SI 1080 part 15

תקן ישראלי ת"י 1080 חלק 15

April 2012

ניסן התשע"ב - אפריל 2012

ICS CODE: 35.060

35.020

טכנולוגיית המידע - הגדרות מונחים: שפות תכנות

Information technology - Vocabulary:
programming languages

מכון התקנים הישראלי
The Standards Institution of Israel

רח' חיים לבנון 42, תל-אביב 69977, טל' 03-6465154, פקס' 03-6412762, www.sii.org.il דק/

תקן זה הוכן ואושר על ידי הוועדה הטכנית 2110 - מונחי טכנולוגיית המידע, בהרכב זה:

אוניברסיטת בר-אילן	-	אריאל פרנק
אוניברסיטת תל-אביב	-	אורה קפלן
איגוד האינטרנט הישראלי	-	יונתן בן אברהם, דורון שקמוני
איגוד ישראלי לעיבוד אינפורמציה (איל"א)	-	מנחם דישון
האקדמיה ללשון העברית	-	מלכה זמלי, מלכה מוצ'ניק, אורה שורצולד
יבמ ישראל בע"מ	-	שלמה מסטבאום
לשכת מנתחי מערכות	-	אמנון שפירא (יו"ר)
מיקרוסופט ישראל	-	תמר שגיא
רשות השידור	-	דוד גומבוש
רשות ההסתדרות לצרכנות	-	יובל פרידמן

כמו כן תרמו להכנת התקן: אבישי בידני, דורון פינקלשטיין, חיים הירשהורן, יבגני זינגרמן, מיכל שחר, שמשון הופמן, שרה חשבי.

דנית לביא ריכזה את עבודת הכנת התקן.

הודעה על רוויזיה

תקן זה בא במקום

התקן הישראלי ת"י 1080 חלק 15 מאוקטובר 2009

מילות מפתח:

עיבוד נתונים, עיבוד מידע, אוצר מילים, מונחים, הגדרות, שפות תכנות.

Descriptors:

data processing, information processing, vocabulary, terms, definitions, programming languages.

עדכניות התקן

התקנים הישראליים עומדים לבדיקה מזמן לזמן, ולפחות אחת לחמש שנים, כדי להתאימם להתפתחות המדע והטכנולוגיה.

המשתמשים בתקנים יוודאו שבידיהם המהדורה המעודכנת של התקן על גיליונות התיקון שלו.

מסמך המתפרסם ברשומות כגיליון תיקון, יכול להיות גיליון תיקון נפרד או תיקון המשולב בתקן.

תוקף התקן

תקן ישראלי על עדכוניו נכנס לתוקף החל ממועד פרסומו ברשומות.

יש לבדוק אם התקן רשמי או אם חלקים ממנו רשמיים. תקן רשמי או גיליון תיקון רשמי (במלואם או בחלקם) נכנסים לתוקף 60 יום מפרסום ההודעה ברשומות, אלא אם בהודעה נקבע מועד מאוחר יותר לכניסה לתוקף.

סימון בתו תקן

כל המייצר מוצר, המתאים לדרישות התקנים הישראליים החלים עליו, רשאי, לפי היתר ממכון התקנים הישראלי, לסמנו בתו תקן:

זכויות יוצרים

© אין לצלם, להעתיק או לפרסם, בכל אמצעי שהוא, תקן זה או קטעים ממנו, ללא רשות מראש ובכתב ממכון התקנים הישראלי.

תוכן העניינים

15.00 - מושגים כלליים - GENERAL TERMS 3
15.01 - ערכים מילוניים - LEXICAL TOKENS 5
15.02 - הכרזות - DECLERATIONS 7
15.03 - עצמי נתונים - DATA OBJECTS 10
15.04 - טיפוסי נתונים - DATA TYPES 13
15.05 - משפטים וביטויים - STATEMENTS AND EXPRESSIONS 18
15.06 - חלקי תוכנית - PROGRAM PARTS 23
15.07 - משימות - TASKS 29
15.08 - ביצוע - EXECUTION 30
15.09 - תכנות כְּוּוּן-עצמים - OBJECT-ORIENTED PROGRAMMING 33
15.10 - תכונות ואופיינים - FEATURES AND CHARACTERISTICS 37
15.11 - תכנות מוּנַע-אירועים - EVENT DRIVEN PROGRAMMING 38
15.12 - יישומים - APPLICATIONS 39
איורים 44
רשימת מונחים עברית-אנגלית 48
רשימת מונחים אנגלית-עברית 60

הקדמה

תקן זה מבוסס על התקן הבין-לאומי ISO/IEC 2382-15 משנת 1999, ומביא מונחים נוספים שנאספו ממקורות שונים. תקן זה הוא חלק מסדרת התקנים הישראליים ת"י 1080, המביאה מונחים והגדרות מהתחומים השונים של טכנולוגיית המידע. סדרת התקנים ת"י 1080 מבוססת על סדרת התקנים הבין-לאומיים ISO 2382 המתעדכנת מפעם לפעם, ועל עבודת איסוף של מונחים חדשים המבוצעת על ידי הוועדה ממקורות שונים. המונחים הנשאבים מהמקורות השונים משמשים להכנת חלקים חדשים או לעדכון חלקים קיימים. המונחים מובאים באנגלית ובעברית, וההגדרות - בעברית.

חלקי סדרת ת"י 1080 **שיצאו עד כה לאור הם אלה**:

ת"י 1080 חלק 1 - מונחים בסיסיים
ת"י 1080 חלק 2 - פעולות אריתמטיות ולוגיות
ת"י 1080 חלק 3 - טכנולוגיית חומרה
ת"י 1080 חלק 4 - ארגון נתונים
ת"י 1080 חלק 5 - ייצוג נתונים
ת"י 1080 חלק 6 - הכנת נתונים ועיבודם
ת"י 1080 חלק 7 - תכנות מחשבים
ת"י 1080 חלק 8 - בקרה, כלילות ואבטחה
ת"י 1080 חלק 9 - תקשורת נתונים
ת"י 1080 חלק 10 - טכניקות ואַפשוֹרות של פעולה
ת"י 1080 חלק 11 - בקרה, קלט-פלט וציוד אריתמטי
ת"י 1080 חלק 12 - ציוד היקפי
ת"י 1080 חלק 13 - גרפיקה מחשבית
ת"י 1080 חלק 14 - אמינות, תחזוקתיות וזמינות
ת"י 1080 חלק 15 - שפות תכנות
ת"י 1080 חלק 16 - תורת המידע
ת"י 1080 חלק 17 - מסדי-נתונים
ת"י 1080 חלק 18 - עיבוד נתונים מבוזר
ת"י 1080 חלק 20 - פיתוח מערכות
ת"י 1080 חלק 23 - עיבוד תמלילים
ת"י 1080 חלק 24 - ייצור כליל-מחשב
ת"י 1080 חלק 25 - רשתות מקומיות (LAN)
ת"י 1080 חלק 26 - חיברור מערכות פתוחות (OSI)
ת"י 1080 חלק 27 - המשרד הממוחשב
ת"י 1080 חלק 28 - בינה מלאכותית - מושגי יסוד ומערכות מומחה
ת"י 1080 חלק 29 - בינה מלאכותית - זיהוי דיבור
ת"י 1080 חלק 30 - בינה מלאכותית – ראיית מחשב
ת"י 1080 חלק 31 - בינה מלאכותית - למידת מכונה
ת"י 1080 חלק 32 - דואר אלקטרוני
ת"י 1080 חלק 33 - היפרמדיה ומולטימדיה
ת"י 1080 חלק 34 - בינה מלאכותית – רשתות עצביות
ת"י 1080 חלק 35 - רישות
ת"י 1080 חלק 36 - למידה מתוקשבת
ת"י 1080 חלק 37 - בינה מלאכותית - מציאות מדומה
ת"י 1080 חלק 38 - רשתות חברתיות (בהכנה)
ת"י 1080 חלק 39 - שפות שאילתה (בהכנה)

זיהוי המונחים

כל מונח בתקן מסומן בקוד בן 6 ספרות. זוג הספרות הראשון (משמאל) מציין את מספר החלק בסדרה, זוג הספרות השני מציין את מספר הפרק וזוג הספרות השלישי מציין את מספר הסעיף (הפריט).

הוספות והשמטות בהשוואה לת"י 15-1080 **משנת** 2009

בתקן זה נוספו מונחים והגדרות שלא נכללו בת"י 15-1080 משנת 2009 (שהיה מבוסס על התקן הבין-לאומי ISO/IEC 2382-15 משנת 1999), ונעשו בו גם כמה תיקוני מונחים בעקבות החלטות האקדמיה ללשון העברית.

סוגריים מרובעים

מילה או מילים המופיעות בסוגריים מרובעים, אינן חלק מהמונח. הן מובאות לשם הבהרה בלבד.

כתיב מלא וכתיב חסר

המונחים נכתבו בכתיב חסר ובניקוד. ההגדרות נכתבו בכתיב מלא (חסר ניקוד).

הדגשה

מילים או ביטויים המודגשים **בעיבוי** בגוף הגדרה הם מונחים.

רשימות מונחים

בסוף התקן מובאות שתי רשימות מונחים: רשימת מונחים עברית-אנגלית ורשימת מונחים אנגלית-עברית.

תחום התקן

חלק זה של התקן הישראלי ת"י 1080 מביא מונחי שפות תכנות והגדרותיהם.

אזכורים

תקנים בין-לאומיים

ISO/IEC10646: 2011 - Information technology - Universal Coded Character Set (UCS)

תקנים ישראליים

ת"י 1080 חלק 1 – טכנולוגית המידע – הגדרות מונחים: מונחים בסיסיים

ת"י 1080 חלק 2 – טכנולוגית המידע – הגדרות מונחים: פעולות אריתמטיות ולוגיות

ת"י 1080 חלק 26 – טכנולוגית המידע – הגדרות מונחים: חיברור מערכות פתוחות (OSI)

15.00 - מושגים כלליים - General Terms

15.01.01 **מִלּוֹנִי** **lexical**

הניתן לתרגום לערך מובן למהדר.

15.00.02 **תְּכוּנָה** **property**

ערך המגדיר רכיב יסודי באופיו של עצם.

15.00.03 **כְּלָלֵי קְדִימוּת** **rules of precedence**

כללים לקביעת סדר הביצוע של פעולות מתמטיות.

15.00.04 **שִׁרְשׁוּר** **concatenation**

צירוף של כמה דברים למעקובת אחת.

דוגמה

בנייה של מען דואר אלקטרוני מרכיבים המחוברים באמצעות נקודה: sii.org.il.

15.00.05 **סִמּוּן בִּנְקֻדָּה** **dot notation**

בתחביר שפות תכנות - שימוש בנקודה להפרדה בין רכיבים של מען.

15.00.06 **מְבֻזָּר** **distributed**

המתקיים בכמה סביבות.

15.00.07 **תִּכְנוּת כְּוּוּן-עֲצָמִים מוּנַע אֵרוּעִים** **object-oriented event-driven programming**

OOED programming

תכנות כוון-עצמים שהעצמים שבו מבוקרים על ידי אירועים.

15.00.08 **שְׂפַת הַגְדָּרַת מִנְשָׂק** **interface definition language**

IDL [abb]

שפה המיועדת להגדרת מִנשָׂק בין תוכנית לבין יישום המזמן אותה.

15.00.09 **שְׂפַת סִמּוּן מִתְרַחֶבֶת** **Extensible Markup Language**

XML

XML [abb]

שפה תקנית פתוחה שנועד לתיאור ולארגון של נתונים.

הערות:

1. השפה משמשת להגדרת רכיבי נתונים בדף מארג (web), לחילופי מסמכים לעסקות באינטרנט, וליישומים אחרים.
2. התסדיר מבוסס על תגים דמויי HTML המוגדרים על ידי המְפַתֵּח.

15.00.10 **מוֹדֵל תִּכְנוּת יִשּׂוּם** **Application Programming Model**

APM

APM [abb]

מודל תכנות המגדיר כיצד להשתמש בתכונות של פלטפורמה וכיצד לשלב אותן כדי ליצור פתרונות משותפים ליישומים שונים.

15.00.11 **מוֹדֵל עַצְמֵי מִסְמָךְ** **Document Object Model**

DOM

DOM [abb]

מודל המתאר מסמך כעץ של צמתים המאפשר לתוכניות ותסריטים לטפלל (to manipulate) את התוכן, את המבנה ואת הסגנון של המסמך.

15.00.12 **הַגְדָּרַת טִפּוּס מִסְמָךְ** **Document Type Definition**

DTD [abb]

תיאור המבנה והמאפיינים של מחלקה של קובצי XML.

15.00.13 **תָּו נָטוּי** **italic**

תו בעיצוב נטוי ימינה.

15.00.14 **בָּג** **bug**

שגיאה בתוכנית המונעת אותה מלתפקד כראוי.

15.00.15 **זְרִימַת נְתוּנִים** **data flow**

מעבר נתונים בין **קבועים**, **משתנים** וקבצים, על ידי ביצוע הוראות, **נהלים**, **פִּרקָנים** או תוכניות.

15.00.16 **סֵמֶל לְתַרְשִׁים זְרִימָה** **flowchart symbol**

בתרשים זרימה, אלמנט גרפי המייצג פעולות, אמצעים ותהליכים.

הערה: ראו ציור 3.

15.01 - ערכים מילוניים - Lexical Tokens

15.01.01 **עֵרֶךְ מִלּוֹנִי** **lexical token**

lexical element

lexical unit

מחרוזת תווים באלף-בית של שפת תכנות, המייצגת יחידה בסיסית של משמעות.

דוגמות:

ליטרל, כגון 2G5, או מזהה כגון last-name בשפת Pascal.

15.01.02 **צֶרֶף-שָׂפָה** **language construct**

חלק תוכנית, נכון תחבירית, שניתן ליצור אותו מכמה **ערכים מילוניים** לפי כללי שפת תכנות.

דוגמות:

<תגית>, If Than, +, GOTO.

15.01.03 **מְזַהֶּה** **identifier**

שם המשמש לזיהוי אלמנט בתוכנית.

דוג**מות**: שמות של משתנים, מערכים, רשומות, תוויות, נהלים, שגרות.

הערה:

המזהה מורכב בדרך כלל מאות, ואחריה ייתכנו אותיות, ספרות או תווים אחרים.

15.01.04 **מְזַהֶּה מֻבְנֶה** **predefined identifier**

מזהה המוגדר מראש כחלק משפת תכנות.

דוגמות:

מילה שמורה, Sin.

הערה:

אם מזהה מובנה אינו שמור, אז הכרזה המשתמשת בו מגדירה מחדש את משמעותו לתחום ההכרזה.

15.01.05 **מִלָּה שְׁמוּרָה** **reserved word**

מזהה מובנה שתוכניתן אינו יכול להגדיר מחדש.

הערה:

לא בכל שפות התכנות יש מילים שמורות.

15.01.06 **תּוֹחֵם** **delimiter**

תו או תווים המציינים את ההתחלה או את הסיום של מחרוזת תווים הנחשבת יחידה אחת.

הערות:

1. תווים מיוחדים או מילים שמורות עשויים לשמש כתוחמים.
2. שונה מ"מפריד".

15.01.07 **מַפְרִיד** **separator**

תוחם המונע פירושם של **ערכים מילוניים** סמוכים, או של יחידות תחביריות סמוכות, כפריט יחיד.

דוגמות:

תו מרווח, תו בקרת תסדיר.

15.01.08 **עֲמִיסַת מַשְׁמָעֻיּוֹת** **overloading**

עֲמִיסָה

ייחוס של יותר ממשמעות אחת ל**ערך מילוני**.

דוגמה:

המשמעות של הערך המילוני "+" יכולה להיות חיבור מספרים שלמים, חיבור מספרים ממשיים, איחוד קבוצות, שרשור וכדומה.

15.01.09 **הֲפָגַת עֲמִימוּת** **disambiguation**

הפניה אל המשמעות המתאימה, מתוך כמה משמעויות אפשריות להיקרות מסוימת של **צרף-שפה** בתוכנית.

15.01.10 **תָּוִית** **label**

בשפות תכנות - **מזהה** של מקום בתוכנית.

15.01.11 **הֶעָרָה** **comment**

remark

תמליל שאינו משפיע על התוכנית והמסומן על ידי **צרף-שפה** ייעודי.

דוגמות:

הסברים לקורא, נתונים למערכת תיעוד אוטומטי.

15.01.12 **קֶבוּעַ תָּו** **character constant**

קבוע מטיפוס תו.

15.01.13 **קֶבוּעַ מַחְרֹזֶת** **string constant**

קבוע מטיפוס מחרוזת.

הערה:

מחרוזת יכולה להיות ריקה.

15.01.14 **תְּלַת-תָּו** **trigraph sequences**

מעקובת בת שלושה תווים המתורגמת על ידי קדם-מעבד של שפה לתו יחיד.

15.01.15 **קֶבוּעַ מֻבְנֶה** **intrinsic constant**

קבוע מובנה בשפת תכנות שנתכן כדי לחסוך בזמן משתמש ולהקל את קריאת הקוד.

דוגמה: π.

15.01.16 **קֶבוּעַ סִמִּין** **literal constant**

קֶבוּעַ לִיטֵרָל

קבוע שערכו נקבע בגוף התוכנית.

15.01.17 **זָקִיף** **sentinel**

קבוע המציין סיום לולאה.

15.01.18 **סֵמֶל הַתְחָלָה אוֹ סִיּוּם** **start/stop symbol**

סמל המציין התחלה או סיום של פעולה או ישות.

15.01.19 **קֶבוּעַ סִמְלִי** **symbolic constant**

קבוע הנשמר בזיכרון ומזוהה על ידי תווית.

15.02 - הכרזות - Declerations

15.02.01 **הַכְרָזָה** **declaration**

בשפות תכנות - **צֶרף-שפה** מפורש המצהיר על **מזהים** בתוכנית ומציין כיצד יפורשו על ידי המהדר.

דוגמות: הכרזות על טיפוסי נתונים, ארגון הזיכרון, חבילות או משימות.

הערה:

בכמה שפות תכנות, ההכרזות נחשבות משפטים.

15.02.02 **אֵזוֹר הַהַכְרָזוֹת** **declarative part**

declarative region

חֲטִיבַת הַנְּתוּנִים **data division**

חלק התוכנית המיועד ל**הכרזות**.

15.02.03 **בְּרֵרַת מֶחְדָּל** **default**

תכונה, ערך או בררה, המשמשים כשאין קביעה מפורשת אחרת.

15.02.04 **הַכְרָזָה מִשְׁתַּמֶּעֶת** **implicit declaration**

הכרזה המשתמעת מהימצאות **מזהה** המצביע על עצם שתכונותיו נקבעות על ידי **בררת מחדל**.

15.02.05 **מֻבְנֶה** **predefined**

built-in

intrinsic

המוגדר מראש בשפת התכנות.

דוגמות:

בשפת PL/1 - הפונקצייה SIN; בשפת Fortran - טיפוס הנתונים INTEGER.

15.02.06 **תְּחוּם הַכְרָזָה** **scope**

scope of a declaration

חלק התוכנית שבו ה**הכרזה** תקפה.

15.02.07 **נְתוּנִים מְשֻׁתָּפִים** **shared data**

נתונים הנגישים לכמה **פִּרקָנים** (**modules**).

15.02.08 **תְּחוּם הַכְרָזָה דִּינָמִי** **dynamic scope**

תחום ההכרזה הנוצר כאשר **פִּרקָן** חסר **הכרזות** מפעיל בעת ביצועו פרקנים או חלקי פרקנים אחרים המכילים הכרזות.

15.02.09 **תְּחוּם הַכְרָזָה סְטָטִי** **static scope**

תחום הכרזה המוגבל ל**פִּרקָן** (module) שבו נעשתה ה**הכרזה**, ושאינו מושפע מקריאות לפרקנים אחרים.

15.02.10 **מְקוֹמִי** **local**

ש**תחום ההכרזה** שלו מוגבל לַ**פִּרקָן** (**module**) שבו הוכרז.

15.02.11 **כְּלָלִי** **global**

ש**תחום ההכרזה** שלו הוא כל ה**פִּרקָנים** (**modules**) של תוכנית.

15.02.12 **חִיצוֹנִי** **external**

המוגדר מחוץ ל**פִּרקָן** (module) הפונה אליו.

15.02.13 **סְטָטִי** **static**

המתקיים ושומר על ערכו לאחר סיום הביצוע של שגרה.

15.02.14 **דִּינָמִי** **dynamic**

המוקצה מחדש עם כל כניסה **לתחום ההכרזה** שלו ושומר על ערכו רק במשך הביצוע בתחום הכרזה זה.

דוגמה:

הערך של התכונה "אורך" של עצם נתונים באורך משתנה, הוא ערך דינמי.

15.02.15 **אֹרֶךְ חַיִּים** **lifetime**

חלק זמן הביצוע שבו מתקיים **צֶרֶף-השפה**.

15.02.16 **נִרְאוּת** **visibility**

תכונה של **צרף-שפה** המאפשרת להתייחס אליו במקום מסוים בפִּרקָן (module).

15.02.17 **אֵזוֹר נִרְאוּת** **visibility region**

חלק של תוכנית שבו ניתן להתייחס ל**צֶרֶף-שפה** מסוים.

15.02.18 **הַכְרָזַת מִשְׁתַּנֶּה** **variable declaration**

הכרזה היוצרת משתנה המוגדר בדרך כלל באמצעות טיפוס ושם.

15.02.19 **הַכְרָזָה כְּלָלִית** **global declaration**

הכרזה בחלק הכללי של תוכנית, התקפה בכל ה**פִּרקָנים** (modules) שלה.

15.02.20 **הַכְרָזָה מְקוֹמִית** **local declaration**

הכרזה בקטע של תוכנית, התקפה רק באותו קטע.

15.02.21 **מִשְׁתַּנֶּה לֹא-מֻכְרָז** **undeclared variable**

משתנה שלא הוכרז במשפט מתאים.

15.02.22 **מִשְׁתַּנֶּה סְטָטִי** **static variable**

משתנה מקומי המוקצה בעת הידור התוכנית, שמתקיים ושומר על ערכו לאחר סיום הביצוע של שגרה.

הערה:

ראו "משתנה דינמי".

15.02.23 **מִשְׁתַּנֶּה דִּינָמִי** **dynamic variable**

משתנה המוקצה מחדש עם כל כניסה ל**תחום ההכרזה** שלו ושומר על ערכו רק במשך הביצוע בתחום.

הערה:

ראו "משתנה סטטי".

15.02.24 **מִשְׁתַּנֶּה מְקוֹמִי** **local variable**

משתנה שמיש בתוך תחום ההכרזה שבו הוכרז בלבד.

15.02.25 **מִשְׁתַּנֶּה כְּלָלִי** **global variable**

משתנה שמיש בכל מקום בתוכנית.

15.02.26 **פָּרָמֶטֶר מַעַן** **reference parameter**

משתנה המוכרז בנוהל לשם קבלת מען של ארגומנט.

15.02.27 **פָּרָמֶטֶר עֵרֶךְ** **value parameter**

משתנה המוכרז בנוהל לשם קבלת ערך של ארגומנט.

15.02.28 **מֻגְדָּר-מִמּוּשׁ** **implementation defined**

המוגדר במיוחד (או מוגדר באופן שונה) בכל מימוש של התוכנית.

הערה:

מאפיין של תוכנית.

15.02.29 **תְּלוּי-מִמּוּשׁ** **implementation dependent**

שאין הכרח להגדירו במיוחד לכל מימוש, והיכול להיות שונה במימושים שונים של תוכנית.

הערה:

מאפיין של תוכנית.

15.02.30 **מִתְאַר פָּרָמֶטְרִים** **parameter profile**

רשימת הפרמטרים של שיטה, שבה לכל פרמטר פורמלי נלווה שם המחלקה שלו.

15.02.31 **תֵּאוּר הַסְּבִיבָה** **environment description**

צרף-שפה המתאר מאפיינים שאינם חלק מרצף פקודות התוכנית, אך משפיעים על ביצועה.

דוגמות:

אופייני מחשב; תכונות קבצים מיוחדות; מִנשקים עם תוכניות אחרות.

15.02.32 **שֶׁטַח** **area**

בשפות תכנות - אזור המיועד לאחסון **עצמי נתונים**, שמצורף אליו מנגנון המאפשר הכנסתם לתוכו, גישה אליהם ומחיקתם.

15.02.33 **מִשְׁתַּנֶּה מִתְעַדְכֵּן** **substitution variable**

משתנה בטופס שתוכנו מוגדר על ידי משתנה כללי.

15.03 - עצמי נתונים - Data Objects

15.03.01 **מִבְנֵה נְתוּנִים** **data structure**

יוחסה פיזית או לוגית בין יחידות נתונים לבין הנתונים.

15.03.02 **עֶצֶם נְתוּנִים** **data object**

בשפות תכנות - אלמנט של **מבנה נתונים** כגון קובץ, מערך או אופרנד.

הערה:

עצם נתונים יכול להיות קבוע או משתנה.

15.03.03 **מִשְׁתַּנֶּה** **variable**

צרף-שפה הנקבע על ידי **הכרזה** או **הכרזה משתמעת** והמורכב מ**מזהה**, מקבוצת **תכונות נתונים**, ממען אחד או יותר ומ**ערכי נתונים**, ושבו היוחסה (relation) בין המענים לבין ערכי הנתונים יכולה להשתנות.

15.03.04 **עֵרֶךְ נְתוּנִים** **data value**

אלמנט מתוך קבוצת **עצמי נתונים** מוכרזת, שבהקשר מסוים מסומך ל**צֶרף-שפה** כגון **משתנה** או **טיפוס נתונים**.

15.03.05 **קָבוּעַ** **constant**

צֶרף-שפה הנקבע על ידי **הכרזה** או הכרזה משתמעת והמורכב מ**מזהה**, מקבוצת **תכונות נתונים**, ממען אחד או יותר ומ**ערך נתונים** אחד בלבד.

15.03.06 **מִצְרָף** **aggregate**

בטכנולוגיית המידע, אוסף מובנה של רכיבים ש**מבני הנתונים** שלהם יכולים להיות זהים או שונים.

הערה:

בדרך כלל במבנה טבלה.

15.03.07 **עֵרֶךְ מִצְרָף** **aggregate value**

ערך הנתונים המסומך ל**מִצְרָף**.

15.03.08 **מִצְרָף בְּגֹדֶל מְשֹׁעָר** **assumed-size aggregate**

מִצרף המשמש כ**פרמטר פורמלי** שתחומי ציוניו, כולם או מקצתם, נובעים מ**פרמטר ממשי** מתאים.

15.03.09 **מִצְרָף בְּגֹדֶל מִתְכַּוְנֵן** **adjustable-size aggregate**

מִצרף המשמש כ**פרמטר פורמלי** שתחומי ציוניו **דינמיים** - כולם או מקצתם.

15.03.10 **מַעֲרָךְ** **array**

בשפות תכנות - אוסף סָדוּר של רכיבים זהים שאפשר להתייחס לכל רכיב או קבוצת רכיבים מתאימה שבתוכו באופן אקראי וללא תלות באחרים.

15.03.11 **פֶּלַח מַעֲרָךְ** **array sliceslice**

פֶּלַח

חלק של **מערך** המורכב מרצף תאים לאורך ממד כלשהו.

הערה:

פלח מערך יכול להיות גם פעולה בסיסית.

15.03.12 **חֵלֶק מִשְׁתַּנֶּה** **variant part**

חלק של רשומה המורכב מ**עצמי נתונים** ש**טיפוסי הנתונים** המוכרזים שלהם יכולים להיות שונים.

הערה:

המספר וההרכב של עצמי הנתונים בחלק המשתנה של הרשומה יכולים להיות שונים.

15.03.13 **רְשׁוּמָה מִשְׁתַּנָּה** **variant record**

רשומה הכוללת **חלק משתנה**.

הערה:

הרשומה עשויה לכלול מבחינים המציינים את טיפוסי הנתונים בחלק המשתנה.

15.03.14 **מַבְחִין** **discriminant** [n]

צרף-שפה דמוי **פרמטר** המציין את **מבנה הנתונים** שבו יש להשתמש ב**רשומה משתנה** נתונה.

15.03.15 **פָּרָמֶטֶר** **parameter**

בשפות תכנות - **צרף-שפה** המשמש להעברת **עצמי נתונים** או **ערכי נתונים** בין **פִּרקָנים**.

15.03.16 **פָּרָמֶטֶר מַמָּשִׁי** **actual parameter**

actual argument

פרמטר כגון **ביטוי**, **מזהה** או **צרף-שפה** אחר, המשמש ב**קריאה** או ביצירת **אִדְגוּם גנרי** כללי כדי לסמך **עצמי נתונים** ל**הכרזות** המתאימות.

15.03.17 **פָּרָמֶטֶר פּוֹרְמָלִי** **formal parameterdummy argument**

פרמטר המוגדר ב**הכרזה** של **פִּרקָנים** מסוימים, שאליו מסמכים את ה**פרמטר הממשי** בעת **קריאה** או בעת יצירת **אִדְגוּם גנרי**.

15.03.18 **סִמּוּךְ פָּרָמֶטְרִים** **parameter association**

התאמת ה**פרמטרים הפורמליים** ל**פרמטרים הממשיים** המפורטים בעת **קריאה** או בעת יצירת **אִדגום גנרי**.

15.03.19 **תְּכוּנַת נְתוּנִים** **data attribute**

אופיין מוגדר-מראש של **טיפוס נתונים**, **עצם נתונים**, **פִּרקָן** או **צֶרף-שפה** אחר.

דוגמות:

לטיפוס ממשי עשויה להיות התכונה Precision (דַּיְק) שערכי הנתונים שלה הם Single (יחיד) או Double (כפול). למשימה עשויה להיות התכונה Terminated (מופסקת) המניבה תוצאת "אמת" כאשר המשימה הסתיימה ותוצאת "שקר" במקרים אחרים.

15.03.20 **שִׁיּוּךְ שֵׁם** **name qualification qualification**

אמצעי להתייחסות ל**צֶרף-שפה** בתחום של חלק מתוכנית, על ידי התייחסות אל החלק ואל **מזהה** שהוכרז לצֶרף-השפה באותו חלק.

דוגמות:

משתמשים בשיוך שם כדי להתייחס אל מרכיבים של רשומה, כגון התייחסות אל מרכיב B של רשומה A בשפת קובול, אל חברים בספרייה ואל צרפי-שפה בפִּרקָן (מודול).

15.03.21 **כִּנּוּי** **alias**

מזהה חלופי של **צֶרף-שפה**.

15.03.22 **מַצְבִּיעַ** **pointer**

בשפות תכנות - **עצם נתונים** ש**ערך הנתונים** שלו הוא מען של עצם נתונים אחר.

הערה:

ראו ציור 1.

15.03.23 **מַצְבִּיעַ רֵיק** **null pointer**

מצביע שבמפורש אינו מצביע על שום **עצם נתונים**.

הערה:

מצביע ריק מיוצג על ידי "nil", "null" וכדומה, לפי שפת התכנות.

15.03.24 **נְתוּנִים לֹא-תְּקֵפִים** **invalid data**

נתונים שאינם קבילים ביישום וייתכן שאינו יכול לעבדם.

15.03.25 **נְתוּנִים תְּקֵפִים** **valid data**

נתונים קבילים ביישום.

15.03.26 **תַּת-מַחְרוֹזֶת** **substring**

מחרוזת שהיא חלק של מחרוזת אחרת.

15.03.27 **מַחְרוֹזֶת רֵיקָה** **empty string**

null stringzero-length string

מחרוזת ללא תווים.

15.03.28 **תַּת-מַעֲרָךְ** **subarray**

מערך שהוא חלק של מערָך אחר.

15.03.29 **מַעֲרָךְ דִּינָמִי** **dynamic array**

מערך שגודלו יכול להשתנות במהלך הרצת התוכנית.

15.03.30 **אַרְכִיטֶקְטוּרָה אֶחִידָה לִמְתַוֵּךְ שְׁמוֹת עֲצָמִים** **Common Object Request Broker Architecture**

CORBA [abb]

CORBA

מודל של עצמים מבוזרים שאינו תָּלוי שפת תכנות.

הערה:

המודל הוגדר על ידי קבוצת ניהול עצמים OMG – (Object Management Group).

15.03.31 **תֵּבַת עֶצֶם** **object box**

תיבת שבה מוצגים השם והטיפוס של העצם שנבחר.

15.03.32 **עֶצֶם מְקֻבָּע** **immutable object**

עצם שאין לשנותו.

15.03.33 **מְתַוֵּךְ שְׁמוֹת עֲצָמִים** **Object Request Broker**

ORB [abb]

שירות המאפשר לעצמים הבנויים **בארכיטקטורה אחידה למתווך שמות עצמים** לאתר זה את זה ולתקשר ביניהם.

15.03.34 **עֵרֶךְ לֹא פָּרִיק** **atomic value**

עֵרֶךְ נְתוּנִים שלא ניתן לפרק אותו יותר.

15.04 - טיפוסי נתונים - Data Types

15.04.01 **טִפּוּס נְתוּנִים** **data type**

datatype

קבוצה מוגדרת של **עצמי נתונים** בעלי **מבנה נתונים** מסוים וקבוצת פעולות מותרות, כך שעצמי נתונים אלה יכולים לשמש אופרנדים בביצוע של כל אחת מפעולות אלה.

דוגמה:

לטיפוס הנתונים "מספר שלם" מבנה פשוט ביותר, שכל היקרות שלו, המכונה בדרך כלל ערך, מיוצגת על ידי מספר מתוך טווח מוגדר של מספרים שלמים והפעולות המותרות כוללות פעולות אריתמטיות רגילות על מספרים שלמים אלה.

הערות:

1. בהקשר של שפות תכנות, המונח "טיפוס נתונים" בא במקום המונח המיושן "סוג נתונים".
2. ראו ציור 1.

15.04.02 **טִפּוּס מֻפְשָׁט** **abstract type**

ADT [abb]

מודל מתמטי שבו **מבנה נתונים** מתואר על ידי רשימת פעולות או תכונות הזמינות במבנה הנתונים ועל ידי המאפיינים הפורמליים של אותן פעולות, כשהמִנשקים מופרדים מהמימוש הפנימי.

15.04.03 **טִפּוּס כָּמוּס** **encapsulated type**

טיפוס נתונים שיש לו מִנשקים מוגדרים גלויים לציבור ואִילו הפעולות המסומכות אליו ממומשות באופן פרטי.

15.04.04 **טִפּוּס סְקָלָרִי** **scalar type**

יפוס נתונים שכל היקרות שלו מייצגת כמות המבוטאת על ידי ערך יחיד (סקלר).

הערות:

1. בשפת Ada טיפוס סקלרי הוא תת- **טיפוס פשוט** הכולל **טיפוסים סודרים** (כגון נמנה, תו, לוגי, שלם) וקבועים כגון ממשי.
2. בשפת Pascal טיפוס סקלרי זהה לטיפוס פשוט, וכולל טיפוסים סודרים (כגון שלם, נמנה, תו, לוגי) וטיפוסים ממשיים.
3. ראו ציור 1.

15.04.05 **טִפּוּס פָּשׁוּט** **simple type**

טיפוס נתונים המייצג ערך לא-פריק.

הערות:

בניגוד לטיפוסים מורכבים כגון **טיפוס רשומה** ו**טיפוס מערך**, המייצגים מבני נתונים.

בשפת Ada טיפוס פשוט כולל טיפוסים סקלריים כגון: קבוע (ממשי, נקודה צפה), סודֵר (שלם, נמנה, תו, לוגי).

ראו ציור 1.

15.04.06 **טִפּוּס לא-פָּרִיק** **atomic type**

טיפוס נתונים שכל **עצם נתונים** שלו הוא **ערך נתונים** יחיד שאינו ניתן לחלוקה.

15.04.07 **טִפּוּס לוֹגִי** **logical type**

Boolean type

Bool type

טיפוס נתונים שכל **עצם נתונים** שלו עשוי לקבל רק ערכים לוגיים (בדרך כלל "אמת" או "שקר") ושאפשר להפעיל עליו אופרטורים לוגיים (Boolean operators) בלבד.

הערה:

ראו גם: טיפוס תו, טיפוס נמנה, טיפוס שלם וטיפוס ממשי.

15.04.08 **טְוָח** **range**

בשפות תכנות - קבוצת **ערכי נתונים** סמוכים מ**טיפוס סקלרי**.

15.04.09 **טִפּוּס מַמָּשִׁי** **real type**

טיפוס נתונים שכל **עצם נתונים** שלו מייצג מספר ממשי, ייתכן על ידי קירוב.

דוגמה:

לייצוג המספר העשרוני 0.1 במערכת בינרית יש מספר אין-סופי של ספרות.

הערות:

1. טיפוסים ממשיים הם טיפוסי נקודה קבועה או טיפוסי נקודה צפה.
2. ראו ציור 1.

15.04.10 **טִפּוּס נְקֻדָּה קְבוּעָה** **fixed-point type**

implied decimal type

טיפוס ממשי שכל **עצם נתונים** שלו מבוטא בשיטת ייצוג בנקודה קבועה.

15.04.11 **טִפּוּס נְקֻדָּה צָפָה** **floating-point type**

טיפוס ממשי שכל **עצם נתונים** שלו מבוטא בשיטת ייצוג בנקודה צפה.

הערה:

ראו ציור 1.

15.04.12 **טִפּוּס סוֹדֵר** **ordinal type**

discrete type

טיפוס נתונים שכל **עצם נתונים** שלו מייצג איבר של קבוצה סדורה הניתנת למנייה.

הערות:

הטיפוסים הסודרים של שפת Pascal הם: נמנה (enumerated), תו, שלם, לוגי (מספר בולֵאני) ונִמְנֶה. הטיפוסים הסודרים של שפת Ada הם נִמְנֶה (תו או לוגי) או שלם.

ראו ציור 1.

15.04.13 **טִפּוּס צִיּוּן** **index type**

טיפוס סוֹדֵר שכל **עצם נתונים** שלו מייצג ציון של מקום בתוך **מערך**.

15.04.14 **טִפּוּס שָׁלֵם** **integer type**

טיפוס סוֹדֵר שכל **עצם נתונים** שלו מייצג מספר שלם ב**טווח** נתון.

הערה:

ראו ציור 1.

15.04.15 **טִפּוּס נִמְנֶה** **enumeration type**

enumerated type

טיפוס סוֹדֵר שכל **עצם נתונים** שלו נמנה במפורש בעת הכרזת **טיפוס הנתונים**.

הערה:

ראו ציור 1.

15.04.16 **טִפּוּס מִסְפָּרִי** **numeric type**

טיפוס סקלרי שכל **עצם נתונים** שלו מייצג מספר שלם או קירוב של מספר ממשי.

הערה:

ראו ציור 1.

15.04.17 **טִפּוּס תָּו** **character type**

טיפוס סודֵר שכל **עצם נתונים** שלו מייצג תו.

הערה:

ראו ציור 1.

15.04.18 **טִפּוּס מַחְרוֹזֶת** **string type**

טיפוס נתונים שכל **עצם נתונים** שלו מייצג מחרוזת.

הערה:

ראו ציור 1.

15.04.19 **טִפּוּס מַצְבִּיעַ** **pointer type**

access type

טיפוס נתונים שכל **עצם נתונים** שלו מייצג **מצביע**.

הערה:

ראו ציור 1.

15.04.20 **טִפּוּס מַעֲרָךְ** **array type**

טיפוס נתונים מורכב שרכיביו הם מאותו טיפוס.

הערות:

1. אפשר לארגן טיפוסי מערך ולהתייחס אליהם כאילו היו רכיביהם מסודרים בעמודות, בשורות וכדומה.
2. ראו ציור 1.

15.04.21 **טִפּוּס רְשׁוּמָה** **record type**

טיפוס נתונים מורכב שרכיביו הם שדות או **טיפוסי רשומות** אחרים.

דוגמה:

רשומת פרטים אישיים עשויה לכלול נתונים אישיים ערוכים בשדות או בתתי-רשומות בתוכה.

הערות:

1. טיפוס רשומה מגדיר קבוצת ערכים ופעולות. אֶדגם של טיפוס רשומה כזה יכול לכלול ערכים שהם עצמם רשומות.
2. ראו ציור 1.

15.04.22 **טִפּוּס רְשׁוּמָה מִשְׁתַּנֶּה** **variant record type**

טיפוס רשומה בעל **חלק משתנה** המפרט מבחָר של הרכבי רכיבים.

15.04.23 **תַּת-טִפּוּס** **subtype**

טיפוס נתונים הנגזר מטיפוס נתונים אחר על ידי הפעלת **אילוץ** אחד או יותר.

15.04.24 **טִפּוּס בָּסִיס** **base type**

host type

underlying type

טיפוס נתונים שממנו בונים טיפוס נגזר.

הערה:

ראו "טיפוס הורה".

15.04.25 **אִלּוּץ** **constraint**

התאמה של **טיפוס נתונים** המגבילה את ה**טווח** שלו או את פעולותיו.

15.04.26 **טִפּוּס פְּרָטִי** **private type**

בתוך תוכנית - **טיפוס נתונים** שהמבנה שלו, קבוצת ערכיו ופעולותיו מוגדרים, אך זמינותו מוגבלת למרכיבים מיוחסים (privileged) של אותה תוכנית.

דוגמה:

בשפת Ada, רק הצבות, שוויונים ואי-שוויונים הם זמינים למשתמשים, ונוסף עליהם זמינות למשתמשים פעולות מיוחדות שנעשו נגישות בהוראה מפורשת.

15.04.27 **טִפּוּס מֻגְבָּל** **limited type**

טיפוס פרטי שבו רק פעולות או תכונות-נתונים שהוכרזו במפורש, זמינות מחוץ לחלק התוכנית הכולל אותן.

15.04.28 **טִפּוּס הוֹרֶה** **parent type**

טיפוס נתונים המשמש תבנית ליצירת טיפוסי נתונים חדשים.

הערה:

ראו "טיפוס בסיס" ו"טיפוס נגזר".

15.04.29 **טִפּוּס נִגְזָר** **derived type**

טיפוס נתונים שערכי נתוניו ופעולותיו הם עותקים משוכפלים של **ערכי נתונים** ופעולות של **טיפוס הורה** קיים.

הערה: ראו "טיפוס הורה".

15.04.30 **הֶמְרַת טִפּוּס** **type conversion**

type casting

בשפות תכנות - התמרת הייצוג של **ערך נתונים** מ**טיפוס נתונים** אחד לערך נתונים מטיפוס נתונים אחר, המבוצעת בדרך כלל כדי למנוע אי-התאמה לא חוקית של טיפוסי נתונים.

הערה:

פעולת ההמרה בין טיפוסים מספריים מותרת לעיתים קרובות, אך עלולה לגרום להפסדי דיוק (accuracy) או דִּיּוּק (precision) או שניהם.

15.04.31 **הַטְפָּסָה חֲזָקָה** **strong typing**

אכיפת הדרישה ש**טיפוסי הנתונים** של אופרנדים ב**צֶרֶף-שפה** חייבים להיות תואמים לטיפוסי הנתונים של הפעולה או כאלה שעברו **המרת טיפוס** לפני שהפעולה בוצעה.

דוגמה:

בשפת Ada, הטפסה חזקה הופכת את החיבור 3+2.5 לפעולה לא חוקית, מאחר ש-3 הוא מספר שלם ואילו 2.5 הוא מספר ממשי.

15.04.32 **הַטְפָּסָה חַלָּשָׁה** **weak typing**

החלשת הכללים של **הטפסה חזקה**.

הערה:

הטפסה חלשה עשויה לכלול או לא לכלול המרת טיפוס משתמעת.

דוגמה: הטפסה חלשה עשויה להתיר חיבור של מספר שלם עם מספר בייצוג בנקודה צפה ללא צורך בהמרת טיפוס מפורשת של אחד האופרנדים.

15.04.33 **טִפּוּס מֻבְנֶה** **predefined type**

טיפוס נתונים המזוהה באמצעות **מזהה** מוגדר מראש, ששפת תכנות מספקת פעולות נאותות לגביו.

15.04.34 **טִפּוּס סְתָמִי** **universal type**

טיפוס הנתונים של ליטרלים מספריים וטיפוס הנתונים של תוצאת כמה פעולות מוגדרות מראש, המשמשים לעמידה בדרישות של **הטפסה חזקה**.

דוגמה:

בשפת Ada, הכרזת מספר (ללא טיפוס נתונים) מניחה טיפוס סתמי.

15.04.35 **עֲלוּם-שֵׁם** **anonymous**

מתייחס ל**עצם נתונים** שאין לו **הכרזה** מפורשת של **טיפוס נתונים**.

15.04.36 **תַּסְדִּיר** **format**

בשפות תכנות - **צֶרֶף-שפה** המפרט באמצעות תווים את הייצוג של **עצמי נתונים** ברשומה, בקובץ, במסר, בהתקן אחסון או באפיק שידור.

הערה:

הפועל: תִּסְדֵּר.

15.04.37 **תְּמוּנָה** **picture**

בשפות תכנות - **צֶרֶף-שפה** המתאר את התסדיר של **עצמי נתונים** מ**טיפוס מחרוזת** באמצעות דגם ליטרל המורכב ממחרוזת תווים.

15.04.38 **טִפּוּס לֹא-מֻשְׁלָם** **incomplete type**

טיפוס נתונים שהוכרז אך לא הוגדר.

15.04.39 **טִפּוּס דַּיִּק-יָחִיד** **single-precision type**

טיפוס נתונים המאחסן מספרים בשיטת ייצוג בנקודה צפה בדיק יחיד (32 סיביות).

15.04.40 **טִפּוּס דַּיִּק-כָּפוּל** **double-precision type**

טיפוס נתונים המאחסן מספרים בשיטת ייצוג בנקודה צפה בדיק כפול (64 סיביות).

15.04.41 **הֶמְרַת טִפּוּס מְפֹרֶשֶׁת** **explicit type casting**
explicit type conversion

המרת טיפוס לפי דרישה מפורשת של התוכנית.

הערה:

ראו "המרת טיפוס משתמעת".

15.04.42 **הֶמְרַת טִפּוּס מִשְׁתַּמַּעַת** **automatic type conversion**
implied type conversion
coercion

המרת טיפוס הנעשית אוטומטית כדי לאפשר שימוש בַּערך בהקשר המתבקש.

הערה:

ראו "המרת טיפוס מפורשת".

15.04.43 **טִפּוּס מִשְׁתַּנֶּה** **variant type**

טיפוס נתונים המאפשר אחסון נתונים מטיפוסים שונים.

15.05 - משפטים וביטויים - Statements and expressions

15.05.01 **מִשְׁפָּט** **statement**

יחידה תחבירית המסתיימת בצורה מפורשת והמייצגת **הכרזה** או קביעה של יחידת עבודה, הכוללת זיהוי של הפעולות לביצוע, של אופרנדים (אם יש) לשימוש בביצוע אותן פעולות, וייעוד לתוצאות.

15.05.02 **מִשְׁפָּט פָּשׁוּט** **simple statement**

משפט שאינו כולל משפט אחר.

15.05.03 **מִשְׁפָּט מֻרְכָּב** **compound statement**

משפט הכולל בתוכו כמה משפטים התחומים כך שכל אחד מהם שקיל תחבירית ל**משפט פשוט**.

15.05.04 **מִשְׁפַּט הַצָּבָה** **assignment statement**

הַצָּבָה **assignment**

משפט פשוט המחליף את **ערך הנתונים** הנוכחי של **משתנה** בערך נתונים חדש המפורט ב**ביטוי**.

הערה:

הפועל: הציב.

15.05.05 **מִשְׁפַּט יְצִיאָה** **exit statement**

משפט פשוט המשמש לסיום הביצוע של **צֶרֶף-השפה** שבו הוא מופיע.

15.05.06 **מִשְׁפַּט חֲזָרָה** **return statement**

צֶרֶף-שפה בתוך **פִּרקָן** המציין את הסיום של מעקובת ביצוע (או כמה מעקובות כאלה) באותו פִּרקָן והגורם לקפיצה לנקודה שצוינה בפִּרקָן הקורא, וייתכן שהוא גם מעביר אליו תוצאות.

15.05.07 **חָזַר** **return** [v] [intransitive]

ביצע **משפט חזרה** הגורם לקפיצה לתוכנית הקוראת.

15.05.08 **הֶחְזִיר** **return** [v] [transitive]

סיפק **ערך נתונים** לתוכנית הקוראת בעת ביצוע **משפט חזרה**.

15.05.09 **כְּנִיסָה** **entry**

נקודת ההתחלה של מעקובת ביצוע הנמצאת בתחילת **תת-תוכנית** או במקום אחר המצוין על ידי **שם כניסה** בתת-התוכנית.

15.05.10 **שֵׁם כְּנִיסָה** **entry name**

מזהה המציין את נקודת ההתחלה של מעקובת ביצוע.

15.05.11 **מִשְׁפַּט "עֲבוֹר"** **goto statement**

מִשְׁפַּט goto

משפט פשוט המציין מעבר מפורש של הבקרה של תוכנית ממקומה במעקובת הביצוע למשפט מטרה המזוהה בדרך כלל על ידי **תווית**.

15.05.12 **מִשְׁפָּט לֹא-מֻתְנֶה** **unconditional statement**

מִשְׁפַּט צִוּוּי **imperative statement**

משפט המבוצע ללא תנאי.

15.05.13 **מִשְׁפַּט תְּנַאי** **conditional statement**

משפט מורכב הכולל **ביטוי תנאי** ומעקובת של משפטים שביצועם מותנה בתוצאת ההערכה של ביטוי התנאי.

15.05.14 **בִּטוּי תְּנַאי** **conditional expression**

ביטוי שתוצאת ההערכה שלו קובעת אם תנאי מוגדר מתקיים.

הערה:

ביטוי תנאי משמש בדרך כלל לבחירת מעקובת ביצוע.

15.05.15 **מִשְׁפַּט "אִם"** **if statement**

מִשְׁפַּט if

משפט תנאי הגורם לביצוע של אחד **המשפטים** המרכיבים אותו על פי ערך האמת של **ביטוי התנאי**.

15.05.16 **מִשְׁפַּט "מִקְרֶה"** **case statement**

מִשְׁפַּט case

משפט תנאי הגורם לביצוע אחד מבין כמה **משפטים** חלופיים על פי הערך של **ביטוי תנאי**.

15.05.17 **מִשְׁפַּט חִזְרוּר** **iteration statement**

loop statement

משפט מורכב הכולל מנגנון לבקרת ביצוע נשנה של המשפטים שלו.

15.05.18 **צֶרֶף "כָּל-עוֹד"** **while-construct**

צֶרֶף while

צֶרֶף-שפה לבקרת חִזְרוּר המגדיר מבחן לביצוע לפני כל צעד חִזְרוּר.

15.05.19 **צֶרֶף "עַד-אֲשֶׁר"** **until-construct**

צֶרֶף until

צֶרֶף-שפה לבקרת חִזְרוּר המגדיר מבחן לביצוע אחרי כל צעד חִזְרוּר.

15.05.20 **צֶרֶף "בַּעֲבוּר"** **for-construct**

צֶרֶף for

צֶרֶף-שפה לבקרת חִזְרוּר המגדיר את המבחן לביצוע בקרת החִזְרוּר, המבוסס בדרך כלל על **משתנה בקרת לולאה** ועל הוראת השינוי שלו שיש לבצע בין כל שני צעדי חִזְרוּר.

15.05.21 **מִשְׁפַּט "בַּצַּע כָּל-עוֹד"** **do while statement**

מִשְׁפַּט do while **repeat while statement**

מִשְׁפַּט repeat while **perform while statement**

מִשְׁפַּט perform while

משפט חִזְרוּר שבו בקרת החִזְרוּר משולבת ב**צֶרֶף while**.

15.05.22 **מִשְׁפַּט "בַּצַּע עַד-אֲשֶׁר"** **until statement**

מִשְׁפַּט until **repeat until statement**

perform until statement

משפט חִזְרוּר שבו בקרת החִזְרוּר משולבת ב**צֶרֶף עד-אשר**.

15.05.23 **מִשְׁפַּט "בַּצַּע"** **perform statement**

מִשְׁפַּט perform

בשפת Cobol - **משפט מורכב** המורה להעביר את הבקרה **לשגרה** אחת או יותר ולהחזירה בתום הביצוע.

15.05.24 **מִשְׁפַּט גּוּשׁ** **block statement**

מעקובת כרוכה של **משפטים** שניתן להתייחס אליה כאל יחידה תחבירית אחת ושעשוי להיות לה **מזהה**.

הערה:

משפט גוש הוא הרכיב התחבירי הבסיסי של שפות מובנות בגושים.

15.05.25 **מִשְׁפַּט קְרִיאָה לְנֹהַל** **procedure-call statement**

קְרִיאָה לְנֹהַל **procedure call**

משפט פשוט המספק ל**נוהל** את ה**פרמטרים הממשיים** והמפעיל אותו.

הערה:

בניגוד ל"קריאה לפונקצייה".

15.05.26 **מִשְׁפַּט קְרִיאַת כְּנִיסָה** **entry-call statement**

משפט פשוט המתיר ל**משימה** לבקש **מפגש** עם משימה אחרת.

15.05.27 **מִשְׁפַּט הַשְׁהָיָה** **delay statement**

משפט פשוט המשמש להשהיית הביצוע של **משימה** הכוללת בקשת השהיה.

15.05.28 **מִשְׁפַּט נְטִישָׁה** **abort statement**

משפט פשוט הגורם סיום שגוי של **משימה** אחת או יותר ועל ידי כך מונע כל **מפגש** נוסף עם משימה כזו.

15.05.29 **מִשְׁפַּט הַעֲלָאַת חָרִיג** **raise statement**

משפט פשוט הגורם להיקרות של חריג והמפיץ אותו.

15.05.30 **מִשְׁפַּט קַבָּלָה** **accept statement**

משפט מורכב בתוך משימת שרת הגורם לה להמתין עד ש**משימה** אחרת או **התוכנית הראשית** יבצעו **משפט קריאת כניסה** ל**סינכרון משימות**.

15.05.31 **מִשְׁפַּט בְּרִירָה** **select statement**

משפט מורכב המאפשר ל**משימה** קוראת או למשימה נקראת לבחור בין דרכי פעולה חלופיות, או להמתין.

15.05.32 **מִשְׁפַּט הַמְתָּנָה בּוֹרֶרֶת** **selective-wait statement**

משפט ברירה הממתין ל**קריאה** מ**משפט קריאת כניסה** לפני שהוא מבצע את מעקובת המשפטים שלו.

15.05.33 **בִּטוּי** **expression**

צֶרֶף-שפה המגדיר חישוב של **ערך נתונים** על פי אופרנד אחד או יותר.

הערה:

האופרנדים יכולים להיות ליטרלים, מזהים, קריאות לפונקצייה וכדומה.

15.05.34 **רַב-טִיפּוּסִים** **mixed mode** [adj]

mixed type [adj]

מתייחס ל**ביטוי** המכיל שני **טיפוסי נתונים** או יותר.

15.05.35 **בִּטוּי לוֹגִי** **Boolean expression**

צֶרֶף-שפה המגדיר חישוב של ערך לוגי.

15.05.36 **קְדִימוּת אוֹפֶרָטוֹרִים** **operator precedence**

כלל המגדיר את **סדר הביצוע** של אופרטורים בתוך **ביטוי**.

הערה:

הכלל עשוי לציין גם את כיוון הערכת הביטוי.

15.05.37 **צֶרֶף הַחְלָטָה** **decision construct**

decision structure

selection structure

צרף-שפה המנחה את המחשב להחליט על סמך השוואה ולאחר מכן להסתעף לקוד מתאים לצורך הביצוע.

15.05.38 **אִגּוּד** **union**

מבנה נתונים המאוחסן פעם אחת והמאפשר גישה אליו לפי טיפוסי נתונים שונים.

15.05.39 **עֲמִיסַת אוֹפֶרָטוֹר** **operator overload**

הוספת משמעות לאופרטור רגיל.

15.05.40 **אוֹפֶרָטוֹר לוֹגִי** **logical operator**

אופרטור שכל אופרנד ותוצאה שלו מקבלים ערך אחד מבין שני ערכים.

הערה:

ההגדרה זהה להגדרה שבת"י 1080 חלק 2.

דוגמות: אופרטור בול (Boolean operator) כגון "גם", "או", "לא".

15.05.41 **אוֹפֵרָטוֹר יֶחְסָה** **relational operator**

סמל המשמש לקביעת יוחסה בין אופרנדים בתוך ביטוי.

דוגמות:

אופרטור יוחסה	פירוש
=	שווה
<>	לא שווה
>	גדול מ...
<	קטן מ...
>=	גדול מ... או שווה לו
<=	קטן מ... או שווה לו

15.05.42 **מַצְבִּיעַ חָכָם** **smart pointer**

אופרטור מצביע עמוס-משמעויות המציינות דרישה להחזרת מצביע.

הערה:

ראו "עמיסת-משמעויות".

15.05.43 **מִשְׁפַּט טְעִינָה** **load statement**

הוראה הטוענת טופס לזיכרון ללא הצגתו.

15.05.44 **מִשְׁפַּט פְּרִיקָה** **unload statement**

הוראה המסירה מצג טופס מהזיכרון ומהמרקע.

15.05.45 **צֶרֶף סִעוּף** **branch construct**

צרף-שפה המפרט, באמצעות ייחוס ל**תוויות**, ברירה בין **סדרי ביצוע** שונים.

15.05.46 **צֶרֶף חִזְרוּר** **iteration construct**

loop construct

repetition structure

repetition construct

צרף-שפה המגדיר תחביר של מנגנון לבקרת ביצוע נשנה של חלק בתוכנית.

15.06 - חלקי תוכנית - Program Parts

15.06.01 **פִּרְקָן** **module**
מוֹדוּל **program unit**

חלק של תוכנית שפותח כך שיהיה נבדל או בר-זיהוי בכל הקשור לפעולות כגון הידור, כריכה או ביצוע, והעשוי להדד (to interact) עם תוכניות אחרות או עם חלקים אחרים של תוכניות.

הערות:

1. למונח "פרקן" ("מודול") עשויות להיות משמעויות שונות בשפות תכנות שונות.
2. ראו ציור 2.

15.06.02 **גּוּף** **body**

בשפות תכנות - **צֶרף-שפה** המהווה את החלק הבציע של **משפט** או של **פִּרְקָן**.

15.06.03 **תַּת-תָּכְנִית** **subprogram**

פִּרְקָן מזוהה הנקרא במהלך תזרים הבקרה מתוכנית אחרת או מפִּרְקָן אחר באמצעות **צֶרף-שפה** מיוחד, ושממנו חוזר תזרים הבקרה לתוכנית הקוראת או לפִּרְקָן הקורא.

15.06.04 **שִׁגְרַת הֶמְשֵׁךְ** **coroutine**

קבוצת פקודות שכאשר היא נקראת שנית לאחר ביצוע אחד, היא מחדשת את פעולתה במקום שממנו חזרה בביצוע הקודם.

15.06.05 **קְרִיאָה** **call**

בשפות תכנות - הוראה להעביר את הבקרה מפִּרְקָן אחד לפִּרְקָן אחר, ובדרך כלל משתמע מזה שהבקרה תוחזר לפִּרְקָן הקורא.

הערה: בקריאה מצוינים בדרך כלל פרמטרים המועברים אל הפִּרְקָן הנקרא וממנו.

15.06.06 **קָרָא** **call** [v]

ביצע **קריאה**.

15.06.07 **קְרִיאָה לְפִי שֵׁם** **call by name**

קריאה שבה הפִּרְקָן הקורא מספק לפִּרְקָן הנקרא את השמות של **פרמטר** אחד או יותר שיש להעריך, בכל פעם שהפִּרְקָן הקורא משתמש בפרמטר המסומך.

15.06.08 **קְרִיאָה לְפִי מַעַן** **call by reference**
call by address
call by location

קריאה שבה הפִּרְקָן הקורא מספק לפִּרְקָן הנקרא את המענים של ה**פרמטרים** המועברים.

הערה:

בקריאה לפי התייחסות, הפִּרְקָן הנקרא יכול לשנות את הערכים של הפרמטרים המאוחסנים בפִּרְקָן הקורא.

15.06.09 **קְרִיאָה לְפִי עֵרֶךְ** **call by value**

קריאה שבה הפִּרְקָן הקורא מספק לפִּרְקָן הנקרא את הערכים הממשיים של ה**פרמטרים** המועברים.

הערה:

בקריאה לפי ערך, הפרקן הנקרא אינו יכול לשנות את הערכים של הפרמטרים המאוחסנים על ידי הפרקן הקורא או בעבורו.

15.06.10 **קְרִיאָה לְתַת-תְּכְנִית** **subprogram call**

קריאה המפעילה **תת-תוכנית**.

דוגמות:

משפט קריאה לנוהל, קריאה לפונקצייה.

15.06.11 **תַּת-שִׁגְרָה** **subroutine**

תת-תוכנית שאינה מחזירה ישירות **ערכי נתונים**, אלא כחלק ממנגנון העברת **פרמטרים**.

15.06.12 **פּוּנְקְצִיָּה** **function**

בשפות תכנות - **תת-תוכנית**, בדרך כלל בעלת **פרמטרים** פורמליים, המפיקה **ערך נתונים** שאותו היא מחזירה לנקודת ה**קריאה** שלה.

דוגמה:

הפונקצייה SIN מחזירה את הסינוס של x כאשר קוראים לה עם הפרמטר x.

15.06.13 **קְרִיאָה לְפוּנְקְצִיָּה** **function call**

צֶרֶף-שפה המספק את ה**פרמטרים** הממשיים הנחוצים ל**קריאה** של **פונקצייה** ולביצועה.

הערות:

1. אפשר להשתמש בקריאה לפונקצייה כאופרנד של ביטוי או כפרמטר ממשי של קריאה לתת-תוכנית.
2. בשונה מ"משפט קריאה לנוהל".

15.06.14 **קְרִיאָה לְמִפְגָּשׁ** **transaction call**

קריאה לפונקצייה המאפשרת ל**משימה** לבקש **מפגש** עם משימה אחרת.

הערה:

המונח אינו זהה למונח "תנועה" שהוא תרגום של transaction בהקשרי מסד נתונים, עיבוד תנועות ועדכון קבצים.

15.06.15 **תַּת-יְחִידָה** **subunit**

גוף של **פִּרְקָן** המהודר בנפרד.

15.06.16 **תּוֹתָב** **body stub**

מבנה שלדי של **גוף**, המציין שהחלק הבציע של **פִּרְקָן** מוגדר ב**תת-יחידה** אחרת.

15.06.17 **חִבּוּר** **connection**

בשפות תכנות - טכניקה המאפשרת הידוד בין **פִּרְקָנִים**, בייחוד בין **קריאות-לנוהל** לבין נהלים אסינכרוניים.

15.06.18 **סִמּוּךְ פָּרָמֶטְרִים לְפִי שֵׁם** **named parameter association**

assignment by name

ב**קריאה לתת-תוכנית** - קריאה בשמות מפורשים ל**פרמטרים פורמליים** בהתאמה ל**פרמטרים ממשיים** כדי ליצור **סימוך פרמטרים**.

הערות:

1. בהצבה לפי שם, אפשר להציג את הפרמטרים הממשיים בכל סדר שהוא.
2. בשונה מ"סימוך פרמטרים לפי מקום".

15.06.19 **סִמּוּךְ פָּרָמֶטְרִים לְפִי מָקוֹם** **positional parameter association**

בקריאה לתת-תוכנית - התאמת **הפרמטר הממשי** ל**פרמטר פורמלי** הנמצא באותו מקום **בהכרזה** של תת-התוכנית.

הערה:

בשונה מ"סימוך פרמטרים לפי שם".

15.06.20 **מַצַּב פָּרָמֶטֶר פּוֹרְמָלִי** **formal parameter mode**

מאפיין המציין האם ניתן להעריך **פרמטר פורמלי** ללא שינוי שלו, האם ניתן לתת לו ערך חדש או האם ניתן להעריכו ולשנותו.

15.06.21 **מַקְרוֹ** **macro**

צרף-שפה המוחלף בסדרת משפטים מוגדרת מראש.

15.06.22 **הוֹרָאַת מַקְרוֹ** **macroinstruction**

הוראה המפעילה **הגדרת מאקרו** ברמת שפת התכנות הקוראת.

15.06.23 **קְרִיאָה לְמַקְרוֹ** **macrocall**

משפט המפעיל **הגדרת מאקרו** ברמת שפת התכנות הקוראת.

15.06.24 **הַגְדָּרַת מַקְרוֹ** **macrodefinition**

מעקובת מוגדרת מראש של פקודות, של **משפטים** או של הוראות, המחליפה כל **הוראת מאקרו** או **קריאה למאקרו** מתאימות המתייחסות אליה.

15.06.25 **חֲבִילָה** **package**

בשפות תכנות - **פִּרְקָן** שנתכן (designed) לספק הפשטה, **כמיסה** או **הסתרת מידע** באמצעות ההקבצה של **צרפי-שפה** המקושרים ביניהם לוגית, כגון **טיפוסי נתונים**, **עצמי נתונים** באותם **טיפוסי נתונים**, ו**תת-תוכניות** עם **פרמטרים** מאותם טיפוסי נתונים.

15.06.26 **הַכְרָזַת חֲבִילָה** **package declaration**

ה**הכרזה** הנפרדת של אותם **צרפי-שפה** שהמפרטים שלהם נדרשים מחוץ ל**חבילה**, למטרות מישוק או הידור.

15.06.27 **חֵלֶק נִרְאֶה** **visible part**

החלק של **הכרזת חבילה** המספק את הפרטים הדרושים למשתמשים באובייקטים או בשירותים של ה**חבילה**.

15.06.28 **חֵלֶק פְּרָטִי** **private part**

החלק של **הכרזת חבילה** המספק את הפרטים המבניים הנחוצים לתהליך הפיתוח אך אינם רלוונטיים ואינם נגישים למשתמשי ה**חבילה**.

15.06.29 **יְסוֹד** [שם תואר] **generic**
גֶּנֶרִי

המשמש תבנית (template) ליצירת צרף-שפה ממשי ב**טיפוסי הנתונים** הישימים תוך עמידה בדרישות כללי **ההטפסה החזקה**.

15.06.30 **הַכְרָזַת יְסוֹד** **generic declaration**
הַכְרָזָה גֶּנֶרִית

ה**הכרזה** של **צֶרף-שפה** גנרי המציג את ה**פרמטרים** ה**גנריים** המיועדים להחלפה ב**פרמטרים ממשיים** במהלך בניית **אדגום גנרי**.

15.06.31 **גּוּף יְסוֹד** **generic body**
גּוּף גֶּנֶרִי

ה**גוף** של **צֶרף-שפה גנרי** המשמש תבנית (template) לגופים של צרפי-שפה ממשיים בשפת התכנות במהלך בניית **אדגום גנרי**.

15.06.32 **פְּעֻלַּת יְסוֹד** **generic operation**
פְּעֻלָּה גֶּנֶרִית

פעולה עמוסת-משמעויות (overloaded) שאינה מצביעה על פעולה מסוימת אחת אלא מספקת **פרמטרים פורמליים** ל**פרמטרים ממשיים** של **טיפוסי נתונים** מסוימים.

הערה: ראו "עמיסת משמעויות".

15.06.33 **חֲבִילַת יְסוֹד** **generic package**

חֲבִילָה גֶנֶרִית

חבילה שנתכנה לשמש תבנית (template) לאלגוריתמים קשורים או לפעולות קשורות.

דוגמות: חבילות גנריות של פונקציות טריגונומטריות, פעולות מחסנית, פונקציות פיננסיות וכדומה.

15.06.34 **פִּרְקַן יְסוֹד** **generic module**

פִּרְקָן גֶנֶרִי

תבנית פרמטרית ליצירת **פִּרְקָנִים** באמצעות **אדגום גנרי**.

הערה: הפרמטרים של התבנית הם בעלי אופי גנרי ואין להתייחס אליהם כאל פרמטרים פורמליים של הפרקנים הנגזרים.

15.06.35 **אִדְגּוּם יְסוֹד** **generic instantiation**

אִדְגּוּם גֶנֶרִי

תהליך הפתרון של **פרמטרים גנריים** של **פִּרְקָן גנרי** כדי ליצור פרקן מוחשי.

15.06.36 **אֶדְגַּם יְסוֹד** **generic instance**

אֶדְגָּם גֶנֶרִי

פִּרְקָן מוחשי הנגזר מתוך **פִּרְקָן גנרי** בתהליך יצירת **אִדגום גנרי**.

15.06.37 **נֹהַל כְּלָלִי** **general procedure**

קבוצת פקודות שאפשר לקרוא לה מכמה מקומות בתוכנית.

15.06.38 **נֹהַל מֻגְדָּר-מִשְׁתַּמֵּשׁ** **user-defined procedure**

קבוצת פקודות שהמשתמש מגדיר.

15.06.39 **תַּת-נֹהַל** **subprocedure**

קבוצת פקודות המבצעת משימה מוגדרת.

15.06.40 **גּוּשׁ בְּסִיסִי** **basic block**

בשפות תכנות - מעקובת הוראות בעלת נקודת כניסה יחידה, נקודת יציאה יחידה וללא הסתעפויות פנימיות.

15.06.41 **קוֹד תָּכְנִית** **program code**

אוסף הוראות היוצרות תוכנית.

15.06.42 **תִּעוּד פְּנִימִי** **internal documentation**

הערות ותזכורות הנכללות בקוד המקור של תוכנית.

15.06.43 **יְחִידַת תִּרְגּוּם** **translation unit**

יחידת קוד מקור בסיסית המוגשת למהדר לתִרגום והכוללת את קובץ המקור ואת כל הקבצים הנלווים הנדרשים להידור.

15.06.44 **יְחִידַת קִשּׁוּר** **link unit**

ישות יחידה המוגשת לַמְקַשֵּׁר.

הערה: יחידת קישור יכולה להיות קובץ יעד, ספריית ארכיון או ספרייה משותפת.

15.06.45 **סֵמֶל חִיצוֹנִי** **external symbol**

שם של פונקצייה או פריט נתונים בקובץ יעד הזמינים לקישור מקובצי יעד אחרים.

15.06.46 **פִּרְקַן קוֹד** **code module**

חלק של תוכנית יישום הכולל קוד מוגדר הנדרש לביצוע התוכנית.

15.06.47 **לוּלָאָה נִבְדֶּקֶת רֹאשׁ** **top-driven loops**

לולאה שבדיקת התנאי שלה מבוצעת בתחילת העיבוד שלה.

15.06.48 **לוּלָאָה נִבְדֶּקֶת זָנָב** **bottom-driven loops**

לולאה שבדיקת התנאי שלה מבוצעת בסוף העיבוד שלה.

15.06.49 **לוּלָאָה מְקוֹנֶנֶת** **nested loops**

לולאה בתוך לולאה אחרת.

15.06.50 **תַּרְמִיל** **bean**

רכיב תוכנה רב-שימושי הכתוב בשפת JAVA .

הערה:

אפשר להרכיב יישום מכמה תרמילים.

15.06.51 **מְכָל** **container**

ישות המספקת לרכיבים שבתוכה שירותי ניהול מחזור חיים, אבטחה, פריסה וזמן-ריצה וכן שירותים מיוחדים לפי טיפוס המכל.

15.06.52 **תֵּבַת חוֹל** **sandbox**

מספר רכיבי מערכת המשתפים פעולה כדי לאבטח את משאבי המערכת כנגד ניסיון גישה לא-מורשית או זדונית.

15.06.53 **מַחְבֵּר** **connector**

מנגנון הרחבה תקני המספק קישוריות בין מכלים ויישומים.

15.06.54 **חוֹזֵה רְכִיב** **component contract**

החוזה בין רכיב לבין **המכל** שלו.

הערה:

החוזה כולל: ניהול מחזור החיים של הרכיב, מנשק הקשר המשמש את האדגם לשם השגת נתונים ושירותים שונים מהמכל שלו ורשימת שירותים שכל מכל חייב לספק לרכיביו.

15.06.55 **תֶּפֶס בַּיִת** **home handle**

עצם שניתן להשתמש בו כדי להשיג פנייה אל מִנְשַׁק הבית.

15.06.56 **פָּרָמֶטֶר אִתְחוּל** **initialization parameter**

פרמטר ראשוני להפעלת תוכנית.

15.06.57 **תֹּכֶן בָּצִיעַ** **executable content**

יישום המורץ מתוך קובץ סימני עריכה.

דוגמה:

הרצת יישומון (applet) מקובץ HTML.

15.06.58 **מִטְמָן** **Easter egg**

מנגנון סמוי לא-מזיק בתוכנית המשופעל לאחר שמשתמש מבצע מעקובת סתומה של פעולות.

15.06.59 **תַּבְנִית קוֹד** **code template**

קוד מוגדר מראש בתיבת עצם.

15.06.60 **טֹפֶס** **form**

מצג מובנה המשמש במנשק משתמש.

15.06.61 **פִּרְקַן טֹפֶס** **form module**

קובץ המכיל קוד מקור של תוכנית טופס.

15.06.62 **מִשְׁתַּנֶּה בְּרָמַת טֹפֶס** **form-level variable**

משתנה המשמש בטופס בלבד.

15.06.63 **טֹפֶס חוֹבָה** **modal form**

מצג טופס הקבוע במקומו על המרקע ודורש פעולת משתמש לשם המשך העבודה.

15.06.64 **טֹפֶס רִשּׁוּת** **modeless form**

מצג טופס בלתי מחייב שאינו דורש פעולת משתמש לשם המשך העבודה.

15.06.65 **טֹפֶס הֶזְנֵק** **startup form**

המצג הראשון הנטען ומוצג בעת הפעלת יישום.

15.06.66 **הַרְחָבָה** **extension**

תוכנית מחשב שאינה מתפקדת כשלעצמה, אלא נתכנה כדי שתשולב בתוכנה אחרת לשם שיפורה או הרחבת תפקודיה.

15.06.67 **טֹפֶס עֶלְיוֹן** **toplevel form**

הטופס הנמצא ברמה הגבוהה ביותר במדרג טפסים.

15.06.68 **תְּחוּם עֶלְיוֹן** **toplevel scope**

התחום המקיף את כל הפעילות של תוכנית.

15.06.69 **נֹהַל אֶסִינְכְּרוֹנִי** **asynchronous procedure**

נוהל המתבצע ללא תיאום בזמן עם התוכנית הקוראת.

15.06.70 **מֶרְחַב שֵׁמוֹת** **namespace**

מתייחס לקבוצת שמות של אלמנטים או מאפיינים הקשורים ביניהם.

15.06.71 **יַעַר** **forest**

קבוצה מסודרת של תת-עצים של צמתים בשפת סימני עריכה.

דוגמה:

שפת סימני עריכה XML.

15.07 - משימות - Tasks

15.07.01 **תָּכְנִית רָאשִׁית** **main program**

בשפות תכנות - **פִּרקָן** של תוכנית, המבוצע ראשון והיכול גם להפעיל פרקנים אחרים.

15.07.02 **מְשִׁימָה** **task**

בשפות תכנות - **פִּרקָן** הניתן לביצוע בו-זמנית עם פִּרקָנים (modules) אחרים במחשב רב-מעבדים או בסירוג במעבד אחד.

15.07.03 **קֶטַע קְרִיטִי** **critical section**

חלק של **משימה** שבמהלך ביצועו אסור הביצוע של משימות אחרות.

15.07.04 **סִנְכְּרוּן מְשִׁימוֹת** **task synchronization**

תיאום בזמן הפעילות של **משימות** באמצעים שונים כגון **סמפור**, **מוניטור** או **מפגש**, כך שבכל פעם יבוצע רק תהליך אחד מתוך כמה תהליכים מקבילים.

15.07.05 **מִפְגָּשׁ** **rendezvous**

בשפות תכנות - ההידוד (interaction) בין שתי **משימות** בנקודה מסוימת בכל ביצוע של המשימה, שבה עיבוד אחד עשוי להמתין לעיבוד אחר.

15.07.06 **סֵמָפוֹר** **semaphore**

בשפות תכנות - **מבנה נתונים** משותף לבקרה, באמצעות תור, של הגישה למשאבים הזמינים לכמה **משימות**, משימה אחת בכל פעם.

15.07.07 **מוֹנִיטוֹר**, **מַשְׁגוֹחַ** **monitor**

בשפות תכנות - **עצם נתונים** משותף וקבוצת פעולות העשויות לטפלל אותו, המיועדים לבקרה על גישות למשאבים, אחד בכל פעם.

15.07.08 **עֲרִיכָה** **editing**

בשפות תכנות - התמרת ערכים לייצוג המתאים לתסדיר נתון.

15.08 - ביצוע - Execution

15.08.01 **סֵדֶר בִּצּוּעַ** **execution sequence**

הסדר שבו מבוצעים הכרזות משפטים וחלקי משפטים.

15.08.02 **זְרִימַת בַּקָּרָה** **control flow**

סדר הביצוע בכל אחד מהנתיבים האפשריים בתוכנית.

הערה:

את זרימת הבקרה ניתן לייצג על ידי תרשים זרימת בקרה.

15.08.03 **תּוֹצָא לְוָאי** **side effect**

כל תוצאה עקיפה הנגרמת מביצוע של **ביטוי**, **משפט** או **תת-תוכנית**.

הערה:

תוצא לוואי עשוי לשמש, למשל, לשינוי ערך הנתונים של פרמטר שהועבר על ידי פונקצייה.

15.08.04 **מִטּוּב בְּסִיסִי** **basic optimizations**

מיטוב שאינו משנה בדרך כלל את ההתנהגות של קוד מקור מובנה.

הערה:

המיטוב הבסיסי הוא תת-קבוצה של מיטוב תַּקִּיף.

15.08.05 **מִטּוּב תַּקִּיף** **aggressive optimization**

מיטוב היכול לשנות את ההתנהגות של קוד מובנה.

הערה:

מיטוב תקיף הוא קבוצת-על של מיטובים בסיסיים.

15.08.06 **מִטּוּב מְבֻסַּס-מִתְאָר** **profile-based optimization**

מיטוב שבו המהדר והמקשר עובדים יחד למיטוב של יישום על בסיס מתאר הנתונים המושג תוך כדי הרצת היישום על נתוני קלט טיפוסיים.

15.08.07 **הַצְנָרַת תָּכְנָה** **software pipelining**

התמרת קוד מקור הממטבת ביצוע של לולאות בתוכנית, המכילות פעולות אריתמטיות.

הערה:

שימושיות במיוחד בלולאות המבצעות פעולות אריתמטיות על מספרים בייצוג בנקודה צפה בדַיִק יחיד או בדַיִק כפול.

15.08.08 **יִחוּד שֵׁמוֹת** **name mangling**

תהליך חילול של מזהים פנימיים חד-משמעיים ממזהים בקוד המקור כדי לפתור בעיות הקשורות לתחום המשתנים ו**עמיסת-משמעויות**.

הערה:

ראו "הסרת ייחוד שמות".

15.08.09 **הֲסָרַת יִחוּד שֵׁמוֹת** **name demangling**

תהליך שינוי הייצוג הפנימי של מזהים חזרה למזהים שבקוד המקור.

הערה:

ראו "ייחוד שמות".

15.08.10 **הַשְׁוָאָה בִּינָרִית** **binary comparison**

השוואת תווים לפי הייצוג הבינרי הפנימי שלהם.

15.08.11 **פִּתְרוֹן-מַעַן** **address resolving**

בקריאה לפונקצייה, תהליך ההמרה של שם לוגי של עצם במען הפיזי שלו.

15.08.12 **טַפְלָל** **manipulator**

פונקציית קלט-פלט המתירה ביצוע של פעולות קלט-פלט מיוחדות המאפשרות חילוץ מזרם הקלט או תחיבה לזרם הפלט.

15.08.13 **הוֹסָפָה** **incrementing**

חיבור ערך לצובר.

15.08.14 **הַפְחָתָה** **decrementing**

הפחתת ערך מצובר.

15.08.15 **הַעֲבָרָה** **pass** [n]

העברת נתונים מנוהל קורא לנוהל נקרא.

15.08.16 **פְּעֻלָּה לֹא-פְּרִיקָה** **atomic operation**

פעולה שאינה יכולה להתבצע באופן חלקי ולכן מתבצעת בשלמותה או בכלל לא.

15.08.17 **קִבּוּעַ** **commit**

בעיבוד תנועות, הפעולה המקבעת את כל העדכונים של משאבים המעורבים בתנועה.

הערה:

ראו "הסגה" - rollback.

15.08.18 **הֲסָגָה** **rollback**

בעיבוד תנועות, הפעולה המבטלת את כל העדכונים של משאבים המעורבים בתנועה ומחזירה אותם למצבם הקודם.

הערה:

ראו "קיבוע" – commit.

15.08.19 **רְמַת בִּדּוּד תְּנוּעָה** **transaction isolation level**

המידה שבה מצבי ביניים של נתונים המעודכנים על ידי תנועה, נראים לתוכניות בו-זמניות אחרות.

הערה:

ככל שרמת הבידוד עולה, הנִראוּת לתוכניות בו-זמניות אחרות נמוכה יותר.

15.08.20 **אֶטוֹמִיּוּת, עֶקִיבוּת, בִּדּוּד וְקַיָּמוּת (עֲמִידוּת)** **Atomicity, Consistency, Isolation, and Durability**

ACID [abb]

ארבע תכונות הנדרשות להגדרת התנועה.

15.08.21 **הַשְׁמָשָׁה** **garbage collection**

תהליך אוטומטי לגילוי ושחרור של קטעי זיכרון תפוסים שאינם בשימוש, לצורך שימוש חוזר.

15.08.22 **זֶרֶם** **stream**

זרם בתים של נתונים המועברים משולח למקבל.

15.08.23 **שְׁגִיאָה לוֹגִית** **logic error**

שגיאה הנגרמת ממשפטים נכונים תחבירית המניבים תוצאות לא רצויות או לא צפויות.

15.08.24 **טָעוּת עִגּוּל** **roundoff error**

טעות חישוב המתרחשת כשאין די סיביות להחזקת המנטיסה של מספר.

15.08.25 **שִׁיטוֹת מַעֲנֶה** **callback methods**

שיטות שמומשו בתוך רכיב תוכנית והמשמשות למענֶה לקריאות של המערכת או של המכל, הנוגעות לאירועים במחזור החיים של הרכיב.

15.08.26 **אִלּוּץ נְתוּנֵי מִשְׁתַּמֵּשׁ** **user data constraint**

מגבלה המציינת כיצד להגן על הנתונים במעבר בין לקוח ובין מכל, אם באמצעות מניעת הטיפול בנתונים ואם בחסימת הציתות להם.

15.08.27 **בַּקָּרָה עַל-יְדֵי קוֹצֵב זְמַן** **timer control**

בקרת תוכנית המבוצעת על ידי רכיב בתוכנית המודיע לַיישום על פרקי זמן קצובים או על שעות מוגדרות.

15.08.28 **הַעֲרָכָה** **evaluation**

חישוב של ביטוי, פונקצייה או טופס.

15.08.29 **כְּרִיכָה מְקֻבַּעַת** **immutable binding**

כריכה שבה אסור לשנות את היוחסה (relation) בין המזהה לבין העצם שהוא מייצג.

15.09 - תכנות כְּוון-עצמים - Object-Oriented Programming

15.09.01 **הַסְתָּרַת מֵידָע** **information hiding**

מניעת גישה ל**צֶרֶף-שפה**, או מניעת הידיעה עליו או על פרטים מסוימים שלו, למעט הפרטים שחיוני למשתמש לדעת.

הערה:

התוצאה – הֶסתר מידע.

15.09.02 **כָּמַס** **encapsulate** [v]

הֵחיל **הסתרת מידע** על **צֶרֶף-שפה**.

15.09.03 **כְּמִיסָה** **encapsulation**

התהליך או התוצאה של **הסתרת מידע**.

15.09.04 **פְּרָטִי** **private**

הַנָּגִיש רק לפונקציות חברות ולידידים של אותה מחלקה.

הערות:

1. בתכנות כוון-עצמים - מתייחס לחבר במחלקה.
2. ראו "מוגן", "ציבורי".

15.09.05 **צִבּוּרִי** **public**

הַנָּגִיש לכל הפונקציות.

הערות:

1. בתכנות כוון-עצמים - מתייחס לחבר במחלקה.
2. ראו "פרטי", "מוגן".

15.09.06 **מוּגָן** **protected**

הַנָּגִיש רק לפונקציות חברות ולידידים של אותה מחלקה, או לפונקציות חברות ולידידים שנגזרו מאותה מחלקה.

הערות:

1. בתכנות כוון-עצמים - מתייחס לחבר במחלקה.
2. ראו "פרטי", "ציבורי".

15.09.07 **עֶצֶם** **object**

בתכנות כוון עצמים - ישות המורכבת מקבוצת פעולות ומנתונים.

15.09.08 **מֶסֶר** **message**

בשפות תכנות - פנייה ל**עצם** שיבצע אחת מפעולותיו.

15.09.09 **פְּרוֹטוֹקוֹל** **protocol**

בשפות תכנות - קבוצת כללים הקובעת את התנהגות ה**עצמים** בעת חילופי **מסרים**.

15.09.10 **שִׁיטָה** **method**

בשפות תכנות - פעולה ש**עצם** מבצע עם קבלת **מסר**.

15.09.11 **מַחְלָקָה** **class**

בשפות תכנות - תבנית של **עצמים** המגדירה את המבנה הפנימי ואת קבוצת הפעולות בכל אֶדגם (instance) שלהם.

הערה: בתכנות כוון עצמים, מחלקות שקולות לטיפוסי נתונים בשפות תכנות פרוצדורליות כגון שפת C ושפת Pascal.

15.09.12 **רַב-צוּרָתִיּוּת** **polymorphism**

היכולת של **עצמים** שונים להגיב בצורה שונה לאותו **מסר**.

15.09.13 **יְרִישָׁה** **inheritance**

העתקת המבנה הפנימי וקבוצת הפעולות בשלמותם או בחלקם, מ**מחלקה** אחת ל**מחלקה** שמתחתיה.

15.09.14 **אֲצִילָה** **delegation**

העתקת המבנה הפנימי וקבוצת הפעולות בשלמותם או בחלקם, מ**עצם** אחד לעצם אחר.

15.09.15 **כִּוּון-עֲצָמִים** **object-oriented**

התומך ב**עצמים** וב**מחלקות** והמאפשר **ירישה** ו**אצילה**.

הערה:

מתייחס לטכניקה או לשפת תכנות.

15.09.16 **אַרְגּוּמֶנְט תַּבְנִית** **template argument**

template

בתכנות כוון-עצמים, שלד או תיאור המשמשים לבניית קבוצה אינסופית של ישויות, עצמים, מחלקות או פונקציות.

15.09.17 **תַּבְנִית מַחְלָקָה** **class template**

תבנית המגדירה קבוצה לא חסומה של מחלקות קרובות לוגית.

15.09.18 **מַחְלֶקֶת בָּסִיס** **base class**

מחלקה שממנה נגזרות מחלקות אחרות על ידי ירישה, וגם היא עצמה יכולה להיגזר ממחלקת בסיס אחרת.

15.09.19 **מַחְלֶקֶת בָּסִיס שׁ׳רֶשׁ** **root base class**

מחלקת בסיס המשמשת שורש במדרג ירישה של מחלקות.

15.09.20 **מַחְלָקָה מְאֻדְגֶּמֶת** **instantiated class**

מחלקה שנוצרה באדגום מתוך **תבנית מחלקה**.

15.09.21 **מַחְלֶקֶת לִבָּה** **core class**

מחלקה או מנשק ציבוריים שהם חברים מקובלים בתשתית המאפשרת לתוכנית מחשב להתבצע בכל מערכת הפעלה שבה משולבת אותה תשתית.

15.09.22 **מֶטָא-מַחְלָקָה** **metaclass**

מחלקה שהאדגמים שלה הם עצמם מחלקות.

15.09.23 **מַחְלָקָה מֻפְשֶׁטֶת** **abstract class**

abstract superclass

מחלקה המוגדרת כמבנה-על שאין יוצרים ממנה אדגמים ישירים אלא רק תת-מחלקות שמהן בונים אדגמים.

15.09.24 **מחלקת-על** **superclass**

מחלקה ברמה גבוהה שמורישה שיטות ומאפיינים (נתונים ועיבוד) במורד מִדרַג המחלקות לתת-מחלקות.

הערה:

מחלקות על מופשטות משמשות כמבנה-על ואינן משמשות לאִדגום עצמים.

15.09.25 **תַּת-מַחְלָקָה** **subclass**

מחלקה במורד מִדרַג המחלקות שיורשת שיטות ומאפיינים (נתונים ועיבוד) מ**מחלקת-על** שמעליה ומשתמשת בהם לאִדגוּם עצמים.

הערה:

תת-מחלקה עשויה להכיל שיטות ומאפיינים שלא התקבלו בירישה ממחלקת-על.

15.09.26 **נָתוּן חָבֵר** **data member**

משתנה מסומך למחלקה הנגיש לשיטות ולאדגמים של המחלקה.

15.09.27 **תַּבְנִית פּוּנְקְצִיּוֹת** **function template**

תבנית המגדירה קבוצה לא חסומה של פונקציות קרובות לוגית.

15.09.28 **פּוּנְקְצִיַּת תַּבְנִית** **template function**

טיפוס או קבוע המוגדרים לתבנית כדי לייעד את התבנית לשימוש מיוחד.

15.09.29 **פּוּנְקְצִיַּת גִישָׁה** **access functions**

פונקצייה ציבורית בהגדרת מחלקה שייעודה לספק גישה לנתונים פרטיים של המחלקה.

הערות:

1. רק פונקציות חברות ופונקציות ידידות רשאיות לשנות ערכים של משתנים פרטיים.
2. גישה יכולה להיות: ציבורית, פרטית או מוגנת.

15.09.30 **פּוּנְקְצִיָּה חֲבֵרָה** **member function**

כל פונקצייה שהוכרזה חלק של מחלקה.

15.09.31 **פּוּנְקְצִיָּה יְדִידָה** **friend function**

פונקצייה שאינה חברה במחלקה אך יש לה גישה לנתונים חברים ולפונקציות חברות – הפרטיים והמוגנים של המחלקה.

15.09.32 **פּוּנְקְצִיָּה תְּחוּבָה** **inline function**

פונקצייה שהקוד שלה מועתק לכל מקום שהיא נקראת ממנו.

15.09.33 **פּוּנְקְצִיָּה מְאֻדְגֶּמֶת** **instantiated function**

פונקצייה שנוצרה באדגום מתוך תבנית פונקצייה.

15.09.34 **פּוּנְקְצִיַּת יְסוֹד** **generic function**

פונקצייה שהתנהגותה נקבעת על פי טיפוסי הפרמטרים הממשיים שלה והכוללת בדרך כלל כמה שיטות.

15.09.35 **גִישָׁה פְּרָטִית** **private access**

גישה מבחוץ רק אל נתונים חברים או אל פונקציות חברות של אותה מחלקה, או אל פונקציות חברות שסומנו כ"ידידים" של המחלקה.

15.09.36 **גִישָׁה מוּגָנֶת** **protected access**

טכניקת המאפשרת למחלקות נגזרות (תת-מחלקות) בלבד לגשת לשדה מוגן כאילו הוגדר כשדה פרטי שלהן.

15.09.37 **גִישָׁה צִבּוּרִית** **public access**

טכניקה המאפשרת גישה מלאה מבחוץ, אל נתונים חברים או אל פונקציות חברות של מחלקה, והמוגבלת על ידי התחום בלבד.

15.09.38 **חָבֵר מוּגָן** **protected member**

נתון חבר או פונקצייה חברה הנגישים רק למחלקה שהגדירה אותם, או לכל מחלקה

הנגזרת מאותה מחלקה ולכל הידידים של המחלקה שבה הוגדרו.

15.09.39 **חָבֵר צִבּוּרִי** **public member**

נתון חבר או פונקצייה חברה הנגישים לכל רכיב תוכנית המשתמש במחלקה.

15.09.40 **אִדְגּוּם אוֹטוֹמָטִי** **automatic instantiation**

אדגום של תבנית המבוצע אוטומטית, בנקודת השימוש בו.

15.09.41 **אִדְגּוּם מְפֹרָשׁ** **explicit instantiation**

אדגום של תבנית בהוראה מפורשת, בנקודת השימוש בו.

15.09.42 **אִדְגּוּם מֻנְחֶה** **directed instantiation**

אדגום של תבנית על פי הנחיות המְפַתח.

15.09.43 **אֶדְגָּם יָשִׁיר** **direct instance**

אדגם שאודגם ישירות ממחלקה ולא מתת-מחלקה שלה.

15.09.44 **סְגִירָה** **closing**

בתכנות כוון-עצמים - תהליך השלמת המילוי של כל האדגמים של תבנית בקבוצת יחידות קישור.

15.09.45 **רְשִׁימַת אִתְחוּל חֲבֵרִים** **member initialization list**

רשימת נתוני חבר המשמשת את ה**בּוֹנֶה** לאתחול נכון של עצם במחלקה.

15.09.46 **אִתְחוּל חָבֵר מֵחָבֵר** **member-wise initialization**

צורת אתחול מיוחדת המשמשת להעתקת רכיבים, זה אחר זה, מעצם מקור לעצם יעד.

15.09.47 **בּוֹנֶה** **constructor**

בתכנות כוון-עצמים, פונקצייה המשמשת לאתחול עצמים מטיפוס המחלקה שלה.

15.09.48 **הוֹרֵס** **destructor**

בתכנות כוון-עצמים, פונקצייה חברה מיוחדת במחלקה המשמשת לחיסול של עצמים מטיפוס המחלקה שלה.

15.09.49 **עוֹטְפָן** **wrapper**

עצם (רכיב תוכנה) המכיל בתוכו (כומס, עוטף) עצמים (נתונים או רכיבי תוכנה) אחרים ומספק להם מנשק מתאים לעצמים אחרים הפונים אל העצם הכמוס.

הערה:

העוטפן משמש לעיתים קרובות, להגנה על תוכנה מדור קודם שאמורה לתפקד בסביבה חדשה.

15.09.50 **שִׁיטַת מַחְלָקָה** **class method**

שיטה שאפשר להפעיל על כל עצם שאודגם במחלקה.

15.09.51 **הֶתֵּר שִׁיטָה** **method permission**

היתר להפעלת קבוצה מוגדרת של שיטות מחלקה.

15.09.52 **שִׁיטַת הֲסָרָה** **remove method**

שיטה המופעלת למחיקת עצם.

15.09.53 **יְרִישָׁה מֵרַבִּים** **multiple inheritance**

ירישה מכמה מחלקות.

הערה: ראו "ירישה מיחיד".

15.09.54 **יְרִישָׁה מִיָּחִיד** **single inheritance**

ירישה ממחלקה אחת בלבד.

הערה: ראו "ירישה מרבים".

15.10 - תכונות ואופיינים - Features and characteristics

15.10.01 **צִיּוּן** **Subscript, index**

התייחסות לאלמנט ב**מערך** באמצעות הפניה למערך בצירוף **ביטוי** אחד או יותר שלאחר הערכתם מצביעים על מקומו של האלמנט.

15.10.02 **הַפְנָיָה עֲקִיפָה** **indirect referencing**

מנגנון המאפשר הפניה באמצעות **עצם נתונים**, המצביע על **צֶרֶף-השפה** המבוקש.

הערה:

ההפניה יכולה להתבצע לאורך שרשרת של עצמי נתונים, שבה כל עצם נתונים, פרט לאחרון, מצביע על הבא אחריו, ועצם הנתונים האחרון מצביע על צרף-השפה המבוקש.

15.10.03 **אִתְחֵל** **initialize** [v]

נתן ערך תחילי ל**עצם נתונים** בראשית תפקודו.

15.10.04 **הַקְצָאַת זִכָּרוֹן דִּינָמִית** **dynamic storage allocation**

dynamic memory allocation

הקצאת מרחב זיכרון ל**עצמי נתונים** רק למשך הביצוע של ה**תחום** שלהם.

15.10.05 **מִתְרַחֲבוּת** **extensibility**

בשפות תכנות - היכולת של שפה לאפשר הגדרה של **צרפי-שפה** ולאפשר את השימוש בהם לפי התחביר של צרפי-השפה התקניים.

15.10.06 **הַפְנָיָה** **reference**

צרף-שפה, כגון **מזהה**, המציין צרף-שפה מוכרז.

15.10.07 **הַפְנָיָה אֲחִידָה** **uniform referencing**

תכונת שפת תכנות, כשכמה **צרפי-שפה** הם שווי-צורה מבחינת ההפניה.

דוגמות:

צרפי-שפה זהים לשיוך שם ולהפניה עקיפה; צרפי-שפה זהים לציון פרמטרים ממשיים.

15.10.08 **מְאַפְיֵן הֶקְשֵׁר** **context attribute**

מאפיין המגדיר סביבת עבודה של שרתית.

15.10.09 **מִתְאַר פְּרִיסָה** **deployment descriptor**

קובץ מפרט המסופק יחד עם כל מודול וישום והמתאר את אופן הפריסה שלהם.

15.10.10 **תְּכוּנַת אִפְשׁוּר** **enabling property**

תכונה שהערך שלה קובע אם עצם יגיב לאירועים שיוזם המשתמש.

15.10.11 **מְאֻפְשָׁר** **enabled**

ש**תכונת האפשור** שלו קובעת שהוא יגיב לאירועים שיוזם המשתמש.

הערה:

מתייחס לעצם.

15.10.12 **לֹא-מְאֻפְשָׁר** **disabled**

שתכונת האפשור שלו קובעת שהוא לא יגיב לאירועים שיוזם המשתמש.

הערה:

מתייחס לעצם.

15.11 - תכנות מונַע-אירועים - Event driven programming

15.11.01 **אֵרוּעַ** **event**

1. כל פעולת משתמש או מערכת המעוררת תגובה של יישום.
מפרט לציון שינויים בתכונות או במצב של עצם.

15.11.02 **נֹהַל אֵרוּעַ** **event procedure**

פקודה המנחה עצם כיצד להגיב על אירוע.

15.11.03 **מונַע אֵרוּעִים** **event-driven**

המגיב לקלט ממשתמשים או למסרים מיישומים אחרים.

הערות:

1. דוגמות לקלט ממשתמשים הן תנועות עכבר, הקשות מקלדת, בחירת אפשרויות מתפריט.
2. עיבוד מונע אירועים שונה מעיבוד באצווה או עיבוד סדרתי.
3. מופיע בצירופים כגון:

יישום מונע אירועים	Event driven application
תוכנית מונעת אירועים	Event driven program
תכנות מונע אירועים	Event driven programming
מוניטור מונע אירועים	Event driven monitor

15.11.04 **תִּכְנוּת מונַע אֵרוּעִים** **event-driven programming**

תכנות מחשבים שבו זרימת התוכנית נקבעת על סמך פעולות משתמש או מסרים מתוכניות אחרות.

הערות:

1. דוגמות לפעולות משתמש הן נקישות עכבר, הקשות מקלדת.
2. שונה מתכנות של עיבוד באצווה ועיבוד ספרתי.

15.12 - יישומים - Applications

15.12.01 **סִפְרִיַּת אַרְכִּיוֹן** **archive library**

אוסף קובצי יעד.

15.12.02 **סִפְרִיָּה מְשֻׁתָּפֶת** **shared library**

אוסף קובצי יעד הכוללים קוד בלתי תלוי במיקומו בזיכרון, המשמשים קלט לַמְקשֵׁר.

15.12.03 **קְבָצִים קְרִיאִים לַכֹּל** **world readable files**

קבצים במערכת קבצים הניתנים לקריאה על ידי כל משתמש.

15.12.04 **קֹבֶץ מָקוֹר** **source file**

קובץ המכיל תוכנית בשפת מקור.

15.12.05 **קֹבֶץ כּוֹתֶרֶת** **header file**

קובץ המכיל הכרזות מחלקה, הנחיות לעיבוד קדם וכדומה, והנכלל ביחידת תרגום.

15.12.06 **קֹבֶץ נְתוּנִים** **data file**

קובץ המכיל נתונים.

15.12.07 **קֹבֶץ בָּצִיעַ** **executable file**

קובץ הניתן לביצוע במערכת הפעלה נתונה.

15.12.08 **קֹבֶץ תָּכְנִית** **program file**

קובץ הכולל הוראות ליצירה ובקרה של תוכנית.

15.12.09 **קֹבֶץ סִדְרָתִי** **sequential file**

קובץ שבו הרשומות מעובדות לפי סדר האחסון שלהן.

15.12.10 **חוֹתֶמֶת זְמַן** **timestamp**

רשומה המכילה את התאריך והשעה של אירוע מוגדר.

15.12.11 **יִשּׁוּם מְבֻזָּר** **distributed application**

יישום המורכב מרכיבים מספר המורצים בסביבות הרצה אחדות, בכמה מחשבים המקושרים ברשת תקשורת.

הערה:

יישום מבוזר טיפוסי יכול להיות דו-שכבתי (לקוח/שרת), תלת-שכבתי (לקוח/מערכת ביניים/שרת) ורב-שכבתי (לקוח/מערכות ביניים רבות/שרתים רבים).

15.12.12 **מְנַהֵל מַשְׁאַבִּים** **resource manager**

תוכנית המתאמת גישה לקבוצת משאבים משותפים.

15.12.13 **אִבְטוּל** **passivation**

העברת תוכנית למצב לא פעיל.

הערה:

מנוגד לשפעול.

15.12.14 **עִבּוּד סִדְרָתִי** **sequential processing**

עיבוד פקודות זו אחר זו על פי סדר.

15.12.15 **זְמַן הַפְסָקָה** **break time**

פרק זמן שבו ההרצה מופסקת.

15.12.16 **הַשְׁוָאַת תַּמְלִיל** **textual comparison**

השוואת תווים ללא הבחנה באותיות רישיות.

15.12.17 **מַעֲרֶכֶת זְמַן-רִיצָה** **runtime system**

סביבת התוכנה הדרושה להרצת תוכנית.

15.12.18 **אוּנִיקוֹד** **Unicode**

קבוצת תווים שבה לכל תו מוקצבות 16 סיביות.

הערה:

קבוצת התווים הוגדרה בתקן ISO 10646.

15.12.19 **לָקוֹחַ יִשּׁוּם** **application client**

בסביבת שרת-לקוח - תוכנית עצמאית, בלתי תלויה בשרת, המתבצעת בלקוח.

15.12.20 **יֵשׁוּת עָמִית** **peer entity**

עָמִית **peer**

ישויות באותה מערכת פתוחה או במערכות פתוחות נפרדות הנמצאות באותה שכבה.

הערות:

1. התקשורת בין ישויות עמית הנמצאות באותה מערכת פתוחה אינה כלולה בתחום ההגדרה של מודל OSI.
2. ההגדרה וההערה שלעיל מובאים מת"י 1080 חלק 26.

15.12.21 **פְּרוֹטוֹקוֹל שִׁכְבַת שְׁקָעִים מְאֻבְטָחַת** **Secure Socket Layer protocol**

פְּרוֹטוֹקוֹל SSL **SSL protocol**

פרוטוקול המאפשר הצפנה של התקשורת בין לקוח ושרת למטרות הגנה על פרטיות.

15.12.22 **שָׂרְתִית** **servlet**

תוכנית המרחיבה את התפקודיות של שרת במארג על ידי הפקת תוכן דינמי והידוד עם הלקוחות שלו בהליך בקשה-תגובה.

15.12.23 **פְּרוֹטוֹקוֹל גִּישָׁה פְּשׁוּטָה לַעֲצָמִים** **Simple Object Access Protocol**

SOAP **SOAP** [abb]

פרוטוקול להפעלת שיטות בעצמים מבוזרים במרשתת, המשתמש במבני נתונים מבוססי XML המועברים בפרוטוקול HTTP.

15.12.24 **שֵׁרוּתֵי מַאֲרָג** **Web services**

רכיבי תוכנה בצימוד רופף המסוגלים לשתף פעולה זה עם זה במרשתת לשם הספקת תוצאה מוגדרת למשתמש קצה.

הערה:

בדרך כלל באמצעות פרוטוקול HTTP.

15.12.25 **רְכִיב מַאֲרָג** **Web component**

רכיב המספק שירותים בתגובה לבקשות.

15.12.26 **פַּקֶּד** **control**

רכיבים כגון לחיצים, תיבות וסרגלי גלילה, המוצגים על ידי היישום.

15.12.27 **מַעֲרַךְ פַּקְּדִים** **control array**

קבוצת **פקדים** מאותו טיפוס ובעלי אותו שם, השותפים לאותה קבוצה של **נוהלי אירועים**.

15.12.28 **תֵּבַת תִּיּוּג** **check box**

תֵּבַת סִמּוּן

פקד המשמש לסימון פריט.

15.12.29 **רְשִׁימַת תִּיּוּג** **checklist**

מערך פקדים מסוג של **תיבת תיוג** (**תיבת סימון**), המאפשר לסמן פריט אחד או יותר.

15.12.30 **פַּקַּד תְּמוּנָה** **image control**

פקד המשמש ליצירת עצם מסוג תמונה.

15.12.31 **תֵּבַת רְשִׁימָה** **list box**

פקד המשמש להצגת רשימת חלופות שהמשתמש בוחר מהן.

15.12.32 **קְבוּצַת לְחִיצֵי בְּרִירָה** **option buttons group**

radio buttons group

פקד המאפשר למשתמש לבחור פריט אחד מתוך רשימת חלופות.

15.12.33 **לְחִיץ בְּרִירָה** **option button**

radio button

לחיץ ב**קבוצת לחיצי ברירה** שבחירה בו מבטלת את הבחירה של שאר הלחיצים בקבוצה.

15.12.34 **תֵּבַת תַּמְלִיל** **text box**

פקֶד מרובע שהוא שטח ב**טופס** שבו המשתמש יכול להזין נתונים.

15.12.35 **עֶרְכַּת כֵּלִים** **toolbox**

שטח תצוגה המכיל כלים המשמשים לתפעול היישום.

דוגמה:

ערכת כלים לסרטוט ולציור.

15.12.36 **סַרְגֵּל כֵּלִים** **tool bar**

חלק במנשק משתמש הכולל לחיצים לביצוע פעולות שגרתיות.

הערה: בדרך כלל הלחיצים הם איקונים.

15.12.37 **תֵּבָה מְשֻׁלֶּבֶת** **combo box**

פקד שהוא שילוב של **תיבת רשימה** עם **תיבת תמליל**, והמיועד לקליטת ערך אחד שאותו יכול המשתמש לבחור מתוך קבוצת ערכים, או להקלידו ידנית.

15.12.38 **מִנְשָׁק יִשּׁוּם** **application interface**

המצג המופיע בעת הרצת יישום.

15.12.39 **דַּפְדְּפַן עֲצָמִים** **object browser**

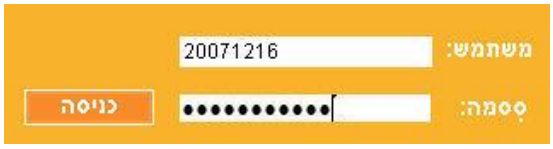
פקֶד המציג עצמים שונים הזמינים ביישום ומאפשר לבחור מהם.

15.12.40 **תּוֹ סִסְמָה** **password char**

סימן המחליף כל תו המוקלד בתיבת סִסמה.

הערה:

ראו דוגמה.



סִסמה מוגנת על ידי תו עיגול שחור

15.12.41 **סִמְלֵי תַּהֲלִיךְ** **process symbols**

סמלים המשמשים לייצוג משימות שונות שעל המחשב להשלים במהלך הרצת תוכנית.

15.12.42 **מַצַּג פְּתִיחָה** **splash screen**

התמונה הראשונה המופיעה בעת הרצת יישום.

15.12.43 **מַקֵּשׁ קִצּוּר** **access key**

shortcut key

keyboard shortcut

מקש או צירוף מקשים המאפשר למשתמש לבצע סדרת פעולות ארוכה בדרך מקוצרת.

15.12.44 **לְחִיץ** **button**

עצם שעליו המשתמש יכול להקיש.

15.12.45 **כִּתּוּבִית** **caption**

תווית תיאור של **לחיץ** או **פקד**.

15.12.46 **לְחִיץ פְּקֻדָּה** **command button**

במנשק משתמש, לחיץ שנועד לאפשר למשתמש לבצע פעולות.

15.12.47 **מַקֵּשׁ פִּקּוּד** **control key**

מקש במקלדת המשמש לביצוע פקודה ללא שימוש בעכבר.

דוגמות:

Page Up, Home, End.

15.12.48 **לְחִיץ בְּרִירַת מֶחְדָּל** **default button**

לחיץ מודגש שהמערכת מציעה למשתמש.

15.12.49 **רֶמֶז צָץ** **tooltip**

תיאור קצר המופיע כאשר מצביעים על עצם כלשהו במנשק גרפי.

15.12.50 **בְּחִירָה** **selecting**

הפעולה ההופכת את אחד העצמים למוקד על המרקע.

15.12.51 **הַצָּגָה** **show**

הקרנת מצג על המרקע.

15.12.52 **הוֹדָעַת-מַעֲרֶכֶת מְחַיֶּבֶת** **system modal**

הודעת-מערכת שהמשתמש חייב להגיב עליה כדי להמשיך בפעילות.

15.12.53 **סֵדֶר טַבְלָר** **Tab order**

הַסֵּדר שבו המיקוד עובר מעצם לעצם עם כל לחיצה על המקש "טבלר" (Tab).

15.12.54 **מוֹקֵד** **focus**

העצם הפעיל ביישום.

15.12.55 **הַסְתָּרָה** **hide**

העלמת מַצָּג מהמרקע ושמירתו בזיכרון.

15.12.56 **מִלַּת מַפְתֵּחַ** **keyword**

ערך מילוני המאפיין **צרף-שפה** מסוים בהקשרים מסוימים.

הערה:

למילת מפתח יש בדרך כלל צורה מזהה.

15.12.57 **חַלּוֹנִית נִווּט** **navigation pane**

consol tree

חלונית שבה מוצג עץ מדרגי של תיקיות שדרכן ניתן להגיע (לנווט) אל תת-תיקיות או אל עצם רצוי ולבחור אותו.

הערה:

בדרך כלל נלווים לתצוגת עץ הניווט סימוני +/- המאפשרים למשתמש לפרוס או לכווץ את התוכן המוצג.

15.12.58 **חַלוֹנִית פְּרָטִים** **details pane**

החלונית שבה מוצגים הפרטים של העצם הנבחר בחלונית הניווט.

15.12.59 **חַלוֹן נִגְלָל** **scrolling window**

החלק של מרחב התצוגה הממופה אל אשנב ההצגה בכל רגע נתון.

15.12.60 **מַצָּג** **panel**

מסגרת מלבנית המוצגת על המרקע שבה מוגדרים מראש מיקומים ומאפיינים של הרכיבים המוצגים בתוך המסגרת.

הערה:

מצג יכול להכיל תמליל מידע, שדות להזנה, אפשרויות לבחירת המשתמש או תערובת של כל אלה. דוגמות:

מצג תפריט, מצג עזרה.

15.12.61 **מַצַּג עֶזְרָה** **help panel**

ריכוז של מידע המיועד להדרכת משתמש ולסיוע בתפעול תחנת עבודה, מסוף או תוכנה.

15.12.62 **סַרְגֵּל כֵּלִים** **tool bar**

אזור במצג של יישום, שבו מרוכזים איקונים המיועדים להפעלת תפקודים שונים של היישום.

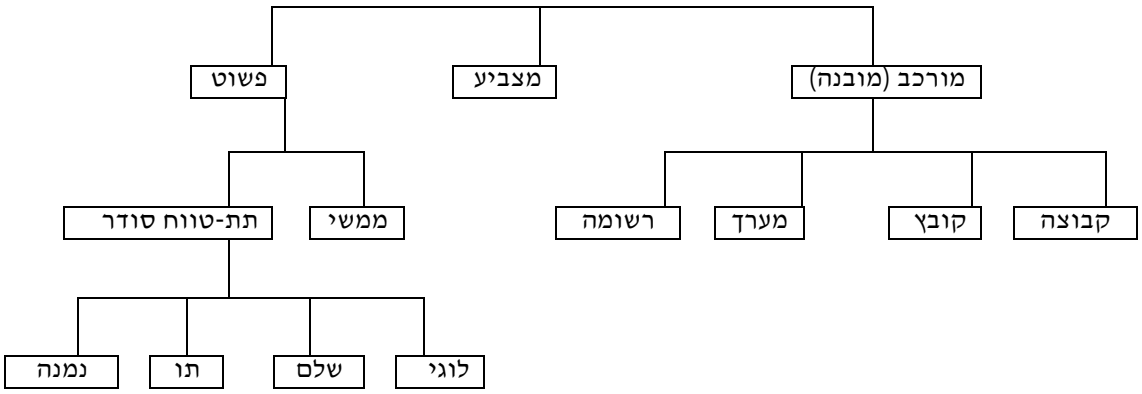
הערה:

הסרגל מוצג, בדרך כלל כשורת איקונים בראש או בתחתית המצג או כעמודה באחד מצדדיו.

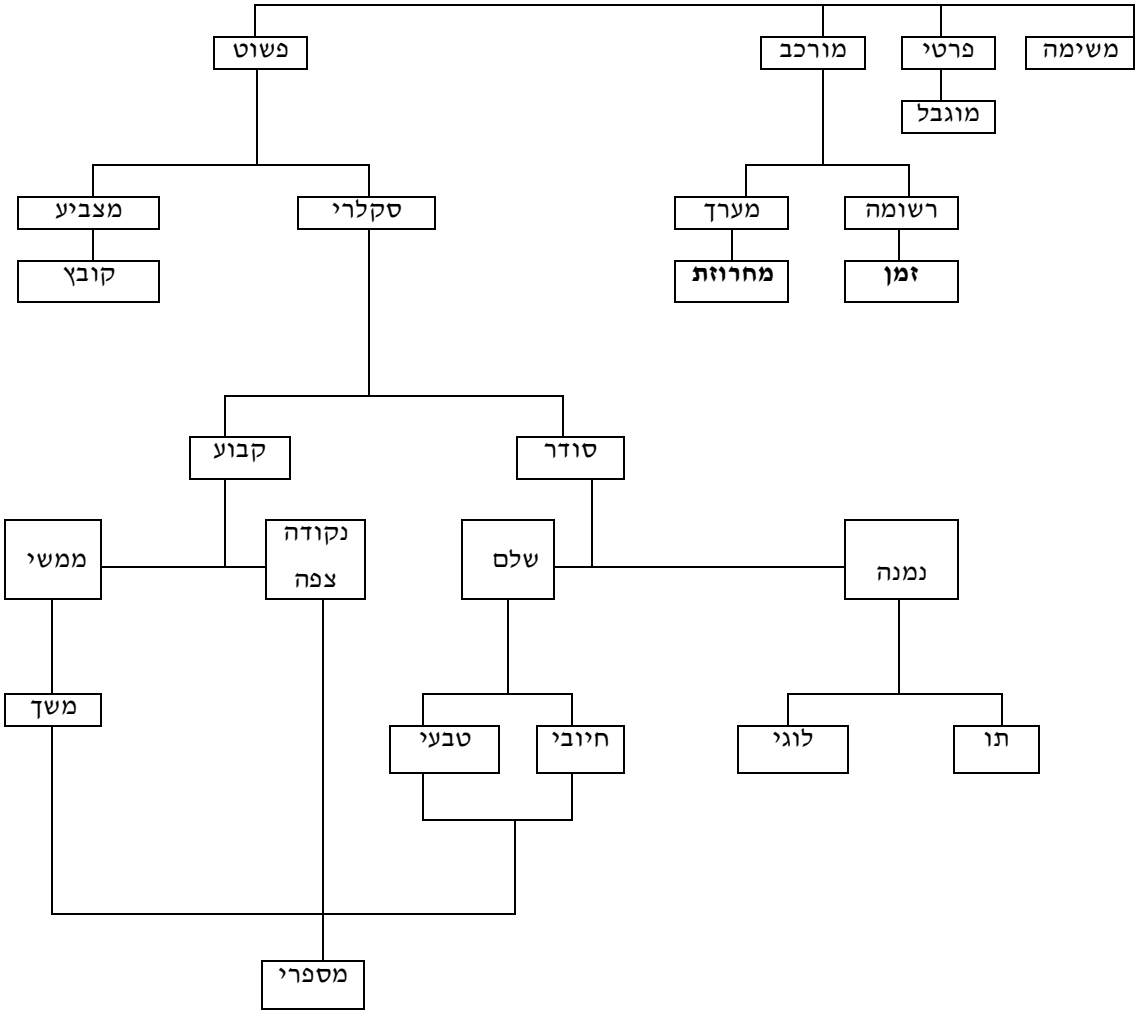
דוגמות:

סרגל כלי ציור, סרגל כלים לעריכת תמונות.

איורים

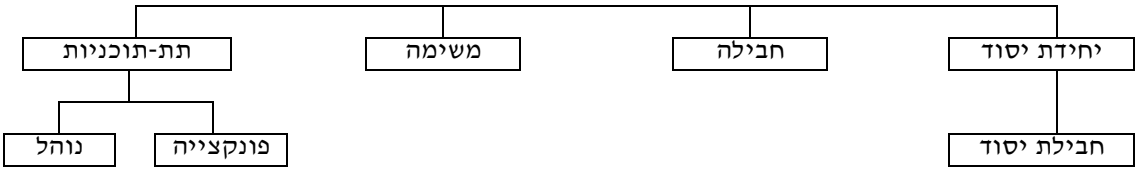


א. טיפוסי נתונים ב-Pascal

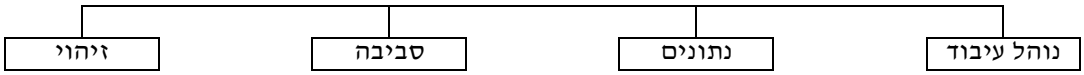


ב. טיפוסי נתונים ב-Ada

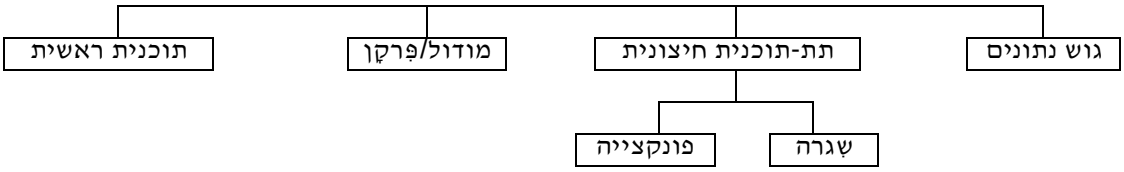
ציור 1 – דוגמות של טיפוסי נתונים ב-Pascal וב-Ada



א. פרקנים ב-Ada



ב. חטיבות נתונים ב-Cobol



ג. יחידות תוכנית ב-Fortran

ציור 2 – דוגמות של חלקי תוכניות

ציור 3 – סמלי תרשים זרימה

1. סמלי קלט-פלט	
קלט-פלט (נתונים) input/output	תצוגה display
קלט ידני manual input	דיסק disk
מסמך document	סרט מגנטי magnetic tape
מסמכים מרובים multiple documents	סרט נייר paper tape
כרטיס card	
2. סמלי עיבוד	
תהליך (עיבוד) process	מַחבֵּר connector
תהליך חֲלוּפי alternate process	מחבר לדף אחר off page connector
תהליך מוגדר מראש predefined process	סְיָם (משמש גם לסימון התחלה) terminator, terminal
קווי זרימה וכיוון זרימה flow lines & direction	צומת מסכם totaling node
החלטה decision	או or
הכנה preparation	חילוץ extract
פעולה ידנית manual operation	מיזוג merge
איסוף collection	השהיה delay
מיון sort	מצב פעולות מקבילות parallel mode

3. **סמלי אחסון**			
אחסון פנימי internal storage		אחסון מקוון online storage	
אחסון נתונים מקוון online data storage		אחסון לא מקוון offline storage	

רשימת מונחים עברית-אנגלית

English Term	מונח עברי	מספר מונח
delegation	אֲצִילָה	15.09.14
passivation	אִבְּטוּל	15.12.13
union	אִגּוּד	15.05.38
automatic instantiation	אִדְּגּוּם אוֹטוֹמָטִי	15.09.40
generic instantiation	אִדְּגּוּם יְסוֹד	15.06.35
explicit instantiation	אִדְּגּוּם מְפֹרָשׁ	15.09.41
directed instantiation	אִדְּגּוּם מֻנְחֶה	15.09.42
constraint	אִלּוּץ	15.04.25
user data constraint	אִלּוּץ נְתוּנֵי מִשְׁתַּמֵּשׁ	15.08.26
initialize [v]	אִתְּחֵל	15.10.03
member-wise initialization	אִתְּחוּל חָבֵר מֵחָבֵר	15.09.46
declarative part	אֵזוֹר הַהַכְרָזוֹת	15.02.02
declarative region	אֵזוֹר הַהַכְרָזוֹת	15.02.02
visibility region	אֵזוֹר נִרְאוּת	15.02.17
event	אֵרוּעַ	15.11.01
direct instance	אֶדְגָּם יָשִׁיר	15.09.43
generic instance	אֶדְגָּם יְסוֹד	15.06.36
template argument	אַרְגּוּמֶנְט תַּבְנִית	15.09.16
template	אַרְגּוּמֶנְט תַּבְנִית	15.09.16
Common Object Request Broker Architecture	אַרְכִּיטֶקְטוּרָה אֲחִידָה לִמְתַוֵּךְ שְׁמוֹת עֲצָמִים	15.03.30
CORBA [abb]	אַרְכִּיטֶקְטוּרָה אֲחִידָה לִמְתַוֵּךְ שְׁמוֹת עֲצָמִים	15.03.30
Atomicity, Consistency, Isolation, and Durability	אָטוֹמִיּוּת, עֲקִיבוּת, בִּדּוּד וְקַיָּמוּת (עֲמִידוּת)	15.08.20
ACID [abb]	אָטוֹמִיּוּת, עֲקִיבוּת, בִּדּוּד וְקַיָּמוּת (עֲמִידוּת)	15.08.20
lifetime	אֹרֶךְ חַיִּים	15.02.15
relational operator	אוֹפֵרָטוֹר יַחְסָה	15.05.41
logical operator	אוֹפֵרָטוֹר לוֹגִי	15.05.40
Unicode	אוּנִיקוֹד	15.12.18
selecting	בְּחִירָה	15.12.50
default	בְּרֵרַת מֶחְדָּל	15.02.03
expression	בִּטּוּי	15.05.33
Boolean expression	בִּטּוּי לוֹגִי	15.05.35
conditional expression	בִּטּוּי תְּנַאי	15.05.14
timer control	בַּקָּרָה עַל-יְדֵי קוֹצֵב זְמַן	15.08.27
bug	בָּג	15.00.14
constructor	בּוֹנֶה	15.09.47
protected access	גִּישָׁה מוּגֶנֶת	15.09.36
private access	גִּישָׁה פְּרָטִית	15.09.35
public access	גִּישָׁה צִבּוּרִית	15.09.37
body	גּוּף	15.06.02
generic body	גּוּף יְסוֹד	15.06.31
basic block	גּוּשׁ בְּסִיסִי	15.06.40

English Term	מונח עברי	מספר מונח
dynamic	דִּינָמִי	15.02.14
object browser	דַּפְדֵּפַן עֲצָמִים	15.12.39
type conversion	הֲמָרַת טִפּוּס	15.04.30
type casting	הֲמָרַת טִפּוּס	15.04.30
explicit type casting	הֲמָרַת טִפּוּס מְפֹרֶשֶׁת	15.04.41
explicit type conversion	הֲמָרַת טִפּוּס מְפֹרֶשֶׁת	15.04.41
automatic type conversion	הֲמָרַת טִפּוּס מִשְׁתַּמֵּעַת	15.04.42
implied type conversion	הֲמָרַת טִפּוּס מִשְׁתַּמֵּעַת	15.04.42
coercion	הֲמָרַת טִפּוּס מִשְׁתַּמֵּעַת	15.04.42
rollback	הֲסָגָה	15.08.18
name demangling	הֲסָרַת יִחוּד שֵׁמוֹת	15.08.09
disambiguation	הֲפָגַת עֲמִימוּת	15.01.09
return [v] [transitive]	הֶחְזִיר	15.05.08
comment	הֶעָרָה	15.01.11
remark	הֶעָרָה	15.01.11
method permission	הֶתֵּר שִׁיטָה	15.09.51
macrodefinition	הַגְדָּרַת מַקְרוֹ	15.06.24
Document Type Definition	הַגְדָּרַת טִפּוּס מִסְמָךְ	15.00.12
DTD [abb]	הַגְדָּרַת טִפּוּס מִסְמָךְ	15.00.12
strong typing	הַטְפָּסָה חֲזָקָה	15.04.31
weak typing	הַטְפָּסָה חַלָּשָׁה	15.04.32
package declaration	הַכְרָזַת חֲבִילָה	15.06.26
generic declaration	הַכְרָזַת יְסוֹד	15.06.30
variable declaration	הַכְרָזַת מִשְׁתַּנֶּה	15.02.18
declaration	הַכְרָזָה	15.02.01
global declaration	הַכְרָזָה כְּלָלִית	15.02.19
local declaration	הַכְרָזָה מְקוֹמִית	15.02.20
implicit declaration	הַכְרָזָה מִשְׁתַּמֵּעַת	15.02.04
information hiding	הַסְתָּרַת מֵידָע	15.09.01
hide	הַסְתָּרָה	15.12.55
pass [n]	הַעֲבָרָה	15.08.15
evaluation	הַעֲרָכָה	15.08.28
decrementing	הַפְחָתָה	15.08.14
reference	הַפְנָיָה	15.10.06
uniform referencing	הַפְנָיָה אֲחִידָה	15.10.07
indirect referencing	הַפְנָיָה עֲקִיפָה	15.10.02
software pipelining	הַצְנָרַת תֻּכְנָה	15.08.07
assignment	הַצָּבָה	15.05.04
show	הַצָּגָה	15.12.51
dynamic storage allocation	הַקְצָאַת זִכָּרוֹן דִּינָמִית	15.10.04
dynamic memory allocation	הַקְצָאַת זִכָּרוֹן דִּינָמִית	15.10.04
extension	הַרְחָבָה	15.06.66
textual comparison	הַשְׁוָאַת תַּמְלִיל	15.12.16
binary comparison	הַשְׁוָאָה בֵּינָרִית	15.08.10
garbage collection	הַשְׁמָשָׁה	15.08.21

English Term	מונח עברי	מספר מונח
system modal	הוֹדַעַת-מַעֲרֶכֶת מְחַיֶּבֶת	15.12.52
incrementing	הוֹסָפָה	15.08.13
destructor	הוֹרֵס	15.09.48
macroinstruction	הוֹרָאַת מַקְרוֹ	15.06.22
break time	זְמַן הַפְּסָקָה	15.12.15
control flow	זְרִימַת בַּקָּרָה	15.08.02
data flow	זְרִימַת נְתוּנִים	15.00.15
stream	זֶרֶם	15.08.22
sentinel	זָקִיף	15.01.17
package	חֲבִילָה	15.06.25
generic package	חֲבִילָה גֶּנֶרִיתחֲבִילַת יְסוֹד	15.06.33
data division	חֲטִיבַת הַנְּתוּנִים	15.02.02
connection	חִבּוּר	15.06.17
external	חִיצוֹנִי	15.02.12
variant part	חֵלֶק מִשְׁתַּנֶּה	15.03.12
visible part	חֵלֶק נִרְאֶה	15.06.27
private part	חֵלֶק פְּרָטִי	15.06.28
scrolling window	חַלּוֹן נַגְלָל	15.12.59
navigation pane	חַלּוֹנִית נִוּוּט	15.12.57
consol tree	חַלּוֹנִית נִוּוּט	15.12.57
details pane	חַלּוֹנִית פְּרָטִים	15.12.58
protected member	חָבֵר מוּגָן	15.09.38
public member	חָבֵר צִבּוּרִי	15.09.39
return [v] [intransitive]	חָזַר	15.05.07
component contract	חוֹזֵה רְכִיב	15.06.54
timestamp	חוֹתֶמֶת זְמַן	15.12.10
range	טְוָח	15.04.08
base type	טִפּוּס בָּסִיס	15.04.24
host type	טִפּוּס בָּסִיס	15.04.24
underlying type	טִפּוּס בָּסִיס	15.04.24
single-precision type	טִפּוּס דִּיּוּק-יָחִיד	15.04.39
double-precision type	טִפּוּס דִּיּוּק-כָּפוּל	15.04.40
parent type	טִפּוּס הוֹרֶה	15.04.28
encapsulated type	טִפּוּס כָּמוּס	15.04.03
incomplete type	טִפּוּס לֹא-מֻשְׁלָם	15.04.38
atomic type	טִפּוּס לֹא-פָּרִיק	15.04.06
logical type	טִפּוּס לוֹגִי	15.04.07
Boolean type	טִפּוּס לוֹגִי	15.04.07
Bool type	טִפּוּס לוֹגִי	15.04.07
numeric type	טִפּוּס מִסְפָּרִי	15.04.16
variant type	טִפּוּס מִשְׁתַּנֶּה	15.04.43
string type	טִפּוּס מַחְרוֹזֶת	15.04.18
real type	טִפּוּס מַמָּשִׁי	15.04.09
array type	טִפּוּס מַעֲרָךְ	15.04.20
pointer type	טִפּוּס מַצְבִּיעַ	15.04.19

English Term	מונח עברי	מספר מונח
access type	טִפּוּס מַצְבִּיעַ	15.04.19
predefined type	טִפּוּס מֻבְנֶה	15.04.33
limited type	טִפּוּס מֻגְבָּל	15.04.27
abstract type	טִפּוּס מֻפְשָׁט	15.04.02
ADT [abb]	טִפּוּס מֻפְשָׁט	15.04.02
floating-point type	טִפּוּס נְקֻדָּה צָפָה	15.04.11
fixed-point type	טִפּוּס נְקֻדָּה קְבוּעָה	15.04.10
implied decimal type	טִפּוּס נְקֻדָּה קְבוּעָה	15.04.10
data type	טִפּוּס נְתוּנִים	15.04.01
datatype	טִפּוּס נְתוּנִים	15.04.01
derived type	טִפּוּס נִגְזָר	15.04.29
enumeration type	טִפּוּס נִמְנֶה	15.04.15
enumerated type	טִפּוּס נִמְנֶה	15.04.15
scalar type	טִפּוּס סְקָלָרִי	15.04.04
universal type	טִפּוּס סְתָמִי	15.04.34
ordinal type	טִפּוּס סוֹדֵר	15.04.12
discrete type	טִפּוּס סוֹדֵר	15.04.12
private type	טִפּוּס פְּרָטִי	15.04.26
simple type	טִפּוּס פָּשׁוּט	15.04.05
index type	טִפּוּס צִיּוּן	15.04.13
record type	טִפּוּס רְשׁוּמָה	15.04.21
variant record type	טִפּוּס רְשׁוּמָה מִשְׁתַּנָּה	15.04.22
integer type	טִפּוּס שָׁלֵם	15.04.14
character type	טִפּוּס תָּו	15.04.17
manipulator	טַפְלָל	15.08.12
roundoff error	טָעוּת עִגּוּל	15.08.24
form	טֹפֶס	15.06.60
startup form	טֹפֶס הֶזְנֵק	15.06.65
modal form	טֹפֶס חוֹבָה	15.06.63
toplevel form	טֹפֶס עֶלְיוֹן	15.06.67
modeless form	טֹפֶס רְשׁוּת	15.06.64
link unit	יְחִידַת קִשּׁוּר	15.06.44
translation unit	יְחִידַת תִּרְגּוּם	15.06.43
generic	יְסוֹד [שם תואר]	15.06.29
inheritance	יְרִישָׁה	15.09.13
single inheritance	יְרִישָׁה מִיָּחִיד	15.09.54
multiple inheritance	יְרִישָׁה מֵרַבִּים	15.09.53
name mangling	יִחוּד שֵׁמוֹת	15.08.08
distributed application	יִשּׁוּם מְבֻזָּר	15.12.11
peer entity	יֵשׁוּת עָמִית	15.12.20
forest	יַעַר	15.06.71
object-oriented	כִּוּוּן-עֲצָמִים	15.09.15
global	כְּלָלִי	15.02.11
rules of precedence	כְּלָלֵי קִדִּימוּת	15.00.03
encapsulation	כְּמִיסָה	15.09.03

English Term	מונח עברי	מספר מונח
entry	כְּנִיסָה	15.05.09
immutable binding	כְּרִיכָה מְקֻבַּעַת	15.08.29
caption	כִּתּוּבִית	15.12.45
alias	כִּנּוּי	15.03.21
encapsulate [v]	כָּמַס	15.09.02
button	לְחִיץ	15.12.44
option button	לְחִיץ בְּרִירָה	15.12.33
radio button	לְחִיץ בְּרִירָה	15.12.33
default button	לְחִיץ בְּרִרַת מֶחְדָּל	15.12.48
command button	לְחִיץ פְּקֻדָּה	15.12.46
application client	לָקוֹחַ יִשּׁוּם	15.12.19
disabled	לֹא-מְאֻפְשָׁר	15.10.12
nested loops	לוּלָאָה מְקוֹנֶנֶת	15.06.49
bottom-driven loops	לוּלָאָה נִבְדֶּקֶת זָנָב	15.06.48
top-driven loops	לוּלָאָה נִבְדֶּקֶת רֹאשׁ	15.06.47
context attribute	מְאַפְיֵן הֶקְשֵׁר	15.10.08
enabled	מְאֻפְשָׁר	15.10.11
distributed	מְבֻזָּר	15.00.06
identifier	מְזַהֶה	15.01.03
predefined identifier	מְזַהֶה מֻבְנֶה	15.01.04
container	מְכָל	15.06.51
resource manager	מְנַהֵל מַשְׁאַבִּים	15.12.12
local	מְקוֹמִי	15.02.10
task	מְשִׂימָה	15.07.02
Object Request Broker	מְתַוֵּךְ שְׁמוֹת עֲצָמִים	15.03.33
ORB [abb]	מְתַוֵּךְ שְׁמוֹת עֲצָמִים	15.03.33
deployment descriptor	מְתָאֵר פְּרִיסָה	15.10.09
data structure	מִבְנֵה נְתוּנִים	15.03.01
Easter egg	מִטְמָן	15.06.58
basic optimizations	מֵטוּב בְּסִיסִי	15.08.04
profile-based optimization	מֵטוּב מְבֻסָּס-מִתְאָר	15.08.06
aggressive optimization	מֵטוּב תַּקִּיף	15.08.05
keyword	מִלַּת מַפְתֵּחַ	15.12.56
reserved word	מִלָּה שְׁמוּרָה	15.01.05
lexical	מִלּוֹנִי	15.01.01
application interface	מִנְשָׁק יִשּׁוּם	15.12.38
rendezvous	מִפְגָּשׁ	15.07.05
aggregate	מִצְרָף	15.03.06
assumed-size aggregate	מִצְרָף בְּגֹדֶל מְשֹׁעָר	15.03.08
adjustable-size aggregate	מִצְרָף בְּגֹדֶל מִתְכַּוְנֵן	15.03.09
substitution variable	מִשְׁתַּנֶּה מִתְעַדְכֵּן	15.02.33
if statement	מִשְׁפַּט "אִם"	15.05.15
do while statement	מִשְׁפַּט "בַּצַּע כָּל-עוֹד"	15.05.21
until statement	מִשְׁפַּט "בַּצַּע עַד-אֲשֶׁר"	15.05.22
perform until statement	מִשְׁפַּט "בַּצַּע עַד-אֲשֶׁר"	15.05.22

English Term	מונח עברי	מספר מונח
perform statement	מִשְׁפַּט "בַּצַּע"	15.05.23
case statement	מִשְׁפַּט "מִקְרֶה"	15.05.16
goto statement	מִשְׁפַּט "עֲבוֹר"	15.05.11
repeat while statement	מִשְׁפַּט do while	15.05.21
perform while statement	מִשְׁפַּט repeat while	15.05.21
repeat until statement	מִשְׁפַּט until	15.05.22
select statement	מִשְׁפַּט בְּרִירָה	15.05.31
block statement	מִשְׁפַּט גּוּשׁ	15.05.24
selective-wait statement	מִשְׁפַּט הַמְתָּנָה בּוֹרֶרֶת	15.05.32
raise statement	מִשְׁפַּט הַעֲלָאַת חָרִיג	15.05.29
assignment statement	מִשְׁפַּט הַצָּבָה	15.05.04
delay statement	מִשְׁפַּט הַשְׁהָיָה	15.05.27
return statement	מִשְׁפַּט חֲזָרָה	15.05.06
iteration statement	מִשְׁפַּט חִזְרוּר	15.05.17
loop statement	מִשְׁפַּט חִזְרוּר	15.05.17
load statement	מִשְׁפַּט טְעִינָה	15.05.43
exit statement	מִשְׁפַּט יְצִיאָה	15.05.05
abort statement	מִשְׁפַּט נְטִישָׁה	15.05.28
unload statement	מִשְׁפַּט פְּרִיקָה	15.05.44
imperative statement	מִשְׁפַּט צִוּוּי	15.05.12
entry-call statement	מִשְׁפַּט קְרִיאַת כְּנִיסָה	15.05.26
procedure-call statement	מִשְׁפַּט קְרִיאָה לְנֹהַל	15.05.25
accept statement	מִשְׁפַּט קַבָּלָה	15.05.30
conditional statement	מִשְׁפַּט תְּנַאי	15.05.13
statement	מִשְׁפָּט	15.05.01
unconditional statement	מִשְׁפָּט לֹא-מֻתְנֶה	15.05.12
compound statement	מִשְׁפָּט מֻרְכָּב	15.05.03
simple statement	מִשְׁפָּט פָּשׁוּט	15.05.02
variable	מִשְׁתַּנֶּה	15.03.03
form-level variable	מִשְׁתַּנֶּה בְּרָמַת טֹפֶס	15.06.62
dynamic variable	מִשְׁתַּנֶּה דִּינָמִי	15.02.23
global variable	מִשְׁתַּנֶּה כְּלָלִי	15.02.25
undeclared variable	מִשְׁתַּנֶּה לֹא-מֻכְרָז	15.02.21
local variable	מִשְׁתַּנֶּה מְקוֹמִי	15.02.24
static variable	מִשְׁתַּנֶּה סְטָטִי	15.02.22
parameter profile	מִתְאַר פָּרָמֶטְרִים	15.02.30
extensibility	מִתְרַחֲבוּת	15.10.05
metaclass	מֵטָא-מַחְלָקָה	15.09.22
message	מֶסֶר	15.09.08
namespace	מֶרְחַב שֵׁמוֹת	15.06.70
discriminant]n [	מַבְחִין	15.03.14
connector	מַחְבֵּר	15.06.53
base class	מַחְלֶקֶת בָּסִיס	15.09.18
root base class	מַחְלֶקֶת בָּסִיס שֹׁרֶשׁ	15.09.19
core class	מַחְלֶקֶת לִבָּה	15.09.21

English Term	מונח עברי	מספר מונח
class	מַחְלָקָה	15.09.11
instantiated class	מַחְלָקָה מְאֻדְגָּמֶת	15.09.20
abstract class	מַחְלָקָה מֻפְשֶׁטֶת	15.09.23
abstract superclass	מַחְלָקָה מֻפְשֶׁטֶת	15.09.23
empty string	מַחְרֹוֶזֶת רֵיקָה	15.03.27
null stringzero-length string	מַחְרֹוֶזֶת רֵיקָה	15.03.27
runtime system	מַעֲרֶכֶת זְמַן-רִיצָה	15.12.17
control array	מַעֲרַךְ פַּקָּדִים	15.12.27
array	מַעֲרָךְ	15.03.10
dynamic array	מַעֲרָךְ דִּינָמִי	15.03.29
separator	מַפְרִיד	15.01.07
pointer	מַצְבִּיעַ	15.03.22
smart pointer	מַצְבִּיעַ חָכָם	15.05.42
null pointer	מַצְבִּיעַ רֵיק	15.03.23
formal parameter mode	מַצַּב פָּרָמֶטֶר פּוֹרְמָלִי	15.06.20
help panel	מַצַּג עֶזְרָה	15.12.61
splash screen	מַצַּג פְּתִיחָה	15.12.42
panel	מַצָּג	15.12.60
macro	מַקְרוֹ	15.06.21
control key	מַקַּשׁ פִּקּוּד	15.12.47
access key	מַקַּשׁ קִצּוּר	15.12.43
shortcut key	מַקַּשׁ קִצּוּר	15.12.43
keyboard shortcut	מַקַּשׁ קִצּוּר	15.12.43
predefined	מֻבְנֶה	15.02.05
built-in	מֻבְנֶה	15.02.05
intrinsic	מֻבְנֶה	15.02.05
implementation defined	מֻגְדָּר-מִמּוּשׁ	15.02.28
Document Object Model	מוֹדֵל עַצְמֵי מִסְמָךְ	15.00.11
DOM [abb]	מוֹדֵל עַצְמֵי מִסְמָךְ	15.00.11
Application Programming Model	מוֹדֵל תִּכְנוּת יִשּׁוּם	15.00.10
APM [abb]	מוֹדֵל תִּכְנוּת יִשּׁוּם	15.00.10
monitor	מוֹנִיטוֹר, מַשְׁגִּוֹחַ	15.07.07
focus	מוֹקֵד	15.12.54
protected	מוּגָן	15.09.06
event-driven	מוּנַע אֵרוּעִים	15.11.03
superclass	מחלקת-על	15.09.24
invalid data	נְתוּנִים לֹא-תְּקֵפִים	15.03.24
shared data	נְתוּנִים מְשֻׁתָּפִים	15.02.07
valid data	נְתוּנִים תְּקֵפִים	15.03.25
visibility	נִרְאוּת	15.02.16
data member	נָתוּן חָבֵר	15.09.26
event procedure	נֹהַל אֵרוּעַ	15.11.02
asynchronous procedure	נֹהַל אָסִינְכְּרוֹנִי	15.06.69
general procedure	נֹהַל כְּלָלִי	15.06.37
user-defined procedure	נֹהַל מֻגְדָּר-מִשְׁתַּמֵּשׁ	15.06.38

English Term	מונח עברי	מספר מונח
closing	סְגִירָה	15.09.44
static	סְטָטִי	15.02.13
process symbols	סִמְלֵי תַּהֲלִיךְ	15.12.41
parameter association	סִמּוּךְ פָּרָמֶטְרִים	15.03.18
positional parameter association	סִמּוּךְ פָּרָמֶטְרִים לְפִי מָקוֹם	15.06.19
named parameter association	סִמּוּךְ פָּרָמֶטְרִים לְפִי שֵׁם	15.06.18
assignment by name	סִמּוּךְ פָּרָמֶטְרִים לְפִי שֵׁם	15.06.18
dot notation	סִמּוּן בִּנְקֻדָּה	15.00.05
task synchronization	סִנְכְּרוּן מְשִׂימוֹת	15.07.04
archive library	סִפְרִיַּת אַרְכִיּוֹן	15.12.01
shared library	סִפְרִיָּה מְשֻׁתֶּפֶת	15.12.02
execution sequence	סֵדֶר בִּצּוּעַ	15.08.01
Tab order	סֵדֶר טַבְלָר	15.12.53
start/stop symbol	סֵמֶל הַתְּחָלָה אוֹ סִיּוּם	15.01.18
external symbol	סֵמֶל חִיצוֹנִי	15.06.45
flowchart symbol	סֵמֶל לְתַרְשִׁים זְרִימָה	15.00.16
semaphore	סֵמָפוֹר	15.07.06
tool bar	סַרְגֵּל כֵּלִים	15.12.36
tool bar	סַרְגֵּל כֵּלִים	15.12.62
anonymous	עֲלוּם-שֵׁם	15.04.35
operator overload	עֲמִיסַת אוֹפֵּרָטוֹר	15.05.39
overloading	עֲמִיסַת מַשְׁמָעֻיּוֹת	15.01.08
editing	עֲרִיכָה	15.07.08
sequential processing	עִבּוּד סִדְרָתִי	15.12.14
atomic value	עֵרֶךְ לֹא פָּרִיק	15.03.34
lexical token	עֵרֶךְ מִלּוֹנִי	15.01.01
lexical element	עֵרֶךְ מִלּוֹנִי	15.01.01
lexical unit	עֵרֶךְ מִלּוֹנִי	15.01.01
aggregate value	עֵרֶךְ מִצְרָף	15.03.07
data value	עֵרֶךְ נְתוּנִים	15.03.04
object	עֶצֶם	15.09.07
immutable object	עֶצֶם מְקֻבָּע	15.03.32
data object	עֶצֶם נְתוּנִים	15.03.02
toolbox	עֶרְכַּת כֵּלִים	15.12.35
peer	עֲמִית	15.12.20
wrapper	עוֹטְפָן	15.09.49
generic operation	פְּעֻלַּת יְסוֹד	15.06.32
atomic operation	פְּעֻלָּה לֹא-פְּרִיקָה	15.08.16
private	פְּרָטִי	15.09.04
protocol	פְּרוֹטוֹקוֹל	15.09.09
SSL protocol	פְּרוֹטוֹקוֹל SSL	15.12.21
Simple Object Access Protocol	פְּרוֹטוֹקוֹל גִּישָׁה פְּשׁוּטָה לַעֲצָמִים	15.12.23
SOAP [abb]	פְּרוֹטוֹקוֹל גִּישָׁה פְּשׁוּטָה לַעֲצָמִים	15.12.23
Secure Socket Layer protocol	פְּרוֹטוֹקוֹל שִׁכְבַת שְׁקָעִים מְאֻבְטַחַת	15.12.21
form module	פִּרְקַן טֹפֶס	15.06.61

English Term	מונח עברי	מספר מונח
generic module	פִּרְקָן יְסוֹד	15.06.34
code module	פִּרְקָן קוֹד	15.06.46
module	פִּרְקָן, מוֹדוּל	15.06.01
program unit	פִּרְקָן, מוֹדוּל	15.06.01
address resolving	פִּתְרוֹן-מַעַן	15.08.11
array sliceslice	פֶּלַח מַעֲרָךְ	15.03.11
image control	פַּקַּד תְּמוּנָה	15.12.30
control	פַּקַּד	15.12.26
parameter	פָּרָמֶטֶר	15.03.15
initialization parameter	פָּרָמֶטֶר אִתְחוּל	15.06.56
actual parameter	פָּרָמֶטֶר מַמָּשִׁי	15.03.16
actual argument	פָּרָמֶטֶר מַמָּשִׁי	15.03.16
reference parameter	פָּרָמֶטֶר מַעַן	15.02.26
value parameter	פָּרָמֶטֶר עֵרֶךְ	15.02.27
formal parameterdummy argument	פָּרָמֶטֶר פּוֹרְמָלִי	15.03.17
access functions	פּוּנְקְצַיַּת גִּישָׁה	15.09.29
generic function	פּוּנְקְצַיַּת יְסוֹד	15.09.34
template function	פּוּנְקְצַיַּת תַּבְנִית	15.09.28
function	פּוּנְקְצִיָּה	15.06.12
member function	פּוּנְקְצִיָּה חֲבֵרָה	15.09.30
friend function	פּוּנְקְצִיָּה יְדִידָה	15.09.31
instantiated function	פּוּנְקְצִיָּה מְאֻדְגֶּמֶת	15.09.33
inline function	פּוּנְקְצִיָּה תְּחוּבָה	15.09.32
public	צִבּוּרִי	15.09.05
Subscript, index	צִיּוּן	15.10.01
language construct	צֶרֶף-שָׂפָה	15.01.02
for-construct	צֶרֶף "בַּעֲבוּר"	15.05.20
while-construct	צֶרֶף "כָּל-עוֹד"	15.05.18
until-construct	צֶרֶף "עַד-אֲשֶׁר"	15.05.19
decision construct	צֶרֶף הַחְלָטָה	15.05.37
decision structure	צֶרֶף הַחְלָטָה	15.05.37
selection structure	צֶרֶף הַחְלָטָה	15.05.37
iteration construct	צֶרֶף חִזְרוּר	15.05.46
loop construct	צֶרֶף חִזְרוּר	15.05.46
repetition structure	צֶרֶף חִזְרוּר	15.05.46
repetition construct	צֶרֶף חִזְרוּר	15.05.46
branch construct	צֶרֶף סִעוּף	15.05.45
world readable files	קְבָצִים קְרִיאִים לַכֹּל	15.12.03
option buttons group	קְבוּצַת לְחִיצֵי בְּרִירָה	15.12.32
radio buttons group	קְבוּצַת לְחִיצֵי בְּרִירָה	15.12.32
operator precedence	קְדִימוּת אוֹפֶּרָטוֹרִים	15.05.36
call	קְרִיאָה	15.06.05
transaction call	קְרִיאָה לְמַפְגָּשׁ	15.06.14
macrocall	קְרִיאָה לְמַקְרוֹ	15.06.23
procedure call	קְרִיאָה לְנֹהַל	15.05.25

English Term	מונח עברי	מספר מונח
call by reference	קְרִיאָה לְפִי מַעַן	15.06.08
call by address	קְרִיאָה לְפִי מַעַן	15.06.08
call by location	קְרִיאָה לְפִי מַעַן	15.06.08
call by value	קְרִיאָה לְפִי עֵרֶךְ	15.06.09
call by name	קְרִיאָה לְפִי שֵׁם	15.06.07
function call	קְרִיאָה לְפוּנְקְצִיָּה	15.06.13
subprogram call	קְרִיאָה לְתַת-תָּכְנִית	15.06.10
commit	קִבּוּעַ	15.08.17
critical section	קֶטַע קְרִיטִי	15.07.03
constant	קָבוּעַ	15.03.05
string constant	קָבוּעַ מַחְרֹוָזֶת	15.01.13
intrinsic constant	קָבוּעַ מֻבְנֶה	15.01.15
literal constant	קָבוּעַ סְמִין	15.01.16
symbolic constant	קָבוּעַ סִמְלִי	15.01.19
character constant	קָבוּעַ תָּו	15.01.12
call [v]	קָרָא	15.06.06
executable file	קֹבֶץ בָּצִיעַ	15.12.07
header file	קֹבֶץ כּוֹתֶרֶת	15.12.05
source file	קֹבֶץ מָקוֹר	15.12.04
data file	קֹבֶץ נְתוּנִים	15.12.06
sequential file	קֹבֶץ סִדְרָתִי	15.12.09
program file	קֹבֶץ תָּכְנִית	15.12.08
program code	קוֹד תָּכְנִית	15.06.41
Web component	רְכִיב מַאֲרָג	15.12.25
member initialization list	רְשִׁימַת אִתְחוּל חֲבֵרִים	15.09.45
checklist	רְשִׁימַת תִּיּוּג	15.12.29
variant record	רְשׁוּמָה מִשְׁתַּנָּה	15.03.13
tooltip	רֶמֶז צָץ	15.12.49
mixed mode [adj]	רַב-טִיפּוּסִים	15.05.34
mixed type [adj]	רַב-טִיפּוּסִים	15.05.34
polymorphism	רַב-צוּרָתִיּוּת	15.09.12
transaction isolation level	רָמַת בִּדּוּד תְּנוּעָה	15.08.19
logic error	שְׁגִיאָה לוֹגִית	15.08.23
coroutine	שִׁגְרַת הֶמְשֵׁךְ	15.06.04
name qualification qualification	שִׁיּוּךְ שֵׁם	15.03.20
remove method	שִׁיטַת הֲסָרָה	15.09.52
class method	שִׁיטַת מַחְלָקָה	15.09.50
method	שִׁיטָה	15.09.10
callback methods	שִׁיטוֹת מַעֲנֶה	15.08.25
concatenation	שִׁרְשׁוּר	15.00.04
entry name	שֵׁם כְּנִיסָה	15.05.10
Web services	שֵׁרוּתֵי מַאֲרָג	15.12.24
area	שֶׁטַח	15.02.32
servlet	שָׁרָתִית	15.12.22
interface definition language	שְׂפַת הַגְדָּרַת מִנְשָׁק	15.00.08

English Term	מונח עברי	מספר מונח
IDL [abb]	שְׂפַת הַגְּדָרַת מִנְשָׂק	15.00.08
Extensible Markup Language	שְׂפַת סִמּוּן מִתְרַחֶבֶת	15.00.09
XML [abb]	שְׂפַת סִמּוּן מִתְרַחֶבֶת	15.00.09
function template	תַּבְנִית פּוּנְקְצִיּוֹת	15.09.27
scope	תְּחוּם הַכְּרָזָה	15.02.06
scope of a declaration	תְּחוּם הַכְּרָזָה	15.02.06
dynamic scope	תְּחוּם הַכְּרָזָה דִּינָמִי	15.02.08
static scope	תְּחוּם הַכְּרָזָה סְטָטִי	15.02.09
toplevel scope	תְּחוּם עֶלְיוֹן	15.06.68
enabling property	תְּכוּנַת אִפְּשׁוּר	15.10.10
data attribute	תְּכוּנַת נְתוּנִים	15.03.19
property	תְּכוּנָה	15.00.02
trigraph sequences	תְּוַלַּת-תָּו	15.01.14
implementation dependent	תְּלוּי-מִמּוּשׁ	15.02.29
picture	תְּמוּנָה	15.04.37
object-oriented event-driven programming	תִּכְנוּת כְּוּוּן-עֲצָמִים מוּנַע אֵרוּעִים	15.00.07
OOED programming	תִּכְנוּת כְּוּוּן-עֲצָמִים מוּנַע אֵרוּעִים	15.00.07
event-driven programming	תִּכְנוּת מוּנַע אֵרוּעִים	15.11.04
internal documentation	תִּעוּד פְּנִימִי	15.06.42
environment description	תֵּאוּר הַסְּבִיבָה	15.02.31
sandbox	תֵּבַת חוֹל	15.06.52
object box	תֵּבַת עֶצֶם	15.03.31
list box	תֵּבַת רְשִׁימָה	15.12.31
check box	תֵּבַת תִּיּוּג	15.12.28
text box	תֵּבַת תַּמְלִיל	15.12.34
combo box	תֵּבָה מְשֻׁלֶּבֶת	15.12.37
home handle	תֶּפֶס בַּיִת	15.06.55
class template	תַּבְנִית מַחְלָקָה	15.09.17
code template	תַּבְנִית קוֹד	15.06.59
password char	תַּו סִסְמָה	15.12.40
format	תַּסְדִּיר	15.04.36
bean	תַּרְמִיל	15.06.50
subtype	תַּת-טִפּוּס	15.04.23
subunit	תַּת-יְחִידָה	15.06.15
subclass	תַּת-מַחְלָקָה	15.09.25
substring	תַּת-מַחְרוֹזֶת	15.03.26
subarray	תַּת-מַעֲרָךְ	15.03.28
subprocedure	תַּת-נֹהַל	15.06.39
subroutine	תַּת-שִׁגְרָה	15.06.11
subprogram	תַּת-תָּכְנִית	15.06.03
label	תָּוִית	15.01.10
italic	תָּו נָטוּי	15.00.13
main program	תָּכְנִית רָאשִׁית	15.07.01
executable content	תֹּכֶן בָּצִיעַ	15.06.57
delimiter	תּוֹחֵם	15.01.06

English Term	מונח עברי	מספר מונח
side effect	תּוֹצָא לְוָאי	15.08.03
body stub	תּוֹתָב	15.06.16

רשימת מונחים אנגלית-עברית

English Term	מונח עברי	מספר מונח
abort statement	מִשְׁפַּט נְטִישָׁה	15.05.28
abstract class	מַחְלָקָה מֻפְשֶׁטֶת	15.09.23
abstract superclass	מַחְלָקָה מֻפְשֶׁטֶת	15.09.23
abstract type	טִפּוּס מֻפְשָׁט	15.04.02
accept statement	מִשְׁפַּט קַבָּלָה	15.05.30
access functions	פּוּנְקְצִיַּת גִּישָׁה	15.09.29
access key	מַקַּשׁ קִצּוּר	15.12.43
access type	טִפּוּס מַצְבִּיעַ	15.04.19
ACID [abb]	אָטוֹמִיּוּת, עֲקִיבוּת, בִּדּוּד וְקַיָּמוּת (עֲמִידוּת)	15.08.20
actual argument	פָּרָמֶטֶר מַמָּשִׁי	15.03.16
actual parameter	פָּרָמֶטֶר מַמָּשִׁי	15.03.16
address resolving	פִּתְרוֹן-מַעַן	15.08.11
adjustable-size aggregate	מִצְרָף בְּגֹדֶל מִתְכַּוְנֵן	15.03.09
ADT [abb]	טִפּוּס מֻפְשָׁט	15.04.02
aggregate	מִצְרָף	15.03.06
aggregate value	עֵרֶךְ מִצְרָף	15.03.07
aggressive optimization	מִטּוּב תַּקִּיף	15.08.05
alias	כִּנּוּי	15.03.21
anonymous	עֲלוּם-שֵׁם	15.04.35
APM [abb]	מוֹדֵל תִּכְנוּת יְשׁוּם	15.00.10
application client	לָקוֹחַ יְשׁוּם	15.12.19
application interface	מְנָשֵׁק יְשׁוּם	15.12.38
Application Programming Model	מוֹדֵל תִּכְנוּת יְשׁוּם	15.00.10
archive library	סִפְרִיַּת אַרְכִּיוֹן	15.12.01
area	שֶׁטַח	15.02.32
array	מַעֲרָךְ	15.03.10
array sliceslice	פֶּלַח מַעֲרָךְ	15.03.11
array type	טִפּוּס מַעֲרָךְ	15.04.20
assignment	הַצָּבָה	15.05.04
assignment by name	סִמּוּךְ פָּרָמֶטְרִים לְפִי שֵׁם	15.06.18
assignment statement	מִשְׁפַּט הַצָּבָה	15.05.04
assumed-size aggregate	מִצְרָף בְּגֹדֶל מְשֹׁעָר	15.03.08
asynchronous procedure	נֹהַל אָסִינְכְּרוֹנִי	15.06.69
atomic operation	פְּעֻלָּה לֹא-פְּרִיקָה	15.08.16
atomic type	טִפּוּס לֹא-פָּרִיק	15.04.06
atomic value	עֵרֶךְ לֹא פָּרִיק	15.03.34
Atomicity, Consistency, Isolation, and Durability	אָטוֹמִיּוּת, עֲקִיבוּת, בִּדּוּד וְקַיָּמוּת (עֲמִידוּת)	15.08.20
automatic instantiation	אִדְגּוּם אוֹטוֹמָטִי	15.09.40
automatic type conversion	הֲמָרַת טִפּוּס מִשְׁתַּמֵּעַת	15.04.42
base class	מַחְלֶקֶת בָּסִיס	15.09.18
base type	טִפּוּס בָּסִיס	15.04.24
basic block	גּוּשׁ בָּסִיסִי	15.06.40
basic optimizations	מִטּוּב בָּסִיסִי	15.08.04

English Term	מונח עברי	מספר מונח
bean	תַּרְמִיל	15.06.50
binary comparison	הַשְׁוָאָה בִּינָרִית	15.08.10
block statement	מִשְׁפַּט גּוּשׁ	15.05.24
body	גּוּף	15.06.02
body stub	תּוֹתָב	15.06.16
Bool type	טִפּוּס לוֹגִי	15.04.07
Boolean expression	בִּטּוּי לוֹגִי	15.05.35
Boolean type	טִפּוּס לוֹגִי	15.04.07
bottom-driven loops	לוּלָאָה נִבְדֶּקֶת זָנָב	15.06.48
branch construct	צֶרֶף סִעוּף	15.05.45
break time	זְמַן הַפְסָקָה	15.12.15
bug	בָּג	15.00.14
built-in	מֻבְנֶה	15.02.05
button	לְחִיץ	15.12.44
call	קְרִיאָה	15.06.05
call [v]	קָרָא	15.06.06
call by address	קְרִיאָה לְפִי מַעַן	15.06.08
call by location	קְרִיאָה לְפִי מַעַן	15.06.08
call by name	קְרִיאָה לְפִי שֵׁם	15.06.07
call by reference	קְרִיאָה לְפִי מַעַן	15.06.08
call by value	קְרִיאָה לְפִי עֵרֶךְ	15.06.09
callback methods	שִׁיטוֹת מַעֲנֶה	15.08.25
caption	כִּתּוּבִית	15.12.45
case statement	מִשְׁפַּט "מִקְרֶה"	15.05.16
character constant	קָבוּעַ תָּו	15.01.12
character type	טִפּוּס תָּו	15.04.17
check box	תֵּבַת תִּיּוּג	15.12.28
checklist	רְשִׁימַת תִּיּוּג	15.12.29
class	מַחְלָקָה	15.09.11
class method	שִׁיטַת מַחְלָקָה	15.09.50
class template	תַּבְנִית מַחְלָקָה	15.09.17
closing	סְגִירָה	15.09.44
code module	פִּרְקַן קוֹד	15.06.46
code template	תַּבְנִית קוֹד	15.06.59
coercion	הֲמָרַת טִפּוּס מִשְׁתַּמֶּעֶת	15.04.42
combo box	תֵּבָה מְשֻׁלֶּבֶת	15.12.37
command button	לְחִיץ פְּקֻדָּה	15.12.46
comment	הֶעָרָה	15.01.11
commit	קִבּוּעַ	15.08.17
Common Object Request Broker Architecture	אַרְכִיטֶקְטוּרָה אֲחִידָה לִמְתַוֵּךְ שְׁמוֹת עֲצָמִים	15.03.30
component contract	חוֹזֶה רְכִיב	15.06.54
compound statement	מִשְׁפָּט מֻרְכָּב	15.05.03
concatenation	שִׁרְשׁוּר	15.00.04
conditional expression	בִּטּוּי תְּנַאי	15.05.14
conditional statement	מִשְׁפַּט תְּנַאי	15.05.13

English Term	מונח עברי	מספר מונח
connection	חִבּוּר	15.06.17
connector	מַחְבֵּר	15.06.53
consol tree	חַלּוֹנִית נִוּוּט	15.12.57
constant	קָבוּעַ	15.03.05
constraint	אִלּוּץ	15.04.25
constructor	בּוֹנֶה	15.09.47
container	מְכָל	15.06.51
context attribute	מְאַפְיֵן הֶקְשֵׁר	15.10.08
control	פַּקֵּד	15.12.26
control array	מַעֲרַךְ פַּקְּדִים	15.12.27
control flow	זְרִימַת בַּקָּרָה	15.08.02
control key	מַקֵּשׁ פִּקּוּד	15.12.47
CORBA [abb]	אַרְכִיטֶקְטוּרָה אֲחִידָה לִמְתַנְּדֵּ שְׁמוֹת עֲצָמִים	15.03.30
core class	מַחְלֶקֶת לִבָּה	15.09.21
coroutine	שִׁגְרַת הֶמְשֵׁךְ	15.06.04
critical section	קֶטַע קָרִיטִי	15.07.03
data attribute	תְּכוּנַת נְתוּנִים	15.03.19
data division	חֲטִיבַת הַנְּתוּנִים	15.02.02
data file	קוֹבֶץ נְתוּנִים	15.12.06
data flow	זְרִימַת נְתוּנִים	15.00.15
data member	נָתוּן חָבֵר	15.09.26
data object	עֶצֶם נְתוּנִים	15.03.02
data structure	מִבְנֵה נְתוּנִים	15.03.01
data type	טִפּוּס נְתוּנִים	15.04.01
data value	עֵרֶךְ נְתוּנִים	15.03.04
datatype	טִפּוּס נְתוּנִים	15.04.01
decision construct	צֶרֶף הַחְלָטָה	15.05.37
decision structure	צֶרֶף הַחְלָטָה	15.05.37
declaration	הַכְרָזָה	15.02.01
declarative part	אֵזוֹר הַהַכְרָזוֹת	15.02.02
declarative region	אֵזוֹר הַהַכְרָזוֹת	15.02.02
decrementing	הַפְחָתָה	15.08.14
default	בְּרִרַת מֶחְדָּל	15.02.03
default button	לְחִיץ בְּרִרַת מֶחְדָּל	15.12.48
delay statement	מִשְׁפַּט הַשְׁהָיָה	15.05.27
delegation	אֲצִילָה	15.09.14
delimiter	תּוֹחֵם	15.01.06
deployment descriptor	מְתָאֵר פְּרִיסָה	15.10.09
derived type	טִפּוּס נִגְזָר	15.04.29
destructor	הוֹרֵס	15.09.48
details pane	חַלּוֹנִית פְּרָטִים	15.12.58
direct instance	אֶדְגֶם יָשִׁיר	15.09.43
directed instantiation	אִדְגּוּם מֻנְחֶה	15.09.42
disabled	לֹא-מְאֻפְשָׁר	15.10.12
disambiguation	הֲפָגַת עֲמִימוּת	15.01.09

English Term	מונח עברי	מספר מונח
discrete type	טִפּוּס סוֹדֵר	15.04.12
discriminant]n [	מַבְחִין	15.03.14
distributed	מְבֻזָּר	15.00.06
distributed application	יִשּׁוּם מְבֻזָּר	15.12.11
do while statement	מִשְׁפַּט "בַּצַּע כָּל-עוֹד"	15.05.21
Document Object Model	מוֹדֵל עַצְמֵי מִסְמָךְ	15.00.11
Document Type Definition	הַגְדָּרַת טִפּוּס מִסְמָךְ	15.00.12
DOM [abb]	מוֹדֵל עַצְמֵי מִסְמָךְ	15.00.11
dot notation	סִמּוּן בִּנְקֻדָּה	15.00.05
double-precision type	טִפּוּס דַּיִּק-כָּפוּל	15.04.40
DTD [abb]	הַגְדָּרַת טִפּוּס מִסְמָךְ	15.00.12
dynamic	דִּינָמִי	15.02.14
dynamic array	מַעֲרָךְ דִּינָמִי	15.03.29
dynamic memory allocation	הַקְצָאַת זִכָּרוֹן דִּינָמִית	15.10.04
dynamic scope	תְּחוּם הַכְרָזָה דִּינָמִי	15.02.08
dynamic storage allocation	הַקְצָאַת זִכָּרוֹן דִּינָמִית	15.10.04
dynamic variable	מִשְׁתַּנֶּה דִּינָמִי	15.02.23
Easter egg	מִטְמָן	15.06.58
editing	עֲרִיכָה	15.07.08
empty string	מַחְרֹוֶת רֵיקָה	15.03.27
enabled	מְאֻפְשָׁר	15.10.11
enabling property	תְּכוּנַת אִפְּשׁוּר	15.10.10
encapsulate [v]	כָּמַס	15.09.02
encapsulated type	טִפּוּס כָּמוּס	15.04.03
encapsulation	כְּמִיסָה	15.09.03
entry	כְּנִיסָה	15.05.09
entry name	שֵׁם כְּנִיסָה	15.05.10
entry-call statement	מִשְׁפַּט קְרִיאַת כְּנִיסָה	15.05.26
enumerated type	טִפּוּס נִמְנֶה	15.04.15
enumeration type	טִפּוּס נִמְנֶה	15.04.15
environment description	תֵּאוּר הַסְּבִיבָה	15.02.31
evaluation	הַעֲרָכָה	15.08.28
event	אֵרוּעַ	15.11.01
event procedure	נֹהַל אֵרוּעַ	15.11.02
event-driven	מוּנַע אֵרוּעִים	15.11.03
event-driven programming	תִּכְנוּת מוּנַע אֵרוּעִים	15.11.04
executable content	תֹּכֶן בָּצִיעַ	15.06.57
executable file	קֹבֶץ בָּצִיעַ	15.12.07
execution sequence	סֵדֶר בִּצּוּעַ	15.08.01
exit statement	מִשְׁפַּט יְצִיאָה	15.05.05
explicit instantiation	אִדְּגּוּם מְפֹרָשׁ	15.09.41
explicit type casting	הֲמָרַת טִפּוּס מְפֹרֶשֶׁת	15.04.41
explicit type conversion	הֲמָרַת טִפּוּס מְפֹרֶשֶׁת	15.04.41
expression	בִּטּוּי	15.05.33
extensibility	מִתְרַחֲבוּת	15.10.05

English Term	מונח עברי	מספר מונח
Extensible Markup Language	שְׂפַת סִמּוּן מִתְרַחֶבֶת	15.00.09
extension	הַרְחָבָה	15.06.66
external	חִיצוֹנִי	15.02.12
external symbol	סֵמֶל חִיצוֹנִי	15.06.45
fixed-point type	טִפּוּס נְקֻדָּה קְבוּעָה	15.04.10
floating-point type	טִפּוּס נְקֻדָּה צָפָה	15.04.11
flowchart symbol	סֵמֶל לְתַרְשִׁים זְרִימָה	15.00.16
focus	מוֹקֵד	15.12.54
for-construct	צֶרֶף "בַּעֲבוּר"	15.05.20
forest	יַעַר	15.06.71
form	טֹפֶס	15.06.60
form module	פִּרְקַן טֹפֶס	15.06.61
formal parameter mode	מַצַּב פָּרָמֶטֶר פּוֹרְמָלִי	15.06.20
formal parameterdummy argument	פָּרָמֶטֶר פּוֹרְמָלִי	15.03.17
format	תַּסְדִּיר	15.04.36
form-level variable	מִשְׁתַּנֶּה בְּרָמַת טֹפֶס	15.06.62
friend function	פּוּנְקְצִיָּה יְדִידָה	15.09.31
function	פּוּנְקְצִיָּה	15.06.12
function call	קְרִיאָה לְפּוּנְקְצִיָּה	15.06.13
function template	תַּבְנִית פּוּנְקְצִיּוֹת	15.09.27
garbage collection	הַשְׁמָשָׁה	15.08.21
general procedure	נֹהַל כְּלָלִי	15.06.37
generic	יְסוֹד [שם תואר]	15.06.29
generic body	גּוּף יְסוֹד	15.06.31
generic declaration	הַכְרָזַת יְסוֹד	15.06.30
generic function	פּוּנְקְצִיַּת יְסוֹד	15.09.34
generic instance	אֶדְגַּם יְסוֹד	15.06.36
generic instantiation	אִדְגּוּם יְסוֹד	15.06.35
generic module	פִּרְקַן יְסוֹד	15.06.34
generic operation	פְּעֻלַּת יְסוֹד	15.06.32
generic package	חֲבִילָה גֶּנֶרִיתחֲבִילַת יְסוֹד	15.06.33
global	כְּלָלִי	15.02.11
global declaration	הַכְרָזָה כְּלָלִית	15.02.19
global variable	מִשְׁתַּנֶּה כְּלָלִי	15.02.25
goto statement	מִשְׁפַּט "עֲבוֹר"	15.05.11
header file	קֹבֶץ כּוֹתֶרֶת	15.12.05
help panel	מַצַּג עֶזְרָה	15.12.61
hide	הַסְתָּרָה	15.12.55
home handle	תֶּפֶס בַּיִת	15.06.55
host type	טִפּוּס בָּסִיס	15.04.24
identifier	מְזַהֶה	15.01.03
IDL [abb]	שְׂפַת הַגְדָּרַת מִנְשָׂק	15.00.08
if statement	מִשְׁפַּט "אִם"	15.05.15
image control	פַּקַּד תְּמוּנָה	15.12.30
immutable binding	כְּרִיכָה מְקֻבַּעַת	15.08.29

English Term	מונח עברי	מספר מונח
immutable object	עֶצֶם מְקֻבָּע	15.03.32
imperative statement	מִשְׁפַּט צִוּוּי	15.05.12
implementation defined	מֻגְדָּר-מִמּוּשׁ	15.02.28
implementation dependent	תְּלוּי-מִמּוּשׁ	15.02.29
implicit declaration	הַכְרָזָה מִשְׁתַּמַּעַת	15.02.04
implied decimal type	טִפּוּס נְקֻדָּה קְבוּעָה	15.04.10
implied type conversion	הֲמָרַת טִפּוּס מִשְׁתַּמַּעַת	15.04.42
incomplete type	טִפּוּס לֹא-מֻשְׁלָם	15.04.38
incrementing	הוֹסָפָה	15.08.13
index type	טִפּוּס צִיּוּן	15.04.13
indirect referencing	הַפְנָיָה עֲקִיפָה	15.10.02
information hiding	הַסְתָּרַת מֵידָע	15.09.01
inheritance	יְרִישָׁה	15.09.13
initialization parameter	פָּרָמֶטֶר אִתְחוּל	15.06.56
initialize [v]	אִתְחֵל	15.10.03
inline function	פּוּנְקְצִיָּה תְּחוּבָה	15.09.32
instantiated class	מַחְלָקָה מְאֻדְגֶּמֶת	15.09.20
instantiated function	פּוּנְקְצִיָּה מְאֻדְגֶּמֶת	15.09.33
integer type	טִפּוּס שָׁלֵם	15.04.14
interface definition language	שְׂפַת הַגְדָּרַת מִנְשָׁק	15.00.08
internal documentation	תִּעוּד פְּנִימִי	15.06.42
intrinsic	מֻבְנֶה	15.02.05
intrinsic constant	קָבוּעַ מֻבְנֶה	15.01.15
invalid data	נְתוּנִים לֹא-תְּקֵפִים	15.03.24
italic	תָּו נָטוּי	15.00.13
iteration construct	צֶרֶף חִזְרוּר	15.05.46
iteration statement	מִשְׁפַּט חִזְרוּר	15.05.17
keyboard shortcut	מַקָּשׁ קִצּוּר	15.12.43
keyword	מִלַּת מַפְתֵּחַ	15.12.56
label	תָּוִית	15.01.10
language construct	צֶרֶף-שָׂפָה	15.01.02
lexical	מִלּוֹנִי	15.01.01
lexical element	עֵרֶךְ מִלּוֹנִי	15.01.01
lexical token	עֵרֶךְ מִלּוֹנִי	15.01.01
lexical unit	עֵרֶךְ מִלּוֹנִי	15.01.01
lifetime	אֹרֶךְ חַיִּים	15.02.15
limited type	טִפּוּס מֻגְבָּל	15.04.27
link unit	יְחִידַת קִשּׁוּר	15.06.44
list box	תֵּבַת רְשִׁימָה	15.12.31
literal constant	קָבוּעַ סְמִין	15.01.16
load statement	מִשְׁפַּט טְעִינָה	15.05.43
local	מְקוֹמִי	15.02.10
local declaration	הַכְרָזָה מְקוֹמִית	15.02.20
local variable	מִשְׁתַּנֶּה מְקוֹמִי	15.02.24
logic error	שְׁגִיאָה לוֹגִית	15.08.23

English Term	מונח עברי	מספר מונח
logical operator	אוֹפֵרָטוֹר לוֹגִי	15.05.40
logical type	טִפּוּס לוֹגִי	15.04.07
loop construct	צֶרֶף חִזְרוּר	15.05.46
loop statement	מִשְׁפַּט חִזְרוּר	15.05.17
macro	מַקְרוֹ	15.06.21
macrocall	קְרִיאָה לְמַקְרוֹ	15.06.23
macrodefinition	הַגְדָּרַת מַקְרוֹ	15.06.24
macroinstruction	הוֹרָאַת מַקְרוֹ	15.06.22
main program	תָּכְנִית רָאשִׁית	15.07.01
manipulator	טַפְלָל	15.08.12
member function	פּוּנְקְצִיַּה חֲבֵרָה	15.09.30
member initialization list	רְשִׁימַת אִתְחוּל חֲבֵרִים	15.09.45
member-wise initialization	אִתְחוּל חָבֵר מֵחָבֵר	15.09.46
message	מֶסֶר	15.09.08
metaclass	מֵטָא-מַחְלָקָה	15.09.22
method	שִׁיטָה	15.09.10
method permission	הֶתֵּר שִׁיטָה	15.09.51
mixed mode [adj]	רַב-טִיפּוּסִים	15.05.34
mixed type [adj]	רַב-טִיפּוּסִים	15.05.34
modal form	טֹפֶס חוֹבָה	15.06.63
modeless form	טֹפֶס רְשׁוּת	15.06.64
module	פִּרְקָן, מוֹדוּל	15.06.01
monitor	מוֹנִיטוֹר, מַשְׁגִּוֹחַ	15.07.07
multiple inheritance	יְרִישָׁה מֵרַבִּים	15.09.53
name demangling	הֲסָרַת יִחוּד שֵׁמוֹת	15.08.09
name mangling	יִחוּד שֵׁמוֹת	15.08.08
name qualification qualification	שִׁיּוּךְ שֵׁם	15.03.20
named parameter association	סִמּוּךְ פָּרָמֶטְרִים לְפִי שֵׁם	15.06.18
namespace	מֶרְחַב שֵׁמוֹת	15.06.70
navigation pane	חַלּוֹנִית נִוּוּט	15.12.57
nested loops	לוּלָאָה מְקוֹנֶנֶת	15.06.49
null pointer	מַצְבִּיעַ רֵיק	15.03.23
null stringzero-length string	מַחְרֹזֶת רֵיקָה	15.03.27
numeric type	טִפּוּס מִסְפָּרִי	15.04.16
object	עֶצֶם	15.09.07
object box	תֵּבַת עֶצֶם	15.03.31
object browser	דַּפְדֵּפָן עֲצָמִים	15.12.39
Object Request Broker	מְתַוֵּךְ שְׁמוֹת עֲצָמִים	15.03.33
object-oriented	כִּוּוּן-עֲצָמִים	15.09.15
object-oriented event-driven programming	תִּכְנוּת כִּוּוּן-עֲצָמִים מוּנַע אֵרוּעִים	15.00.07
OOED programming	תִּכְנוּת כִּוּוּן-עֲצָמִים מוּנַע אֵרוּעִים	15.00.07
operator overload	עֲמִיסַת אוֹפֵרָטוֹר	15.05.39
operator precedence	קְדִימוּת אוֹפֶרָטוֹרִים	15.05.36
option button	לְחִיץ בְּרִירָה	15.12.33
option buttons group	קְבוּצַת לְחִיצֵי בְּרִירָה	15.12.32

English Term	מונח עברי	מספר מונח
ORB [abb]	מְתַוֵּךְ שְׁמוֹת עֲצָמִים	15.03.33
ordinal type	טִפּוּס סוֹדֵר	15.04.12
overloading	עֲמִיסַת מַשְׁמָעֻיּוֹת	15.01.08
package	חֲבִילָה	15.06.25
package declaration	הַכְרָזַת חֲבִילָה	15.06.26
panel	מַצָּג	15.12.60
parameter	פָּרָמֶטֶר	15.03.15
parameter association	סִמּוּךְ פָּרָמֶטְרִים	15.03.18
parameter profile	מִתְאַר פָּרָמֶטְרִים	15.02.30
parent type	טִפּוּס הוֹרֶה	15.04.28
pass [n]	הַעֲבָרָה	15.08.15
passivation	אִבְטוּל	15.12.13
password char	תַּו סִסְמָה	15.12.40
peer	עָמִית	15.12.20
peer entity	יֵשׁוּת עָמִית	15.12.20
perform statement	מִשְׁפַּט "בַּצַּע"	15.05.23
perform until statement	מִשְׁפַּט "בַּצַּע עַד-אֲשֶׁר"	15.05.22
perform while statement	מִשְׁפַּט repeat while	15.05.21
picture	תְּמוּנָה	15.04.37
pointer	מַצְבִּיעַ	15.03.22
pointer type	טִפּוּס מַצְבִּיעַ	15.04.19
polymorphism	רַב-צוּרָתִיּוּת	15.09.12
positional parameter association	סִמּוּךְ פָּרָמֶטְרִים לְפִי מָקוֹם	15.06.19
predefined	מֻבְנֶה	15.02.05
predefined identifier	מְזַהֶה מֻבְנֶה	15.01.04
predefined type	טִפּוּס מֻבְנֶה	15.04.33
private	פְּרָטִי	15.09.04
private access	גִּישָׁה פְּרָטִית	15.09.35
private part	חֵלֶק פְּרָטִי	15.06.28
private type	טִפּוּס פְּרָטִי	15.04.26
procedure call	קְרִיאָה לְנֹהַל	15.05.25
procedure-call statement	מִשְׁפַּט קְרִיאָה לְנֹהַל	15.05.25
process symbols	סִמְלֵי תַּהֲלִיךְ	15.12.41
profile-based optimization	מֵטוּב מְבֻסַּס-מִתְאָר	15.08.06
program code	קוֹד תָּכְנִית	15.06.41
program file	קֹבֶץ תָּכְנִית	15.12.08
program unit	פִּרְקָן, מוֹדוּל	15.06.01
property	תְּכוּנָה	15.00.02
protected	מוּגָן	15.09.06
protected access	גִּישָׁה מוּגֶנֶת	15.09.36
protected member	חָבֵר מוּגָן	15.09.38
protocol	פְּרוֹטוֹקוֹל	15.09.09
public	צִבּוּרִי	15.09.05
public access	גִּישָׁה צִבּוּרִית	15.09.37
public member	חָבֵר צִבּוּרִי	15.09.39

English Term	מונח עברי	מספר מונח
radio button	לְחִיץ בְּרִירָה	15.12.33
radio buttons group	קְבוּצַת לְחִיצֵי בְּרִירָה	15.12.32
raise statement	מִשְׁפַּט הַעֲלָאַת חָרִיג	15.05.29
range	טְוָח	15.04.08
real type	טִפּוּס מַמָּשִׁי	15.04.09
record type	טִפּוּס רְשׁוּמָה	15.04.21
reference	הַפְנָיָה	15.10.06
reference parameter	פָּרָמֶטֶר מַעַן	15.02.26
relational operator	אוֹפֵרָטוֹר יַחְסָה	15.05.41
remark	הֶעָרָה	15.01.11
remove method	שִׁיטַת הֲסָרָה	15.09.52
rendezvous	מִפְגָּשׁ	15.07.05
repeat until statement	מִשְׁפַּט until	15.05.22
repeat while statement	מִשְׁפַּט do while	15.05.21
repetition construct	צֶרֶף חִזְרוּר	15.05.46
repetition structure	צֶרֶף חִזְרוּר	15.05.46
reserved word	מִלָּה שְׁמוּרָה	15.01.05
resource manager	מְנַהֵל מַשְׁאַבִּים	15.12.12
return [v] [intransitive]	חָזַר	15.05.07
return [v] [transitive]	הֶחְזִיר	15.05.08
return statement	מִשְׁפַּט חֲזָרָה	15.05.06
rollback	הֲסָגָה	15.08.18
root base class	מַחְלֶקֶת בָּסִיס שׁוֹרֶשׁ	15.09.19
roundoff error	טָעוּת עִגּוּל	15.08.24
rules of precedence	כְּלָלֵי קְדִימוּת	15.00.03
runtime system	מַעֲרֶכֶת זְמַן-רִיצָה	15.12.17
sandbox	תֵּבַת חוֹל	15.06.52
scalar type	טִפּוּס סְקָלָרִי	15.04.04
scope	תְּחוּם הַכְרָזָה	15.02.06
scope of a declaration	תְּחוּם הַכְרָזָה	15.02.06
scrolling window	חַלּוֹן נִגְלָל	15.12.59
Secure Socket Layer protocol	פְּרוֹטוֹקוֹל שִׁכְבַת שְׁקָעִים מְאֻבְטַחַת	15.12.21
select statement	מִשְׁפַּט בְּרִירָה	15.05.31
selecting	בְּחִירָה	15.12.50
selection structure	צֶרֶף הַחְלָטָה	15.05.37
selective-wait statement	מִשְׁפַּט הַמְתָּנָה בּוֹרֶרֶת	15.05.32
semaphore	סֵמָפוֹר	15.07.06
sentinel	זָקִיף	15.01.17
separator	מַפְרִיד	15.01.07
sequential file	קוֹבֶץ סִדְרָתִי	15.12.09
sequential processing	עִבּוּד סִדְרָתִי	15.12.14
servlet	שֶׁרֶתִית	15.12.22
shared data	נְתוּנִים מְשֻׁתָּפִים	15.02.07
shared library	סִפְרִיָּה מְשֻׁתָּפֶת	15.12.02
shortcut key	מַקֵּשׁ קִצּוּר	15.12.43

English Term	מונח עברי	מספר מונח
show	הַצָּגָה	15.12.51
side effect	תּוֹצָא לְוָאי	15.08.03
Simple Object Access Protocol	פְּרוֹטוֹקוֹל גִּישָׁה פְּשׁוּטָה לַעֲצָמִים	15.12.23
simple statement	מִשְׁפָּט פָּשׁוּט	15.05.02
simple type	טִפּוּס פָּשׁוּט	15.04.05
single inheritance	יְרִישָׁה מִיָּחִיד	15.09.54
single-precision type	טִפּוּס דַּיִּק-יָחִיד	15.04.39
smart pointer	מַצְבִּיעַ חָכָם	15.05.42
SOAP [abb]	פְּרוֹטוֹקוֹל גִּישָׁה פְּשׁוּטָה לַעֲצָמִים	15.12.23
software pipelining	הַצְנָרַת תָּכְנָה	15.08.07
source file	קוֹבֶץ מָקוֹר	15.12.04
splash screen	מַצַּג פְּתִיחָה	15.12.42
SSL protocol	פְּרוֹטוֹקוֹל SSL	15.12.21
start/stop symbol	סֵמֶל הַתְּחָלָה אוֹ סִיּוּם	15.01.18
startup form	טֹפֶס הֶזְנֵק	15.06.65
statement	מִשְׁפָּט	15.05.01
static	סְטָטִי	15.02.13
static scope	תְּחוּם הַכְּרָזָה סְטָטִי	15.02.09
static variable	מִשְׁתַּנֶּה סְטָטִי	15.02.22
stream	זֶרֶם	15.08.22
string constant	קָבוּעַ מַחְרוֹזֶת	15.01.13
string type	טִפּוּס מַחְרוֹזֶת	15.04.18
strong typing	הַטְפָּסָה חֲזָקָה	15.04.31
subarray	תַּת-מַעֲרָךְ	15.03.28
subclass	תַּת-מַחְלָקָה	15.09.25
subprocedure	תַּת-נֹהַל	15.06.39
subprogram	תַּת-תָּכְנִית	15.06.03
subprogram call	קְרִיאָה לְתַת-תָּכְנִית	15.06.10
subroutine	תַּת-שִׁגְרָה	15.06.11
Subscript, index	צִיּוּן	15.10.01
substitution variable	מִשְׁתַּנֶּה מִתְעַדְכֵּן	15.02.33
substring	תַּת-מַחְרוֹזֶת	15.03.26
subtype	תַּת-טִפּוּס	15.04.23
subunit	תַּת-יְחִידָה	15.06.15
superclass	מחלקת-על	15.09.24
symbolic constant	קָבוּעַ סִמְלִי	15.01.19
system modal	הוֹדָעַת-מַעֲרֶכֶת מְחַיֶּבֶת	15.12.52
Tab order	סֵדֶר טַבְלָר	15.12.53
task	מְשִׂימָה	15.07.02
task synchronization	סִנְכְּרוּן מְשִׂימוֹת	15.07.04
template	אַרְגּוּמֵנְט תַּבְנִית	15.09.16
template argument	אַרְגּוּמֵנְט תַּבְנִית	15.09.16
template function	פּוּנְקְצִיַּת תַּבְנִית	15.09.28
text box	תֵּבַת תַּמְלִיל	15.12.34
textual comparison	הַשְׁוָאַת תַּמְלִיל	15.12.16

English Term	מונח עברי	מספר מונח
timer control	בַּקָּרָה עַל-יְדֵי קוֹצֵב זְמַן	15.08.27
timestamp	חוֹתֶמֶת זְמַן	15.12.10
tool bar	סַרְגֵּל כֵּלִים	15.12.36
tool bar	סַרְגֵּל כֵּלִים	15.12.62
toolbox	עֶרְכַּת כֵּלִים	15.12.35
tooltip	רֶמֶז צָץ	15.12.49
top-driven loops	לוּלָאָה נִבְדֶּקֶת רֹאשׁ	15.06.47
toplevel form	טֹפֶס עֶלְיוֹן	15.06.67
toplevel scope	תְּחוּם עֶלְיוֹן	15.06.68
transaction call	קְרִיאָה לְמִפְגָּשׁ	15.06.14
transaction isolation level	רָמַת בִּדּוּד תְּנוּעָה	15.08.19
translation unit	יְחִידַת תִּרְגּוּם	15.06.43
trigraph sequences	תְּלַת-תָּו	15.01.14
type casting	הֲמָרַת טִפּוּס	15.04.30
type conversion	הֲמָרַת טִפּוּס	15.04.30
unconditional statement	מִשְׁפָּט לֹא-מֻתְנֶה	15.05.12
undeclared variable	מִשְׁתַּנֶּה לֹא-מֻכְרָז	15.02.21
underlying type	טִפּוּס בָּסִיס	15.04.24
Unicode	אוּנִיקוֹד	15.12.18
uniform referencing	הַפְנָיָה אֲחִידָה	15.10.07
union	אִגּוּד	15.05.38
universal type	טִפּוּס סְתָמִי	15.04.34
unload statement	מִשְׁפַּט פְּרִיקָה	15.05.44
until statement	מִשְׁפַּט "בַּצַּע עַד-אֲשֶׁר"	15.05.22
until-construct	צֶרֶף "עַד-אֲשֶׁר"	15.05.19
user data constraint	אִלּוּץ נְתוּנֵי מִשְׁתַּמֵּשׁ	15.08.26
user-defined procedure	נֹהַל מֻגְדָּר-מִשְׁתַּמֵּשׁ	15.06.38
valid data	נְתוּנִים תְּקֵפִים	15.03.25
value parameter	פָּרָמֶטֶר עֵרֶךְ	15.02.27
variable	מִשְׁתַּנֶּה	15.03.03
variable declaration	הַכְרָזַת מִשְׁתַּנֶּה	15.02.18
variant part	חֵלֶק מִשְׁתַּנֶּה	15.03.12
variant record	רְשׁוּמָה מִשְׁתַּנֶּה	15.03.13
variant record type	טִפּוּס רְשׁוּמָה מִשְׁתַּנֶּה	15.04.22
variant type	טִפּוּס מִשְׁתַּנֶּה	15.04.43
visibility	נִרְאוּת	15.02.16
visibility region	אֵזוֹר נִרְאוּת	15.02.17
visible part	חֵלֶק נִרְאֶה	15.06.27
weak typing	הַטְפָּסָה חַלָּשָׁה	15.04.32
Web component	רְכִיב מַאֲרָג	15.12.25
Web services	שֵׁרוּתֵי מַאֲרָג	15.12.24
while-construct	צֶרֶף "כָּל-עוֹד"	15.05.18
world readable files	קְבָצִים קְרִיאִים לַכֹּל	15.12.03
wrapper	עוֹטָף	15.09.49
XML [abb]	שְׂפַת סִמּוּן מִתְרַחֶבֶת	15.00.09