

Problema Master mind. Extracte de solució

Josep M. Ribó

Juny, 2005

(Problema de l'examen de MTP-gestió de Setembre-2003)

1 Fitxer *combinacio.txt*

Advertiment: Es presenta només un extracte de l'especificació de la classe. A l'especificació completa han d'aparèixer **Precondicions, postcondicions i forma de la crida** per cada operació

Classe: `Combinacio`

Objectiu: Modelitza una combinació de `n` caràcters ($n \leq N=10$)

Operacions:

- `Combinacio();`
Genera una combinació de 4 caràcters triats aleatòriament
- `Combinacio(unsigned int nc, bool& er);`
Genera una combinació de `nc` caràcters triats aleatòriament.
Error si `nc > N`
- `Combinacio(char car[], unsigned int n, bool& er);`
Genera una combinació amb els `nc` caràcters de l'array `car`.
Error si `nc > N`
- `void setCombinacio(char car[], unsigned int n, bool& er);`
- `void setCombinacio(unsigned int n, bool& er);`

Les dues operacions anteriors tenen el mateix sentit que les constructores corresponents, però es poden aplicar explícitament sobre objectes de la classe `Combinacio` ja existents (recordem que les operacions constructores només es poden aplicar implícitament en el moment de crear un objecte). Aquestes operacions `setCombinacio(...)` seran usades per la classe `Director`, com veurem.

- void comprovarCombinacio(Combinacio& combi, int& npos, int& ncar);
Crida: `combiref.comprovarCombinacio(combi,npos,ncar);`
Indica el nombre d'encerts de posició (`npos`) i de caràcter (`ncar`) que hi ha a `combi` en relació a `combiref`
- char getCaracter(unsigned int pos, bool& er);
Retorna el caràcter que ocupa la posició `pos` de la combinació
Error si `pos` és més gran que `getNCaracters()`
- void setCaracter(unsigned int pos, char car, bool& er);
.....
- unsigned int getNCaracters();
Retorna el nombre de caràcters de la combinació
- void presentarCombinacio();
Escriu per la sortida estàndar el contingut de la combinació
Aquesta operació no és estrictament necessària i hi ha autors que ens són contraris

2 Fitxer *combinacio.h*

```

const unsigned int N=10;

class Combinacio
{
private:

    char c[N];
    unsigned int nc;

    bool encertCaracter(char car, bool usats[], Combinacio& combi);

public:
    Combinacio();
    Combinacio(unsigned int nc, bool& er);
    Combinacio(char car[], unsigned int n, bool& er);

    void setCombinacio(char car[], unsigned int n, bool& er);
    void setCombinacio(unsigned int n, bool& er);

    void comprovarCombinacio(Combinacio& combi,
                             int& npos, int& ncar);
    char getCaracter(unsigned int pos, bool& er);
    void setCaracter(unsigned int pos, char car, bool& er);
    unsigned int getNCaracters();
    void presentarCombinacio();
};

```

3 Fitxer *combinacio.cc*

```
#include "combinacio.h"
#include <iostream>

using namespace std;

Combinacio::Combinacio()
{
    //.....EXERCICI
}

Combinacio::Combinacio(unsigned int n, bool& er)
{
    int i;
    if (n>=N) er=true;
    else{
        for (i=0;i<n;i++){
            c[i]=rand()%n+'a';
        }
        nc=n;
        er=false;
    }
}

Combinacio::Combinacio(char car[], unsigned int n, bool& er)
{
    int i;
    if (n>=N) er=true;
    else{
        for (i=0;i<n;i++){
            c[i]=car[i];
        }
        nc=n;
        er=false;
    }
}

void Combinacio::setCombinacio(unsigned int n, bool& er)
{
    int i;
    if (n>=N) er=true;
    else{
        for (i=0;i<n;i++){
            c[i]=rand()%n+'a';
        }
        nc=n;
        er=false;
    }
}
```

```

void Combinacio::setCombinacio(char car[], unsigned int n, bool& er)
{
    int i;
    if (n>=N) er=true;
    else{
        for (i=0;i<n;i++){
            c[i]=car[i];

        }
        nc=n;
        er=false;
    }
}

```

```

void Combinacio::comprovarCombinacio(Combinacio& combi,
                                     int& npos, int& ncar)

```

```

{
    int i;
    bool usats[N];

    npos=0;
    ncar=0;

    for (i=0;i<nc;i++){
        usats[i]=false;
    }

    for (i=0;i<nc;i++){

        if (combi.c[i]==c[i]) {
            npos=np+1;
            usats[i]=true;
        }
        else if (encertCaracter(combi.c[i],usats,combi)){
            ncar=ncar+1;
        }
    }
}

```

```

void Combinacio::presentarCombinacio()

```

```

{
    int i;

    for (i=0;i<nc;i++){

        cout<<c[i]<<" ";
    }
}

```

```

/*****

```

OPERACIO PRIVADA:

```
*****/

bool Combinacio::encertCaracter(char car, bool usats[], Combinacio& combi)
{
    int i;

    i=0;
    while (i<nc && (car!=c[i] || usats[i] || c[i]==combi.c[i])){
        i++;
    }

    if (i<nc) {usats[i]=true; return true;}
    else return false;

}
```

Comentaris:

- Per implementar aquesta operacio usem l'array `usats`:
`usats[i]= cert` si el caracter que ocupa la posicio `i` de la combinacio de referencia ha sigut usat per comptar un encert de posicio o de caracter
La versió que us proposo en aquest document:

- Conté un sol bucle per comptar simultàniament els encerts de posició i de caràcter i
- **No** usa un array adicional `ep` per saber si un caràcter de la combinació ha sigut ja comptat com a encert de posició (en la solució que vam donar a classe sí que l'usàvem).

El resultat és, potser, una mica més difícil d'entendre, però una mica més eficient.

- Noteu que `encertCaracter(car,usats,combi)` retorna `cert` si trobem a la combinació de referència (o sigui, a l'objecte `Combinació` sobre el qual es crida a `encertCaracter`) un caràcter `c[i]` tal que:
 - `c[i]=car`,
 - `c[i]` no usat prèviament (`!usats[i]`) i
 - `c[i]` no és encert de posició (`c[i]!=combi.c[i]`)

4 Fitxer *director.txt*

Advertiment: Només he inclòs les capceleres de les operacions. Caldria fer-ne una especificació completa.

- `Director();`
- `void iniciJoc(unsigned int ncarcombi, bool& er);`

- void entraCombinacio(Combinacio& combi, int& npos, int& ncar, bool& err);
- void getICombinacio(unsigned int i, Combinacio& combi, bool& err);
- void getCombinacioPensada(Combinacio& combi);

5 Fitxer *director.h*

```
#include "director.h"

const unsigned int MAXCOMBI=100;

class Director{
private:

    Combinacio combiref;

    Combinacio registrecombi[MAXCOMBI];
    unsigned int ncombi;

public:
    Director();
    void iniciJoc(unsigned int ncarcombi, bool& er);
    void entraCombinacio(Combinacio& combi, int& npos, int& ncar, bool& err);
    void getICombinacio(unsigned int i, Combinacio& combi, bool& err);
    void getCombinacioPensada(Combinacio& combi);

};
```

6 Fitxer *director.cc*

```
Director::Director()
{
    bool er=false;
    char comb[4]={'a','b','c','d'};

    ncombi=0;
    combiref.setCombinacio(comb,4,er);
}

void Director::iniciJoc(unsigned int ncarcombi, bool& er)
{
    ncombi=0;
    combiref.setCombinacio(ncarcombi,er);
}

void Director::entraCombinacio(Combinacio& combi,
                                int& npos, int& ncar,
                                bool& err)
{
```

```

if (ncombi<MAXCOMBI){

    err=false;
    registrecombi[ncombi]=combi;
    //C++ permet assignar objectes de classes creades
    //pel programador.
    //Per tenir un major control de com es fa aquesta
    //assignacio, podeu definir l'operacio:
    // void assigna(Combinacio& combi);
    //a la classe Combinacio (com ja vam fer a la classe
    // Cadena)

    ncombi=ncombi+1;

    combiref.comprovarCombinacio(combi,npos,ncar);
}
else err=true;
}

void Director::getICombinacio(unsigned int i, Combinacio& combi, bool& err)
{

    if (i>=ncombi) err=true;
    else{ err=false; combi=registrecombi[i];}

}

void Director::getCombinacioPensada(Combinacio& combi)
{

    combi=combiref;

}

```

7 Fitxer *usuari.cc*

```

#include "combinacio.h"
#include "director.h"
#include <iostream>

using namespace std;

void demanarCombinacioJugador(Combinacio& combi)
{

    //Demana al jugador que entri per l'entrada estandar una
    //combinacio de 4 characters

}

```

```

void informaJugadorEncerts(unsigned int npos, unsigned int ncar)
{
    //Informa al jugador del nombre d'encerts de la seva darrera
    //combinacio
}

int main()
{
    Director dir;
    Combinacio combi;
    bool er;
    unsigned int npos, ncar;

    dir.iniciJoc(4,er);

    do{

        demanarCombinacioJugador(combi)
        dir.entraCombinacio(combi,npos,ncar,er);
        informaJugadorEncerts(ncar,npos);
    }while(npos<4);
}

```