**Car Rental System**

*Motivatie:*

Avem dorinta de a crea un sistem usor utilizabil pentru oameni ce are ca scop eficientizarea modului in care sunt inchiriate masinile.

*Structura:*

**Clasa pentru tot sistemul - impreuna**

* numele firmei – static
* **Clasa pentru locatia firmei - impreuna**
  + tara
  + oras
  + numar masini disponibile
  + nota firma
  + review firma
  + **Clasa pentru clienti - Emanuel**
    - nume
    - prenume
    - adresa de mail
    - numar telefon
    - CNP
    - seria CI
    - nr CI
    - **Clasa pentru inchiriere**
      * zi inceput
      * luna inceput
      * an inceput
      * zi sfarsit
      * luna sfarsit
      * an sfarsit
      * ora inceput
      * ora sfarsit (ora fixata [:00] sau ora jumate [:30] )
    - **Clasa pentru permis**
      * zi emitere
      * luna emitere
      * an emitere
      * zi expirare
      * luna expirare
      * an expirare
      * eligibl pentru condus (true/false)
    - **Clasa pentru adresa clientului**
      * tara
      * oras
      * cod postal
      * strada
      * numar
    - **Clasa pentru data nasterii**
      * zi
      * luna
      * an
  + **Clasa pentru masini - Alexandru**
    - marca
    - clasa
    - culoare
    - putere (BHP)
    - motor
    - usi
    - locuri
    - trasmisie (automata/manuala)
    - consum
    - valabilititate (daca in perioada respectiva mai sunt disponibile masini de genul acesteia)
    - pret/zi
    - pret garantie
    - avans
    - tip (economy/medium/premium)
    - **Clasa pentru review**
      * valoare comparativ cu pret
      * curatenie
      * comfort
      * starea masinii
      * nota overall
  + **Clasa pentru adresa locatiei - Alexandru**
    - strada
    - numar
    - cod postal
* **Clasa pentru contact**
  + numar telefon
  + adresa de mail
  + nume administrator

*Conventii:*

1. precedam numele de atribute cu “m\_”
2. Indentarea codului

Pentru editarea codului se va folosi indentarea cu 4 spatii, fara TAB-uri. In Visual Studio, setarile se fac astfel: la Tools — Options — Text Editor — C/C++ se seteaza: Tab size 4, Ident size 4, [x] Insert spaces.

1. Identiﬁcarea ﬁsierelor header

Fiecare ﬁsier .H va avea la inceput si sfarsit un #ifndef / #deﬁne / #endif cu un macro unic sau #pragma once, cu numele ﬁsierului. De exemplu, pentru un ﬁsier help\_tools.h vom folosi:

#ifndef \_HELP\_TOOLS\_H\_

#define \_HELP\_TOOLS\_H\_

...

#endif // \_HELP\_TOOLS\_H\_

sau

#pragma once

...

1. Denumirea parametrilor si numelelor functiilor se face folosind CapitalCase. Exemplu:

int MyFunction( int FirstParameter );

1. Denumirea variabilelor locale

Denumirea variabilelor locale se face folosind almostCapitalCase. De exemplu:

int retStatus;

CC\_LIST bufferLength;

Identiﬁcatorii dintr-un bloc de cod nu au voie sa aiba nume identic cu identiﬁcatori din blocurile ”parinte”, pentru a nu-i ascunde pe acestia.

1. Denumirea variabilelor globale

Denumirea variabilelor globale se face folosind almostCapitalCase iar numele va fi prefixat cu „g”. De exemplu:

int gCcVector;

CC\_LIST gList;

Identiﬁcatorii dintr-o functie nu au voie sa aiba nume identic cu identiﬁcatori globali, pentru a nu-i ascunde pe acestia.

1. Descrierea blocurilor de cod, utilizarea si pozitionarea acoladelor

Fiecare for, while, if, do, switch va ﬁ urmat de { }, chiar daca exista o singura instructiune in bloc. Blocul {} se scrie la acelasi nivel de identare cu instructiunea, iar continutul blocului, indentat cu un nivel. In cazul if-ului, else-ul se scrie la acelasi nivel de identare cu if-ul. De exemplu:

for (i = 0; i < 100; i++)

{

...

}

if (a > b)

{

while (x < 5)

{

x++;

}

// ...

}

else

{

...

}

1. Semniﬁcatia numelor de variabile locale, functii, structuri etc

Parametrii, variabilele locale, functiile, structurile etc. trebuie sa aiba nume descriptive (si nu x, cucu, aB, myList, etc.). Singura exceptie este cazul variabilelor de ciclu / indecsi, dar si asta doar in cazul unor cicluri / indexari simple, evidente.