

טקסונומיה של משימות תכנות

A Taxonomy of Task Types in Computing

Matt Bower

המאמר השלם ניתן להורדה בכתובת:

<http://db.grinnell.edu/sigcse/iticse2008/Program/viewAcceptedProposal.asp?sessionType=paper&sessionNumber=30>

הוצג בכנס ITiCSE שהתקיים במדריד ספרד 2008

טקסונומיה היא סיווג שיטתי (בדרך כלל היררכי) של עצמים מאותו תחום. המאמר מציע טקסונומיה של סוגי משימות במדעי המחשב לפי 10 קטגוריות. הטקסונומיה מבוססת על ניתוח תכניות הלימודים במדעי המחשב, סקר ספרות, והתייעצות עם עמיתים.

הקדמה

הקהילה של הוראת מדעי המחשב הקדישה רבות להגדרת הידע הקונספטואלי שלומדים צריכים לרכוש. אבל בדרך כלל, הדגש בפרסומים האלה היה על התכנים ולא היה בו דגש מספיק על התהליך שהלומד עובר.

בספרות נעשו כבר מספר נסיונות להגדיר סוגי ידע שונים בהקשר לתכנות. למשל, אחת האבחנות היא בין ידע תכנותי (ידע בעל אופי הצהרתי. למשל, להיות מסוגל להסביר איך לולאה עובדת) לבין אסטרטגיות תכנות (הדרך שבה משתמשים בידע. למשל, שימוש נכון בלולאה בתכנית). חוקרים רבים מגדירים 3 רמות של ידע:

- ידע הצהרתי (declarative or factual knowledge) כמו התחביר הדרוש לכתובת לולאה.
- ידע פרוצדוראלי (procedural knowledge) למשל, איך להפעיל IDE לצורך קומפילציה. הידע הזה מורכב מפיסות של ידע הצהרתי וקשרים ביניהם.
- ידע מושגי (conceptual knowledge) המתייחס להבנה איך ניתן להשתמש במנגנון של לולאה כדי לתכנן פתרון לבעיה נתונה. הידע הזה הוא ברמה הגבוהה ביותר.

הטקסונומיה המוצעת במאמר לסוגי משימות כוללת את 10 הרמות הבאות:

1. משימות הצהרתיות (Declarative tasks)
2. משימות הבנה (Comprehension tasks)
3. משימות ניפוי (Debugging tasks)
4. משימות ניבוי (Prediction tasks)
5. משימות "ספק דוגמה" (Provide-an-example tasks)
6. משימות "ספק מודל" (Provide-a-model tasks)
7. משימות הערכה (Evaluate tasks)
8. משימות תכנון לפי דרישות (Meet a design specification tasks)
9. משימות פתור בעיה (Solve-a-problem tasks)
10. משימות רפלקציה (Self-reflect tasks)

1. משימות הצהרתיות

ידע הצהרתי הוא סטטי, בדרך כלל תלוי בקשר אחד לכל היותר בין פיסות ידע. הוא פועל ברמת הזיהוי והיזכרות. משימות כאלה הן ברמה הנמוכה ביותר.

דוגמאות:

- אמת או שקר: כדי לכלול את התו '\ (לכסן אחורי) במחרוזת צריך לשים לפניו לוכסן אחורי אחר.
- מהי שיטת accessor ?

2. משימות הבנה

משימות כאלה קשורות בדרך כלל לקטע קוד או פריט של ידע הצהרתי ומתן הסבר (שלם או חלקי). הן מחייבות את התלמיד ליצור פתרון המבוסס על מודל מנטלי של המושג או הסיטואציה.

דוגמאות:

- הסבר את ההבדל בין השלם 127 לבין המחרוזת "127".
- מדוע קשה יותר לבדוק שוויון בין מספרים מטיפוס נקודה צפה מאשר שוויון בין שלמים?

3. משימות ניפוי

המשימות האלה מחייבות את התלמיד לגלות שגיאות. משימות לניפוי שגיאות תחביריות מבוססות על ידע הצהרתי, ואילו משימות לניפוי שגיאות סמנטיות מבוססות יותר על הבנה מסוג משימות הבנה.

דוגמאות:

- מהן שגיאות התחביר בקטע קוד הבא

```
System.out.println("Hello);
```
- מהן שגיאות התחביר בתכנית הבאה

```
int i;  
int factorial = 1;  
for (i = 1; i<=5; i++);  
    {factorial = factorial * i;}  
System.out.println(i+"! = "+ factorial);
```

4. משימות ניבוי

חיוני לתכנות כי בלי יכולת לנבא את השפעת ההוראות לא ניתן לכתוב תכניות. ניבוי נכון מבוסס על הבנה נכונה וידע הצהרתי.

דוגמאות:

- מה מדפיסה השורה הבאה?

```
System.out.println("11 + 5" + 20);
```
- מה תהיה ההשפעה של החלפת 5 ב- i+1 בקוד הבא

```
public class TwoDtester  
{  
    public static void main(String[] args)
```

```

{
  int[][] steps = new int [4][];
  for (int i = 0; i<steps.length;i++)
  {
    steps[i] = new int[5];
    for (int j=0; j<steps[i].length; j++)
    {
      steps[i][j] = i+j;
      System.out.print(steps[i][j] + ",");
    }
  }
}

```

5. משימות "ספק דוגמה"

משימות יצירתיות, יכולות להיות הצהרתיות או הבנה

דוגמאות:

- תן דוגמה לשגיאה לוגית
- צור דוגמה מקורית שלך לבעית ה- dangling else

6. משימות "ספק מודל"

דוגמאות:

- הסבר מה קורה במחשב (מאחורי הקלעים) כאשר מריצים תכנית ג'אווה
- שרטט דיאגרמה שתדגים מה קורה בזכרון המחשב כאשר אתה יוצר משתנה וכאשר אתה מאתחל את המשתנה

7. משימות הערכה

דוגמאות:

- הערך את השיטה (פעולה) הבאה כגישה עקרונית להכנסת ערך לשדה username במחלקה User


```

public String getUsername ()
{
  System.out.println("The username is: "+ username);
  return username;
}

```
- מהם היתרונות והחסרונות לקיומם של טיפוסים בשפת תכנות?

8. משימות תכנון לפי דרישות

דוגמאות:

- כתוב תכנית המשתמשת בלולאת for כדי להדפיס את המספרים הזוגיים בין 100 ל-2 בסדר יורד.

כלומר 100, 98, 96 וכן הלאה

- תכנן מערכת שתקיים את הדרישות הבאות:
המערכת כוללת UnderGradStudents, Lecturers ו- PostGradStudents.
א. לכל אדם במערכת יש שם.
ב. המערכת מחזיקה גם את המידע הבא:
האם Lecturer הוא בדרגה בכירה 'S' או בדרגה רגילה 'N'
מספר הסטודנט של כל Student
מבנה התשלום של כל סטודנט ("Full Fee" for PostGradStudent, "HECS" for UnderGradStudent)
מחרוזת המציינת את ההכשרה הקודמת של כל PostGradStudent
האם UnderGradStudent רוצה להשתתף בתכנית הדרכה ('Y' או 'N')

9. משימות פתור בעיה

דוגמאות:

- מהו המספר הקטן ביותר שיש לו 50 גורמים ראשוניים שונים?
- דיאנה רוצה לבדוק האם הציונים של תלמידיה במבחן הם עקביים ומתאימים לציוניהם במשימות האחרות. תכנן מערכת שתסייע לדיאנה לבצע את הבדיקה.

10. משימות רפלקציה

דוגמאות:

- בצע רפלקציה על הדרך שבה ביצעת את המשימה האחרונה.
- בצע רפלקציה על הדרך שבה עבדת עם הסטודנטים האחרים כדי לבצע את המשימה.