Normes bàsiques de programació PRO1: C3

Professorat de Programació 1

15 de setembre de 2020

1 Preàmbul

Aquestes normes són la continuació del document 'Normes bàsiques de programació PRO1: C2'.

2 Estructura d'un programa

Recorda que els programes tenen cinc apartats:

1. Inclusions i espai de noms
2. Definició de constants — pot ser buit
3. Definició de nous tipus (typedef i structs) — pot ser buit
4. Procediments — pot ser buit
5. Programa principal (`main`)

2.1 Inclusions

- `<algorithm>`: Si no es diu explícitament que no es pot fer servir `sort()` i es vol ordenar un vector els elements del qual no són bàsics, es pot fer fàcilment usant una funció de comparació que indiqui quan un element és més petit que un altre.

```
struct Soci {
    string nom;
    int edat;
    ...
};

// Volem ordenar els socis primer per edat de més vells a més
// joves, i en cas d'empat creixentment per nom.
// Per fer-ho, implementem aquesta funció:
bool comp(const Soci& a, const Soci& b) {
    if (a.edat != b.edat) return a.edat > b.edat;
    return a.nom < b.nom;
}
```

```
int main() {
    ...
    sort(v.begin(), v.end(), comp);
    ...
}
```

Fixeu-vos que la següent funció de comparació *no* és correcta, perquè en cas que `a` i `b` siguin iguals, indica que `a` hauria d'anar abans que `b`, cosa que és falsa:

```
bool comp(const Soci& a, const Soci& b) {
    if (a.edat != b.edat) return a.edat > b.edat;
    return a.nom <= b.nom;
}
```

Està prohibit:

- Usar cap altra operació de la llibreria `<algorithm>`.

2.2 Definició de nous tipus

Només definirem nous tipus de dades mitjançant `typedef` i `struct`.

```
typedef vector<int> Fila;

struct Soci {
    string nom;
    int edat;
    ...
};
```

Està prohibit:

- Usar tipus enumerats (`enum`).
- Usar arrays a l'estil de C (declarats amb `[]`).
- Usar `class` per definir nous tipus.

3 Noms de tipus de dades

Els noms dels tipus de dades estan format per lletres minúscules i majúscules. Igual que amb els noms de constants, variables i procediments, si un nom està format per diverses paraules aquestes se separen amb un '_'.

Seguirem la següent convenció:

- Cada paraula del nom d'un tipus de dades comença amb una lletra majúscula, seguida de lletres minúscules.

```
typedef vector<int> Fila;
typedef vector<Fila> Matriu;

struct Punt_Tridimensional {
    double x, y, z;
};
```

4 Estil

4.1 Línies en blanc

- Totes les definicions de tipus amb `typedef` s'escriuen, en general, sense cap separació.

Opció preferida:

```
typedef vector<int> Fila;
typedef vector<Fila> Matriu;
```

Opció a evitar:

```
typedef vector<int> Fila;

typedef vector<Fila> Matriu;
```

- Les definicions de tipus amb `struct` s'escriuen amb separació.

Correcte:

```
struct Soci {
    string nom;
    int edat;
};

struct Punt_Tridimensional {
    double x, y, z;
};
```

Penalitzable:

```
struct Soci {
    string nom;
    int edat;
};
struct Punt_Tridimensional {
    double x, y, z;
};
```

Referències

- http://www.chris-lott.org/resources/cstyle/CppCodingStandard.html
- http://www.research.att.com/~bs/JSF-AV-rules.pdf
- http://gcc.gnu.org/onlinedocs/libstdc++/17_intro/C++STYLE
- http://www.horstmann.com/bigcpp/styleguide.html
- http://www.spelman.edu/~anderson/teaching/resources/style/
- http://geosoft.no/development/cppstyle.html
- http://geosoft.no/development/cpppractice.html