

דרישות סגנון תכנות בסדנת C++

שמות פונקציות ושמות משתנים

שם אמור לתאר היטב את המהות המיוצגת. אנו 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 שמות ארוכים, למשתנים המייצגים אינדקסים למשל רצוי לתת שמות כמו i או j. שמות המורכבים ממספר מילים ייעזרו ב - '_' או באותיות גדולות על מנת שיהיה 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  
const int BUFFERSIZE 70
```

שמות שדות של מבנים או class-ים, יתחילו בקו תחתון או ב m- וקו תחתון. לדוגמה:

```
class Strack {  
    int _top;  
    int _array[SIZE];  
public:  
    int pop();  
};
```

או:

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

שמות של טיפוסים

גם לשם של טיפוס יש משמעות מרובה. שם קולע – בהירות מושגית. שם גרוע – ערפל מושגי. טיפוסים מייצגים מושגים והשמות שאנו בוחרים להם הם המונחים. בחירת מונחים טובה היא הכרחית לבהירות מחשבתית.

שם של טיפוס שאתם מגדירים יתחיל תמיד באות גדולה. שאר החוקים לגביו 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 מסיימים לכתוב את הקוד ולתעד אותו אתם נוסעים לעשר שנים לגואה שבהודו. לאחר עשר שנים אתם חוזרים וצריכים להוסיף מאפיינים חדשים לקוד שלכם. חישבו איזה הערות יכניסו אותכם בקלות חזרה לעניינים לאחר שכבר שכחתם הכל. התיעוד לא חייב להיות מסיבי. לפעמים חלוקה טובה של המטלות לפונקציות בעלות שמות טובים מורידה הרבה מהצורך בתיעוד. בדרך כלל כדאי לכתוב לכל פונקציה תיאור קצר וממצא לגבי תפקידה, הנתונים שהיא מקבלת, הנחות תחילת העבודה שלה, הערכים שהיא מחזירה ומצבים שהיא משמרת. אם הפונקציה מכילה אלגוריתם לא טריוויאלי, כדאי להסביר אותו. לרב, בגוף הפונקציה כדאי להמעיט בתיעוד מכיוון שהוא מפריע לרצף הקריאה. למרות זאת, אם ישנה הערה מאירת עיניים המתאימה למקום מסוים בגוף הפונקציה אז כדאי לכתוב אותה.

קובץ ה- `readme` (בקורס זה, תמיד באותיות קטנות) יכיל את פרטי הסטודנט ומידע כללי על התרגיל. אין צורך לחזור מחדש על הגדרת התרגיל שניתנה לכם, אפשר לחשוב אל מסמך התרגיל שנכתב על ידינו כחלק מהתיעוד שלכם. בקובץ ה- `readme` טבעי לכתוב את תפקדי הקבצים השונים (אם הם לא הוגדרו במסמך התרגיל) וארכיטקטורה כללית. תיעוד של מבני נתונים ואלגוריתמים יופיעו 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 בכלי אוטומטי (כמו `emacs`).