

PPL - ASSIGNMENT 1

חלק 1 - שאלות תאוריות

1. (a) λ שומר אימפרייטיב - שומר האוריינטציה מרכז של פקודות. אופן ביצוע הפקודה הוא כי אחר כי.

zz שומר בזיכרון - תחביר השפה מאפשר להפחיר קוד בכוונתו, כך שניתן עקובו ממקורות שונים בקוד.

zzz שומר פונקציונליטיב - תוכנית היא כיטוי או סדרת כיטויים. אופן ביצוע התוכנית הוא חישוב של כיטויים, כלומר מציאת הערך שלו. פונקציות כיטויים וקריאה לפונקציה היא חישוב של כיטוי. כל פונקציה מקבלת פרמטרים ומחזירה ערך: אין פלטת השעה, ובאופן דומה יותר אין side effects.

(b) יתכונות של שומר בזיכרון על שומר אימפרייטיב:

- בקוד מסוים בזירה האפשרת חזרה על קטעים בו, מה שמנוע חזרתיות.
- בקוד קט יותר עקביה כאלו הוא מאורכן בתח תפסיכים בלתי שומר משאבים.

(c) יתכונות של שומר פונקציונליטיב על שומר בזיכרון:

- פונקציות תמיד יחזירו את אותו ערך עבור קטע נתון. לכן קט יותר עוזר את הפתעפות של תוכנית.
- ניתן עתה קוד שני בלעז פונקציות ע"י האברת פונקציה באמצעות פונקציה אחרת.

2. `function sumEven(numbersAsString:string[]):number {`

`return numbersAsString.map((num) => parseInt(num,10))`

`.filter((num) => num % 2 == 0)`

`.reduce((x,y) => x+y, 0);`

}

3. (a) $(x, y) \Rightarrow x.some(y)$

$\langle T \rangle (x: T[], y: (z: T) \Rightarrow \text{boolean}) \Rightarrow \text{boolean}$

(b) $x \Rightarrow x.reduce((acc, cur) \Rightarrow acc + cur, 0)$

$(x: \text{number}[]) \Rightarrow \text{number}$

(c) $(x, y) \Rightarrow x ? y[0] : y[1]$

$\langle T \rangle (x: \text{boolean}, y: T[]) \Rightarrow T$

(d) $(f, g) \Rightarrow x \Rightarrow f(g(x+1))$

$\langle T_1, T_2, T_3 \rangle (f: (y: T_2) \Rightarrow T_3, g: (x: T_1) \Rightarrow T_2) \Rightarrow (x: T_1) \Rightarrow T_3$

4. abstraction barrier - מחלק את תפקיד האתחול של שני: @ מאפיר את הכיסים ומכיל את האימון. @ משתמש בכיסים על מנת עתה תוכנית מורכבת יותר.