

Matam Code Conventions

1 מבוא

Code Conventions (מוסכמות קוד) מוגדרים כאוסף כללים המנחים את סגנון כתיבת הקוד. כללים אילו אינם באים לשפר את איכות הקוד בצורה ישירה אך מקלים על הבנתו וקריאתו, בעיקר ע"י אנשים נוספים מלבד הכותבים המקוריים. שימוש במוסכמות לכתיבת הקוד הוא בעל היתרונות הבאים:

- מוסכמות בכתיבת הקוד מקלות על הבנת הקוד ע"י מתכנתים (המקוריים, ומתכנתים נוספים).
 - תחזוקת הקוד לוקחת עד 80% מכל העבודה על התוכנה, ולכן קריאות הקוד היא חשובה ביותר.
 - מוסכמות כתיבת קוד מקלות על מציאת משתנים ופונקציות וחיפושן.
- שימו לב, העמידה במוסכמות המובאות כאן היא חובה בתרגילי הבית בקורס ואי עמידה בכללים אלו תגרור הורדה של נקודות מהציון.

2 שמות משתנים ופונקציות

- ניתן לבחור באחד משני הסגנונות הבאים לשמות המשתנים, והפונקציות:
 1. אותיות קטנות ואות גדולה בתחילת כל מילה שאינה ראשונה.

```
int doStuff(int number) {
    int myNumberVariable = 0;
    return myNumberVariable + number;
}
```

2. אותיות קטנות בלבד, ותו underscore להפרדה בין מילים.

```
int do_stuff(int number) {
    int my_number_variable = 0;
    return my_number_variable+number;
}
```

- על סגנון זה להיות עקבי והוא אינו יכול להשתנות.

- שמות עבור macros וקבועים המוגדרים ב-define יהיו באותיות גדולות ועם underscore בין מילים. למשל:

```
#define MAX_LEN 120
```

- שמות המשתנים צריכים להיות ברורים ולעמוד בכללים הבאים:
 - ללא קיצורים - למשל movie (טוב) ולא mov (רע).
 - ללא השמטת vowels (תנועות) - למשל controller (טוב) ולא cntrlr (רע).
 - מותר להשתמש בקיצורים הבאים בלבד:
 - i, j, k - עבור משתני אינדקס.
 - ptr - עבור מצביעים.
 - n - עבור מסי' שלם.
 - min, max - עבור מינימום ומקסימום.
 - fd - עבור file descriptor.
 - len - עבור length.
 - str - עבור string.
 - arg - עבור argument.
 - temp, tmp - עבור temporary.

- שמות הפונקציות יהיו מנוסחים כפעלים

- למשל findMoviesRecommendedForClient (טוב) ולא moviesRecommendedForClient (רע).
- במקרה והפונקציה היא עבור מבנה/ADT ניתן לתת לה שם המתחיל בשם המבנה, למשל:

▪ dateIsValid, dateCreate ו-dateDestroy עבור טיפוס Date.

- שמות טיפוסים המוגדרים בעזרת typedef:

- יוגדרו ללא קיצורים ויתחילו באות גדולה (ואת גדולה בתחילת כל מילה נוספת בשם הטיפוס).
- שמות הטיפוסים יהיו ביחיד ולא ברבים (למשל Movie אבל לא Movies).
- דוגמאות:

```
typedef struct {
    ...
} MovieReview;

typedef int Length;

typedef enum { ... } ErrorCode;
```

3 מבנה הקוד

- חובה להשתמש בהזחות בקוד (indentation) לכל בלוק:

```
int doStuff(int n) {
    if (n>5) {
        return 0;
    }
    return 1;
}
```

- בכל שורה קוד תופיע רק פקודה (statement) אחת. אין להוסיף פקודות נוספות באותה שורה לאחר ה-";".
- אורך שורת קוד לא יעלה על 80 תווים.
- אורך פונקציה לא יעלה על 50 שורות, מסי' זה כולל שורות ריקות ושורות המכילות רק סוגריים מסולסלים.
- חובה להשתמש בסוגריים מסולסלים לאחר כל if, else, while, for, do while. גם אם יש אחריהם רק פקודה אחת.
- גודל ההזחה (גודל תו הטאב) צריך להיות 4 תווים או 8 תווים.

- יש לבחור אחת משתי האפשרויות הבאות לעיצוב הסוגריים המסולסלים במסמך:
- סוגריים פותחים מופיעים באותה השורה עם הפקודה וסוגריים סוגרים יכולים להופיע החל מהשורה הבאה.

```
int doSomething(int n) {
    if (n > 0) {
        return n;
    } else if (n < 0) {
        return -n;
    } else {
        return 0;
    }
}
```

- סוגריים פותחים מופיעים בשורה הבאה אחרי הפקודה וסוגריים סוגרים יכולים להופיע החל מהשורה אחרי זה.

```
int doSomething(int n)
{
    if (n > 0)
    {
        return n;
    }
    else if (n < 0)
    {
        return -n;
    }
    else
    {
        return 0;
    }
}
```

על סגנון זה להיות עקבי והוא אינו יכול להשתנות (בין פונקציות או בין קבצים).

4 שפה

את כל ההערות, שמות המשתנים, שמות הפונקציות ובו' יש לכתוב באנגלית בלבד! אין לכתוב הערות בעברית. אין לקרוא למשתנים בעזרת תעתיק, למשל "histabrut" במקום "probability".

5 מקורות / קריאה נוספת:

1. *Java code conventions of Sun* <http://java.sun.com/docs/codeconv/html/CodeConvTOC.doc.html>
2. *Code Complete, Steve McConnell*