

שפות תכנות – תרגיל מספר 5 חלק יבש

תאריך הגשה: 15.1.18

מגישים:

סהר כהן – 206824088

יובל נהון – 206866832



שאלה 1

1. RTTI (Run Time Type Information) הוא מידע שנשמר לצד כל ערך בזמן ריצה ומתאר את הטיפוס שלו. השימוש ב-RTTI הוא לצורך העתקה עמוקה, איסוף אשפה, סריאליזציה, ובשפות תכנות שהן dynamically typed גם לצורך בדיקת טיפוס בזמן ריצה.
2. כאשר נעשה שימוש ב-RTTI חייב להשמר לצד האובייקט/משתנה שהמתכנת יוצר גם "מתאר טיפוס" ושפת C דוגלת בעקרון שאומר שאין מחירים נסתרים לפעולות שמבצע המתכנת. לכן אין ב-C מנגנון RTTI.
3. מנגנון איסוף האשפה נעזר ב-RTTI על מנת לדעת כיצד בדיוק לשחרר את הזיכרון שעליו לשחרר. המידע שהמנגנון מספק הוא גודל הבלוק שעליו לשחרר וגם מידע על קיום מצביעים בתוך האובייקט למקרה שיש צורך לסמן אותם כפי שנדרש באלגוריתם mark & sweep.
4. יש שימוש במנגנון RTTI גם לצורך העתקה עמוקה למשל. על מנת להעתיק אובייקט בשלמתו יש צורך גם להעתיק את כל הערכים הישיגים ממנו. על מנת להעתיק את כל הערכים יש צורך לדעת את הטיפוס שלהם, וזה אפשרי אך ורק באמצעות מנגנון RTTI.

שאלה 2

1. הפקודות האטומיות ב-kotlin הן:
(א) השמה(=) - הפקודה הזו נחשבת אטומית מכיוון שאין לה ערך החזרה(בניגוד ל-C למשל).
(ב) Sequencers - כמו בכמעט כל שפת תכנות הם פקודות אטומיות.
(ג) הפונקציה הריקה.
2. Block command: {}
Conditional commands: if-else, when
Iterative commands: for, while
3. Return, break, continue, throw, exitProcess הם Sequencers בשפה.
4. בשפה קיימים מערכים סטטיים בלבד. בקריאה רגילה של תא במערך תזרק שגיאה מתאימה במקרה של חריגה אבל אם משתמשים בפונקציה getOrNull לא תזרק שגיאה אלא יוחזר NULL.
5. Kotlin משתמשת ב-GC של Java. ה-GC הזה פועל באמצעות אלגוריתם mark & sweep על סמך ה-references שקיימים.

שאלה 3

1. נסווג את מערכת הטיפוסים של bash לפי הקריטריונים הבאים:
Existence : כל המשתנים בשפה הם מאותו סוג: מערך של מחרוזות. לכן אין ב-BASH מערכת טיפוסים, כלומר השפה היא Untyped. (הערה: נשים לב שפעולות אריתמטיות על משתנים אפשריות רק כאשר המחרוזת היא מספר, אך מחרוזות כאלו אינן מהוות טיפוס נפרד. כלומר הסמנטיקה נקבעת לפי הערך ולא לפי הטיפוס).
Strength: מכיוון שאין מערכת טיפוסים (כל המשתנים מאותו טיפוס) אז אין צורך באכיפה, או בהמרות מסוימות לפני ביצוע פעולות כלשהן בין משתנים. התנהגות כזאת מכונה Weak Typing.
Time of checking: השפה מאופיינת ב-Dynamic Typing. כפי שהוזכר לעיל אין מערכת טיפוסים, לכן אין צורך לבדוק טיפוסיות של משתנים לפני ריצה. בזמן ריצה תיווצר שגיאה כאשר ננסה להפעיל פעולות אריתמטיות על מחרוזות שאינן מייצגות מספרים.
Type Equivalence: קיים רק טיפוס אחד לכן כל הערכים שקולים מבחינת הטיפוס.
Flexibility: כתוצאה מהעדר טיפוסיות למתכנת יש חופש פעולה רחב מאוד.
Implicit Typing - בהכרזה על משתנה אין צורך להכריז על הטיפוס שלו.
2. 1. הפקודות האטומיות בשפה הן: פקודת הצבה, ביטול הגדרה של משתנה, Sequencers והפקודה הריקה ":".

2. Block command: {}.
- Conditional command: if-else, case.
- Iterative commands: for, while.
3. Sequencers – continue, break, return, exit.
4. מערכים גמישים (flexible) – אין למערכים בשפה גודל קבוע וניתן להוסיף ולהסיר מהם איברים בצורה חופשית.
5. ב-bash אין GC.

3. נקודת ההתחלה של קטע קוד בשפת BASH היא הפקודה הראשונה בקובץ. באותו אופן, סוף התכנית מוגדר להיות הפקודה האחרונה בקובץ - אלו הם גבולות התכנית. משום שגבולות אלו מצויים בקובץ אחד ויחיד השפה עונה על ההגדרה של **שפה אוטרקית** - שפה שבה ניתן לדעת בדיוק איזה קוד יתבצע מהסתכלות בתכנית בלבד. כמו כן, אין צורך לייבא ספריות חיצוניות על מנת להשתמש בפונקציות קלט\פלט שונות אשר מובנות בשפה, כמו echo.