



תרשים זרימה ותוכנית בשפה מובנית

- הקדמה ומילון מונחים ⇐
- תוכנית בתרשים זרימה
- תוכנית בשפה מובנית
- סיכום

הקדמה ומילון מונחים



בפרק זה נקבל מושג כיצד תוכנית מחשב עובדת. לא נשתמש בשום שפת תכנות, אבל נכיר את הרכיבים הבסיסיים ביותר שקיימים בכל שפת תכנות ועל בסיסם בנויות כל התוכנות בעולם.

מחשב לא יודע מה זה "אולי". מחשב גם לא יודע מה זה "בערך" או "כמעט". מחשב יודע דבר אחד בלבד: 0 או 1.

השפה שבה המחשב מדבר היא בינארית. כל החישובים הכי מורכבים שהמחשב שלכם מסוגל לעשות, כל הצבעים שהוא מציג לכם, כל האותיות והמספרים, כל התמונות והסרטונים, כל קבצי האודיו, כל מה שאתם מקבלים מהמחשב שלכם בכל אמצעי פלט כלשהו, בסופו של דבר מתורגם לבינארית. 0 או 1. לרוב 0 מסמן "לא", ו-1 מסמן "כן".

לאחר שהבנו את זה, אנחנו יכולים גם להבין שמחשב מסוגל לקבל החלטות אך ורק על בסיס של שאלות שהתשובה להן היא או "כן" או "לא". ולכן המחשב שואל כל הזמן שאלות, אם התשובה שהוא מקבל לשאלה ששאל היא "כן" הוא מבצע משהו מסוים, ואם התשובה היא "לא" הוא מבצע משהו אחר.

איך המחשב שומר נתונים? מה זה משתנים?

בתוך המחשב קיים זיכרון המחולק להמון תאים. כל תא מחולק לכמה תתי תאים שבכל אחד מהם אפשר לכתוב 0 או 1.

כאשר אנחנו מריצים תוכנה כלשהי על המחשב, היא מקבלת גישה לזיכרון ויכולה לקרוא או לכתוב נתונים בתאי הזיכרון השונים. כדי שיהיה למתכנת שבונה את התוכנה נוח, הוא יכול להגדיר משתנים. משתנה הוא למעשה שם כלשהו שהמתכנת שבנה את התוכנית החליט, והוא מגדיר אזור מסוים בזיכרון ששמור עבור המשתנה הזה. בכל פעם שנרצה לשנות את הערך ששמור בתא הזה בזיכרון, נוכל להציב ערך חדש במשתנה הזה ואוטומטית הערך של התא ישתנה. אם נרצה לקרוא את הערך שכתוב בתא הזה, נקרא למשתנה ונקבל את הערך. זאת אומרת שמשתנה הוא למעשה מצביע לכתובת כלשהי בזיכרון של המחשב.

הקדמה ומילון מונחים



איך המחשב מחליט איזו פעולה לעשות? מה זה תנאי "אם"?

התנאי הבסיסי ביותר שעליו מבוססת כל שפת תכנות וכן כל שאר רכיבי בקרת הזרימה שקיימים בשפות תכנות הוא התנאי "אם". התנאי בנוי באופן הבא:

```
if (condition) {  
    do something  
} else {  
    do something else  
}
```

אם (תנאי כלשהו של כן או לא) אז
פקודה א'
אחרת
פקודה ב'
סוף אם

בתוך התנאי אנחנו יכולים לשאול כל שאלה שאנחנו רוצים כמו לדוגמה: האם $1=2$ או האם המשתנה "כמות" גדול מ-5. בפקודה אנחנו יכולים לבצע כל פעולה שנרצה, לדוגמה: להציב ערך חדש במשתנה כלשהו, או לחבר כמה ערכים יחד ולהציב אותם במשתנה. כמובן שלא חייבים להשתמש ב"אחרת" אם אין צורך בכך.

- כאשר רוצים לבדוק שוויון בתוך התנאי "אם", מקובל לרשום `==` ולא `=`, כיוון שהסימון `=` הינו סימון המציין השמה של ערך כלשהו בתוך משתנה. לדוגמה, אם נרצה לבדוק האם המשתנה `x` שווה למשתנה `y` נרשום זאת כך: `if (x==y)`

הקדמה ומילון מונחים



מה קורה כאשר הערך שיש במשתנה מסוים יכול להיות מספר אפשרויות (נגיד כחול / לבן / סגול / אפור)? מה זה "בצע מקרה"?

למעשה כל בעיה יכולה להיפתר באמצעות תנאי "אם", או באמצעות שילוב של כמה תנאי "אם". אך בשפות תכנות רבות, בכדי להקל על המפתחים, קיים משפט הנקרא "בצע מקרה" היכול לחסוך קצת כתיבת קוד מיותרת במקרים שבהם למשתנה מסוים יכולים להיות יותר משני ערכים. מבנה המשפט נראה כך:

```
do case (color) {  
  blue: do something 1  
  red: do something 2  
  black: do something 3  
  green: do something 4  
}
```

בצע מקרה (שם המשתנה)
ערך א': פקודה א'
ערך ב': פקודה ב'
ערך ג': פקודה ג'
ערך ד': פקודה ד'
סוף בצע מקרה

מה עושים אם צריך לחזור על אותה פעולה מספר ידוע של פעמים? מה זו "לולאה נומרית"?

כאשר מעוניינים לבצע את אותה פעולה מספר ידוע של פעמים, אפשר להשתמש בלולאה נומרית. ברוב שפות התכנות לולאה זו נקראת לולאת for. מגדירים משתנה המאותחל למספר התחלתי כלשהו, נגיד 0 או 1, ואז נכנסים ללולאה, מבצעים את הפקודה שבתוכה ומעלים את הערך של המשתנה בכל חזרה של הלולאה עד שמגיעים למספר המקסימלי שהגדרנו, לולאת for נראית כך:

```
for (i = 0 ; i < 100 ; i++) {  
  do something  
}
```

בצע כל עוד (אינדקס = 0, אינדקס > 100, אינדקס ++)
פקודה
סוף בצע כל עוד

הקדמה ומילון מונחים



מה עושים אם צריך לחזור על אותה פעולה מספר לא ידוע של פעמים? מה זו "לולאה לוגית"?

כאשר מעוניינים לבצע את אותה פעולה מספר לא ידוע של פעמים, אפשר להשתמש בלולאה לוגית. ברוב שפות התכנות לולאה זו נקראת `while`. לולאה זו ממשיכה לרוץ ולבצע עוד ועוד חזרות כל עוד התנאי המוגדר בה מתקיים. אפשר להשתמש בלולאה זו בדיוק כמו בלולאה נומרית על ידי הגדרת משתנה אינדקס וקידומו בכל איטרציה (חזרה). לולאה זו יכולה בקלות להיכנס ללופ אינסופי של חזרות, ולכן חשוב לוודא שהתנאי שמוגדר בה מתישהו יהפוך לשקר כדי שהיא תפסיק לרוץ. לולאת `while` נראית כך:

```
while (condition) {  
    do something  
}
```

בצע כל עוד (תנאי)
פקודה
סוף בצע כל עוד

האם יש קיצורי דרך לביצוע פעולות מסוימות?

ברוב שפות התכנות, מקובל שכאשר מעוניינים להוסיף 1 למשתנה כלשהו, שתי דרכי הרישום הבאות זהות זו לזו:

שם המשתנה ++

שם המשתנה = שם המשתנה + 1

כמו כן, מקובל שכאשר מעוניינים להוסיף ערך של משתנה y למשתנה x , שתי דרכי הרישום הבאות זהות זו לזו:

$y =+ x$

$y + x = x$

הקדמה ומילון מונחים



אז איזה מושגים הכרנו לקראת הפרק הזה?

משתנה - מקום שאפשר לאחסן בו ולקרוא ממנו נתונים

תנאי אם - בדיקה האם תנאי כלשהו מתקיים וביצוע פעולה אם כן

בצע מקרה - תנאי אם משוכלל, חוסך עבודה כאשר הערך של המשתנה יכול להיות יותר מכן או לא

לולאה לוגית - לולאה הממשיכה להתבצע כל עוד תנאי לוגי כלשהו מתקיים

לולאה נומרית - לולאה המתבצעת מספר מדויק וידוע מראש של פעמים