

עקרונות שפות תכנות עבודה 1: חלק 1

שאלה 1

סעיף a:

גישה אימפרטיבית: תוכנית זה רצף של פקודות, שורה אחרי שורה
גישה פרוצדורלית: היכולת להגדיר קוד כפרוצדורה, כך שניתן לקרוא ממקומות שונים מהקוד, כמו קריאה לפונקציות. ברגע שהגדרנו פונקציה, נוכל להפעיל אותה כפרוצדורה נפרדת.
גישה פונקציונאלית: התוכנית היא ביטוי, או סדרת ביטויים שצריך לחשב את הערך שלו / שלהם. כלומר מציאת הערך שלו / שלהם, ולא ביצוע של הפקודות. בגישה זו ניתן להעביר פונקציה כפרמטר לפונקציה אחרת. פונקציות הן גם כן ביטויים, וקריאה לפונקציה היא חישוב של ביטוי ואין side-effects מעבר לכך, לדוגמא אין שינוי של ה Heap.

סעיף b:

יתרונות הגישה הפרוצדורלית על פני הגישה האימפרטיבית:

היתרונות של פונקציות בעצם: קוד יותר קריא, ניתן להמנע מכתובה חוזרת של קוד, יותר אינטואיטיבי וקל לכתובה, ניתן להשתמש בפונקציות כקופסה שחורה.

סעיף c:

יתרונות הגישה הפונקציונאלית על פני הגישה הפרוצדורלית:

- אימות קוד: פונקציה תלויה רק בפרמטרים שלה ולא במשתנים ב Heap. מכיוון שהיא תלויה בדברים קבועים, כך יותר קל לאמת קוד ולבצע טסטים.
- מקבול: מכיוון שבגישה זו אין side-effects כמו שינוי של משתנים, יש לנו רק getters והתוכנית בטוחה במובן המקבילי.
- הפשטה/עיצוב: אפשרות יותר טובה לעצב את הקוד בצורה "גרעינית", כלומר כל פונקציה עושה משהו אחד.

שאלה 2

```
const getDiscountedProductAveragePrice = (inventory: Product[]) : number => {  
  const filtered = inventory.filter((product) => product.discounted).map(discountedProduct  
=> discountedProduct.price)  
  return filtered.length !== 0 ? filtered.reduce((sum : number, discountedPrice : number) =>  
sum += discountedPrice) / filtered.length : 0;  
}
```

שאלה 3

- $(T[], T \rightarrow \text{boolean}) \rightarrow \text{boolean}$
- $\text{number[]} \rightarrow \text{number}$
- $(\text{boolean}, T[]) \rightarrow T$
- $(\text{number} \rightarrow \text{number}, \text{number} \rightarrow \text{number}) \rightarrow (\text{number} \rightarrow \text{number})$

עוד ניסוח:

- $\langle T \rangle (x : T[], y : (\text{value} : T) \Rightarrow \text{boolean}) : \text{boolean} \Rightarrow x.\text{some}(y)$
- $(x : \text{number[]} : \text{number} \Rightarrow x.\text{reduce}((\text{acc} : \text{number}, \text{cur} : \text{number}) \Rightarrow \text{acc} + \text{cur}, 0)$
- $\langle T \rangle (x : \text{boolean}, y : T[]) : T \Rightarrow x ? y[0] : y[1]$
- $(f : (\text{num} : \text{number}) \Rightarrow \text{number}, g : (\text{num} : \text{number}) \Rightarrow \text{number}) : \text{number} \Rightarrow (x : \text{number}) : \text{number} \Rightarrow f(g(x+1))$