

PPL 2021 Assignment 1

1)

Explain the following programming paradigms:

(a) Imperative:

זוהי פרדיגמת תכנות שבה זרימת הקוד (Control Flow) מוכתבת ע"י הצהרה מפורשת על רצף של פקודות. הרצת התוכנית היא למעשה ביצוע של פקודות אלה זו אחרי זו. זוהי פרדיגמה נפוצה ביותר, דוגמאות לשפות המקיימות אותה ניתן למצוא בשפות מוכרות כמו C++, Java, Python.

(b) Procedural:

זוהי פרדיגמת תכנות שמבוססת על הפרדיגמה האימפרטיבית שבה הקוד מאורגן בצורה היררכית של קריאות לפרוצדורות של קוד. שפות הממשות פרדיגמה זו מאפשרות להגדיר קטעי קוד כפרוצדורה כך שניתן להשתמש בו במקומות שונים בקוד. דוגמאות לשפות המקיימות אותה ניתן למצוא בשפות מוכרות כמו C++, Java, Python.

(c) Functional:

זוהי פרדיגמת תכנות שמרחיבה את הפרדיגמה הפרוצדורלית ומבוססת על חישוב של פונקציות מקוננות שנמנעות משינוי במצב החיצוני (global state) של התוכנית ושינוי (mutation) של ערכים. בפרדיגמה זו נקודת המבט על הקוד כרצף של ביטויים במקום רצף של פקודות, והרצת התוכנית היא למעשה חישוב של ביטויים אלה (במקום ביצוע פקודות) ודומה מאוד לאופן שבו מסתכלים על ביטויים מתמטיים. בפרדיגמה זאת נהוג כי פונקציות יהיו "טהורות" כלומר מקבלות ערכים ומחזירות ערכים מבלי פעולות השמה ומבלי השפעות חיצוניות על התוכנית. דוגמאות לשפות שמכילות יישום של פרדיגמה זאת ניתן למצוא בשפות כמו Python, JS, Haskell, Scheme.

How does the procedural paradigm improve over the imperative paradigm? How does the functional paradigm improve over the procedural paradigm?

לפרדיגמה האימפרטיבית בפני עצמה ישנם הרבה חסרונות כמו חזרה על קוד, שינויים בערכים בודדים עלול לדרוש שינוי מרובה בקוד, וחוסר קריאות בשל חוסר ארגון של הקוד. הפרדיגמה הפרוצדורלית משפרת את הפרדיגמה האימפרטיבית ועוזרת להתמודד עם חסרונות אלה בכך שמאפשרת מחזור של קטעי קוד, ארגון הקוד לקטעים מצומצמים בעלי תפקיד ברור ומוגדר ובכך שיפור קריאות הקוד, תיקון של שגיאות בו והתחזוקה שלו.

הפרדיגמה הפונקציונלית משפרת את הפרדיגמה הפרוצדורלית ע"י שיפור תקינות הקוד בהרצה מקבילית, שיפור היכולת לאימות הקוד (בשל הדמיון בין הקוד לביטויים מתמטיים) והפשטה של הקוד בצורה שמאפשרת שיפור בארגון ותחזוקה של הקוד.

2) Write the most specific types for the following expressions:

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

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

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

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

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

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

3) Explain the concept of "abstraction barriers".

"חסמי הפשטה" הוא רעיון מרכזי מאוד בתכנות ובתכנון תוכניות. הרעיון הוא שלכול תוכנה צריך להיות הפרדה מופשטת לשני חלקים מרכזיים: הלקוח שמשתמש בתוכנה והמימוש שלה. הלקוח לא צריך לדעת איך התוכנה עובדת אלה עליו לסמוך שהיא עושה את תפקידה כנדרש. במימוש נדרשת הבנה מלאה של איך התוכנה עובדת ובחלק זה צריך להביא חוסר אימון בה, ולבחון אותה מנקודת הנחה זו. לעיתים ההפרדה בין השניים לא ברורה מאליו (לדוגמא מפתח שמשתמש בתוכנה בזמן הפיתוח) ועל כן חשוב לשמור על הפרדה זו לאורך כול תהליך הפיתוח והתחזוקה של התוכנה.