

דרישות סגנון תכניות

שמות פונקציות ושמות משתנים

שם אמור לתאר היטב את המהות המיוצגת. אלו מייחסים לכך חשיבות מרובה. שם של פונקציה אמור לתאר את המשימה שהיא ממלאת ושם של משתנה אמור לתאר את המשמעות של הערך המוחזק בו. בדרך כלל שמות הפונקציות חשובים במיוחד. אין חובה לתת תמיד שם ארכויים, למשתנים המייצגים אינדקסים למשל רצוי לתת שמות כמו `z` או `j`. שמות המורכבים ממספר מיללים יעוזו ב-`_` או באותיות גדולות על מנת שייהי אפשר להבין ממה השם מורכב. מלבד המקרה של הפרדה בין מילים ומשתנים קבועים, השם יהיה מורכב רק מאותיות קטנות.

לדוגמה:

```
int isValidPlace() {...}
```

```
int is_valid_place() {...}
```

```
int countJumps;
```

```
int count_jumps;
```

שמות של משתנים קבועים יהיו באותיות גדולות בלבד:

```
#define BUFFERSIZE 70
```

שמות של שדות של מבנים יתחילה ב-`m` ו-`ko` תחתון. לדוגמה:

```
struct Stack {  
    int m_top;  
    int m_array[SIZE];  
};
```

שמות של טיפוסים

גם שם של טיפוס יש משמעות מרובה. שם כולל – בהירותמושגית. שם גרוע – ערפלמושגי. טיפוסים מייצגים מושגים והשמות שאנו בוחרים להם הם המונחים. בחירת מונחים טובות היא הכרחית בהירות מחשבתית.

שם של טיפוס שאתם מגדרים יתחיל תמיד באות גדולה ושאר החוקים לגבי הם כמו של שמות הפונקציות והמשתנים.

לדוגמה:

```
struct BigStack {  
    int m_array[2*SIZE];  
};
```

```
enum WeekDay {SUNDAY,MONDAY,TUESDAY};
```

תיעוד

כתיבת התיעוד צריכה להתבצע לפי "קריטריון גואה":

דמיינו שאחרי שאתם מסייםים לכתוב את הקוד ולתעד אותו אתם נועסים לעשר שנים לגואה שבחו. לאחר עשר שנים אתם חוזרים וצריכים להוציא מאפיינים חדשים לקוד שלכם. חישבו איזה העורות יכניסו אותן בקהלות זהורה לעניינים לאחר שכבר שכחتم הכל. התיעוד לא חייב להיות מסיבי. לעיתים חלקה טובה של המטלות לפונקציות בעלות שמות טובים מוריידה הרבה מה צריך בתיעוד. בדרך כלל כדאי לכתוב לכל פונקציה תיאור קצר וממצא לגבי תפקידה, הנתונים שהוא מקבלת, הנקודות העבודה שלה, הערכאים שהוא מחזירה ומצבים שהוא משמרת. אם הפונקציה מכילה אלגוריתם לא טריוויאלי, כדאי להסביר אותו. לרבות, בגין הפונקציה כדי להמעיט בתיעוד מכיוון שהוא מפיער לרצף הקריאה. למרות זאת, אם ישנה הערה מאיימת על מקום מסוים בגוף הפונקציה אז

כדי לכתוב אותה.

קובץ ה – `readme` (בקורס זה, תמיד באותיות קטנות) יוכל את פרטי הסטודנט ומידע כללי על התרגיל. אין צורך להזכיר מחדש על הגדרת התרגיל שניתנה לכם, אפשר לחשב אל מסמך התרגיל שנכתב על ידיו כחלק מההיעוד שלכם. בקובץ ה – `readme` טבעי לכתוב את תפקיד הקבצים השונים (אם הם לא הוגדרו במסמך התרגיל) וארכיטקטורה כללית. תיעוד של מבני נתונים ואלגוריתמים יופיע בדרך כלל בקבצים הרלוונטיים. בהחלט אפשרי שבתרגילים הראשונים, קובץ ה – `readme` יכיל רק את פרטי הסטודנט. אין צורך לכתוב את פרטי הסטודנט בשאר קבצי הקוד.

עימוד

יש להקפיד על עימוד עקבי, רצוי להשתמש בכלים אוטומטיים (כמו `emacs`).