עבודה 3 – תארוטי

קוט אנטולי

324413756

6) עבור **פונקציה ללא שם** סמנו אילו מהטענות נכונות, הסבירו בקצרה לכל טענה ותנו דוגמא:

א) לא יכולה לקבל פונקציה כפרמטר.

הטענה הזאת היא היא **לא** נכונה ! , פונקציות אלו **כן** יכולות לקבל פונקציות כפרמטר לדוגמא כמו שעשינו בכיתה , פוקניות compose אשר משלבת 2 פונקציות

ב) יכולה לקבל פרמטרים רק מטיפוסים פרימיטיביים (**int**, **float**, **bool**).

טענה זו היא לא נכונה , פוקנציות אלו יכולות לקבל גם טיפוסים מסוגים שונים כמו רשימה כלשהי או פונקציוצ

ג) יכולה להכיל יותר משורת קוד אחת.

לא נכון , פונקיוצת אלו הן one expression לכן הל נעשה ב single return

ד) לא יכלה להחזיר אחד משלושת ערכים שונים, לפי התנאי המוגדר ב-**if**.

לא נכון , פוקנציות אלו יכולות לגיע למצב שהן מחזירות 3 ערכים שונים , ניתן להגדיר תנאים נוסיף של if בתוך ה else

לדגומא : lambda x: x if x==0 else True if x > 0 else False

ה) פונקציה אחת יכולה להחזיר אובייקטים מטיפוסים שונים לפי תנאי (**לדוגמא**: **True** או **3**).

נכון בהתאם לסעיף ד', יש if and else לכן ניתן להחליט מה להחזיר

ו) לא ניתן לשכתב **dispatch function** (למימוש טיפוס שלא ניתן לשנות, **immutable**) עם פונקציה ללא שם.

לא נכון , כן ניתן לממש dispatch function בעזרת פונקיוצות למבדה , בגלל הסעיף ד'

ז) רק פונקציה ללא שם ניתן להעביר כארגומנט לפונקציות מובנות כמו: **map**, **filter**, **reduce**.

לא נכון ! , ניתן להעביר גם פונקיות שהן לא אנונימיות , לדגומא כו שעשינו עם דוגמא של פיבונאצי , או isEven לפילטר

7) סמנו אילו מהטענות נכונות והסבירו בקצרה כל טענה:

א) בפייטון ניתן להגדיר שני משתנים שמצביעים לאותו אובייקט.

נכון , מכיוון שכל משתנה הוא אובייקט וניתן לקשר אותו לשם מסויים , בפייתון ניתן לקשר את האובייקט לכמה שמות

ב) בפייטון ניתן להגדיר שני משתנים שמצביעים לאותה פונקציה.

כמו בסעיף א', אותה דוגמא

ג) לפי שיטת **dispatch function** עם **message passing** פונקציה **dispatch** יכולה לקבל פונקציה כפרמטר.

טענה נכונה , ניתן לשלוח פונקציה ובגלל מודל הסביבות ניתן לשלוח אליה בפנים פרמטרים ולבצע מניפולציות

ד) לפי שיטת **dispatch function** עם **message passing** למימוש טיפוסי נתונים, ניתן ליצור רק טיפוסים שלא ניתן לשנות אותם (**immutable**).

לא נכון , ניתן ליצור גם טיפוסים כמו מילון או ליסט שהם טיפוסים שכן ניתנים לשונוי , ניתן לראות שבשאלה 5 ישנו טיפוס מסוג מילון

ה) מילון (**dictionary**) הוא מבנה גמיש לחלוטין, שאין שום מגבלה על הטיפוסים של אובייקטים שניתן להשתמש בתוכו.

לא נכון , ישנם הגבלות על מילון , ה key שלו חייב להיות immutable

ו) רשימה רקורסיבית (**rlist**) שמימשתם בכיתה (כפונקציה **dispatch function**) היא רצף (**sequence**).

כן מכיוון שזה מומש בצורת tuple ו פונקיוצות firsl and last מחזריות פונקציונליות של tuple