

מטלה 1 – עקרונות שפות תכנות: חלק 1:

שאלה 1, חלק ראשון:

שפות אימפרטיביות (Imperative Languages):

רצף של פקודות שנקראים באופן סדרתי ורציף פקודה אחר פקודה.

שפות פרוצדוראליות (Procedural Languages):

בדומה לתכנות אימפרטיבי גם כאן יש רצף של פקודות שנקראות בזו אחר זו, רק שכאן יש את האפשרות לקפוץ בין קטעים בקוד, בהתאם לדרישות שונות (לדוגמא קלט, מצב מסויים של משתנה וכו') בזמן ריצה. תחביר זה מאפשר להגדיר קוד כפונקציה, כך שניתן לקרוא לה ממקומות שונים בקוד.

יתרונות ביחס לשפה אימפרטיבית:

- * מפשט את הקוד ומונע שכפול קוד.
- * הופך את הקוד לקריא ומובן יותר.
- * במידה ויש טעות בפעולה בקוד יהיה ניתן לתקן את הטעות במקום אחת כדי להשפיע על כל המופעים של הפעולה.

שפות פונקציונאליות (Functional Languages):

התוכנית היא אוסף של פונקציות, כך שניתן לקרוא לפונקציה מסוימת ברגע נתון ולבצע אותה (לחשב ערך מסוים, למצוא ערך מסוים וכו').

פונקציות בתבנית זאת הם גם כן ביטויים, קריאה לפונקציה היא חישוב של ביטוי. אפשר לשלוח פונקציה כפרמטר לפונקציה אחרת ואפשר להחזיר פונקציה כערך מפונקציה אחרת.

פונקציה בתבנית זאת מקבלת פרמטרים ומחזירה ערך, כלומר אין פעולת השמה, ובמובן הרחב יותר אין side-effects – שורת קוד שלא מחזירה ערך, למשל השמה.

פונקציות מסדר גבוה מאחר שגם פונקציה היא ביטוי, ניתן להעביר פונקציה כפרמטר לפרוצדורה אחרת, וכן להחזיר כערך מפרוצדורה פונקציה.

יתרונות ביחס לשפה פרוצדורלית:

- * עקביות- תמיד נחזיר את אותו הפלט עבור אותו הקלט המתאים.
- * אימות קוד
- * מקבול
- * הפשטה/עיצוב

שאלה 1, חלק שני:

(a)

1. `x.some(y)`
2. `<T>((x: T[], f: (y: T) => boolean) => boolean)`

(b)

1. `x => x.reduce((acc, cur) => acc + cur, 0)`
2. `(x: number[]) => (y: number)`

(c)

1. `(x, y) => x ? y[0] : y[1]`
2. `<T>(x: boolean, y: T[]) => T`

שאלה 1, חלק שלישי:

בעזרת קונספט זה, נוכל לבצע הפשטה שתפריד בין אופן השימוש בהליך לבין הפרטים כיצד ייושם ההליך במונחים של פרוצדורות פרימיטיביות יותר. התפיסה האנלוגית לנתונים מורכבים נקראת הפשטת נתונים. הפשטת נתונים היא מתודולוגיה המאפשרת לנו לבודד את אופן השימוש באובייקט נתונים מורכב מפרטי אופן בנייתו מאובייקטים נתונים פרימיטיביים יותר.

הרעיון הבסיסי של הפשטת נתונים הוא לבנות את התוכניות שאמורות להשתמש באובייקטים של נתונים מורכבים כך שהם יפעלו על "נתונים מופשטים". כלומר, על התוכניות שלנו להשתמש בנתונים באופן שלא יניחו הנחות לגבי הנתונים. שאינם נחוצים בהחלט לביצוע המשימה שעל הפרק.

יחד עם זאת, ייצוג נתונים "קונקרטי" מוגדר ללא תלות בתוכניות המשתמשות בנתונים. הממשק בין שני החלקים הללו של המערכת שלנו יהיה מערך נהלים, המיישמים את הנתונים המופשטים מבחינת הייצוג הקונקרטי.