

1.

ישנם מספר מרחבי שונות לפרדיגמות תכנות, כאשר הפרדיגמות נבדלות על ידי נוהלי תכנות שהם מעודדים ואוסרים. הנה כמה ממדים שבהם הפרדיגמות נבדלות זו מזו:

*Syntax:

כמה טבעי, קצר, קריא הוא הביטוי של קוד בהינתן התחביר של השפה. חלק משפות התכנות קלות יותר לכתיבה וקריאה ביחס לשפות אחרות. לדוגמא SQL הרבה יותר קל ודומה במבנה שלה למשפטים בשפה האנגלית, וזאת לעומת שפת C++ שהרבה מוצאים כהרבה יותר מורכבת.

*Control flow:

זוהי צורה שבה תכנית רצה. יש כמה צורות לכך. יש קו זרימה שבו התכנית קוראת ברצף את כל שורות הקוד. לעומת זאת יש קוד הבנוי מקפיצות, היכול להוביל לכניסה לשורת קוד מסוימת או קפיצה מעליה, וזאת לדוגמא באמצעות שימוש בפונקציה if. יש תכנית הבנויה מכמה תרדים שרצים עליה במקביל. יש תכנית המריצה קוד מסוים אך ורק בתגובה ל"אירוע" מסוים. יש צורה הצהרתית, בה אנו מצהירים על מה שאנו מעוניינים לקבל, ללא בקשה מפורשת לגבי האופן שבה התכנית תפעל על מנת להחזיר את הערך המבוקש(דוגמא טובה לכך היא SQL).

*Performance:

גם הביצועים של התכנית משתנים בהתאם לפרדיגמה שבה אנו בוחרים להשתמש. בתכנות פרוצדורלי לדוגמא, בשל העובדה שאנחנו מיישמים אותה צעד אחר צעד במבנה של תכנית אחת המחולקת לתת תכניות רבות, אנו נמנעים מהרבה אופטימיזציות ביצועים. זאת מכיוון שניתן לבדוק כיצד כל תת תכנית של התכנית הגדולה עובדת, ובכך לעבוד בצורה יעילה ופשוטה לדיבוג. זאת לעומת פרדיגמות אחרות, שבהן שהקוד לא בנוי באותה צורה, וכתוצאה מכך הרבה פעמים נצטרך לבצע הרבה אופטימיזציות/שינויים על מנת שהכל יעבוד בצורה מסונכרנת ונכונה. דבר העלול לגבות זמן יקר בעבור המתכנת.

2.

ה-types של הפונקציות הבאות הן:

a. (x: number, y: number) => number

b. <T>(x: T[]) => T

c. (x: boolean, y: number) => number

Short-cut semantics הוא קונספט שבו אנו מגיעים לבדיקת האופרנד הבא, אך ורק אם תוצאת התכנית עדיין לא נקבעה על פי האופרנד הקודם. בנקודת הזמן הראשונה שיש תוצאה לתכנית, התכנית נעצרת ומוחזר הערך הנדרש. בין אם משתמשים ב-Short-cut semantics ובין אם לא, תוצאת התכנית תהייה זהה. לאחר ריצת התכנית קשה להבחין מתי משתמשים ב-Short-cut semantics ומתי לא, ולכן הרבה פעמים עושים זאת באמצעות הרצה של משהו שיחולל שגיאה מכוונת. בשיטה זו נדע כי אם התכנית הגיעה לנקודה של השגיאה מדובר בשימוש ב-non short-cut semantic.

דוגמא לפונקציות הממשות את הקונספט short-cut semantic הן מתודות every ו-some.

מתודת some עוצרת מיד כאשר מוצאת איזשהו איבר המקיים בחיוב את הפרדיקט המדובר, ומחזירה ערך true. זאת במקום לבצע בדיקה מיותרת על שאר האיברים כאשר הדבר כבר לא רלוונטי, משום שכבר מצאנו את התשובה שאנו צריכים.

בדומה לאותה שיטת פעולה, מתודת every עוצרת מיד כאשר מוצאת איזשהו איבר אשר לא מקיים בחיוב את הפרדיקט המדובר, ומחזירה ערך false.