקום הפנוי בפחית קטן מ-vol, אז הפחית תמולא עד סופה. אם שלילי, וכמות המשקה בפחית קטנה מהערך המוחלט של vol, הפחית תרוקן ממשקה.) <u>הנחות</u> : הפחית מאותחלת. vol ערך תקין.</p>
<p><code>void can_update_price (can_type *can; float price);</code></p> <p>הפעולה מעדכנת את מחירה של הפחית can להיות price. <u>הנחה</u> : הפחית מאותחלת.</p>

הערה : ניתן היה לקבוע את מחיר המשקה ואת הכמות ההתחלתית כבר בשלב האתחול, אולם בחרנו בדרך זו על מנת להמחיש את פעולות העדכון.

בעיה לדוגמא:

נתון מאגר של משקאות השמור במערך can_arr בגודל N . פתח ויישם אלגוריתם המקבל שם משקה $drink$, שם יצרן $maker$ ואת המאגר ומחזיר את מספר פחיות המשקה מסוג $drink$ של היצרן $maker$.

מספר-פחיות-משקה ($can, maker, drink$)

} פעולה המקבלת מאגר משקאות can_arr , שם-משקה $drink$ ושם-יצרן $maker$ ומחזירה את מספר פחיות המשקה מסוג $drink$ של היצרן $maker$ הקיימות במאגר.
הנחות: הנתונים תקינים.

$counter \leftarrow 0$

בזמן N פעולות:

$maker = can_arr[i]$ אם $can_arr[i] = maker$ ו- $can_arr[i] = drink$

$counter \leftarrow counter + 1$

חזרה

Pascal
(* פעולה המקבלת מאגר משקאות can_arr , שם-יצרן $maker$ ושם-משקה $drink$ ומחזירה את מספר פחיות המשקה מסוג $drink$ של היצרן $maker$ הקיימות במאגר. הנחות: הנתונים תקינים. *) function can_number (can_arr : arr_type; $maker$: maker_type; $drink$: drink_type): integer; var $i, counter$: integer; begin $counter := 0$; for $i := 1$ to N do if (can_retrieve_maker ($can_arr[i]$) = $maker$) and (can_retrieve_drink ($can_arr[i]$) = $drink$) then $counter := counter + 1$; can_number := $counter$; end;

```

/* פעולה המקבלת מאגר משקאות can_arr, שם-יצרן maker ושם-משקה drink
ומחזירה את מספר פחיות המשקה מסוג drink של היצרן maker הקיימות במאגר.
הנחות: הנתונים תקינים. */
int can_number (arr_type can_arr, maker_type maker, drink_type drink)
{
    int i, counter = 0;

    for (i = 0 ; i < N ; I++)
        if (can_retrieve_maker (can_arr[i]) == maker &&
            can_retrieve_drink (can_arr[i]) == drink)
            counter ++;
    return counter ;
}

```

מימוש בסביבת העבודה

מימוש הטנ"מ בסביבת העבודה פירושו כתיבת הייצוג והפעולות בשפת תכנות, במודול מיוחד הקרוי בפסקל **יחידת ספריה**. בשפת C לא קיים מושג מקביל, ולכן נאמץ מושג זה גם עבור C. חלק המימוש יכול להכיל פעולות עזר הנחוצות לפעולות הממשק אך אינן עומדות לרשותו של המשתמש בממשק.

הסבר כיצד לכתוב יחידת ספריה בשפת פסקל מצוי בפרק ב' בספר "עיצוב תכנה".
הסבר כיצד לכתוב יחידת ספריה בשפת C מצוי כאן בפרק 3 "יחידת ספריה ופרויקט ב-C".

דוגמא לטנ"מ: תאריך

נגדיר טנ"מ המטפל בתאריך.

קבועים וטיפוסים:

טיפוס-תאריך הכולל את השדות: יום, חודש, שנה.

פעולות היחידה: הפעולות הגלויות שביחידה:

פעולה הקולטת תאריך מן המשתמש. הפעולה מחזירה תאריך תקין.	קלט-תאריך
פעולה המקבלת תאריך ומציגה אותו על המסך. <u>הנחה</u> : התאריך מאותחל ותקין.	הצגת-תאריך (תאריך)
פעולה המקבלת שני תאריכים ומחזירה את מספר הימים שעבר בין התאריך המוקדם יותר לתאריך המאוחר יותר. מספר ימים זה מוגדר כ"מרחק" שבין התאריכים. <u>הנחה</u> : התאריך מאותחל ותקין.	מרחק-בין-שני-תאריכים (תאריך-1, תאריך-2)

<p>פעולה המקבלת תאריך ומספר שלם k, ומחזירה תאריך חדש שהוא התאריך המחושב k ימים מהתאריך שהתקבל. הנחה: התאריך מאותחל ותקין.</p>	<p>חישוב-תאריך-חדש (תאריך, k)</p>
---	---

לצורך מימוש פעולות היחידה יש צורך במשתנים ופעולות עזר נוספות. כמו פעולה הבודקת תקינות התאריך שהקליד המשתמש, פעולה המסייעת בחישוב מרחק שבין התאריכים וכד'. פעולות אלו יופיעו רק בחלק המוסתר של היחידה, בחלק המימוש, ראה פתרון מלא בפרק הבא.

שאלות

שאלה 2.1

במערכת המחשוב של רשת טלוויזיה עולמית, הוחלט לפתח מודול מיוחד המטפל בטיפוס הנתונים המופשט שעות. המודול דרוש לחישובים מידיים של זמן במקומות שונים בעולם ביחס למקומות אחרים וביחס לתחנה המקומית, ולהצגת זמנים מתאימים על מסך החדשות.
 א. כתוב ייצוג מתאים לטני"מ **שעות**.
 ב. כתוב ממשק עברי לטיפול בטני"מ **שעות** לפי דרישות הרשת.

■

שאלה 2.2

במחסן קיימת מערכת תכנה המנהלת את המלאי. עבור כל פריט סחורה המצוי במחסן שמורים פרטי המידע הכוללים: מספר קטלוגי, שם הפריט, כמות במלאי, כמות מקסימאלית, כמות קריטית (אם הכמות במלאי נמוכה מהכמות הקריטית, יש לבצע הזמנת פריטים עד להשלמה לכמות המקסימאלית).
 א. כתוב ייצוג מתאים לטני"מ **פריט-במחסן**.
 ב. כתוב ממשק עברי לטיפול בטני"מ **פריט-במחסן** לפי דרישות המחסן.

■

שאלה 2.3

טיפוס הנתונים **שיר** כולל את הנתונים הבאים: שם השיר, שם המבצע, שם חברת ההקלטה, שנת ההקלטה, אורך השיר, סגנון מוסיקלי.
 א. כתוב ייצוג מתאים לטני"מ **שיר**.
 ב. כתוב ממשק עברי לטיפול בטני"מ **שיר**.

■