

# Definició de classes: La màquina de begudes

Josep Maria Ribó

Maig-2004

## 1 L'especificació de la classe

L'especificació (també anomenada interfície o contracte) de la classe està contingut al fitxer *maquina.txt*. Conté la informació següent:

### Nom de la classe:

Maquina

### Objectiu:

Modelitza una màquina de begudes amb retorn de canvi i selecció de *n* tipus diferents de begudes.

### Consideracions:

- Les begudes estan numerades des de 1 fins a 4 i estan codificades d'acord a la taula següent:

1	aigua (0.33 l.)
2	suc préssec (0.5 l.)
3	suc poma (0.33 l.)
4	suc taronja (0.33 l.)

- Les monedes acceptades són de 0.10, 0.20, 0.50 i 1 euros
- La màquina té una capacitat de 100 ampolles de cadascun dels productes.
- La màquina té una capacitat de 100 monedes de cada tipus.

### Operacions:

*Observació important: A les postcondicions de les operacions, la notació cometa (') acompanyant un identificador d'objecte (e.g., mb') fa referència al valor d'aquell objecte a l'inici de l'operació.*

- `void reset(float preu1, float preu2, float preu3, float preu4)`
  - **Crida:** `mb.reset(preu1,preu2,preu3,preu4);`
  - **Pre:** `preu1, preu2, preu3, preu4 >0`
  - **Post:** `mb` està inicialitzada amb els preus `preu1` (aigua), `preu2` (suc préssec), `preu3` (suc poma) i `preu4` (suc taronja). `mb` no conté cap beguda de cap tipus ni tampoc cap moneda.
- `bool reposarBegudes(int tipus, int quantitat)`

- **Crida:** `b=mb.reposarBegudes(tipus,quantitat);`
- **Pre:** S'ha fet un reset de `mb`
- **Post:** `mb` és la màquina que conté les mateixes begudes que `mb'` a les que s'han afegit `quantitat` ampolles del tipus `tipus` i `b=fals`  
 Si `tipus>4` o `tipus < 1` `mb=mb'` i `b=cert`  
 Si a `mb'` no caben les `quantitat` d'ampolles addicionals, `mb=mb'` i `b=cert`  
 Codificació dels tipus: 1 (aigua), 2 (suc préssec), 3 (suc poma), 4 (suc taronja)
- `bool reposarMonedes(int tipus, int quantitat)`
  - **Crida:** `b=mb.reposarMonedes(tipus,quantitat);`
  - **Pre:** S'ha fet un reset de `mb`
  - **Post:** `mb` és la màquina que conté les mateixes monedes que `mb'` a les que s'han afegit `quantitat` monedes del tipus `tipus` i `b=fals`  
 Si `tipus>4` o `tipus < 1` `mb=mb'` i `b=cert`  
 Si a `mb'` no caben les `quantitat` de monedes addicionals, `mb=mb'` i `b=cert`  
 Codificació dels valors: 1 (10 cent.), 2 (20 cent.), 3 (50 cent.), 4 (1.0 euros)
- `bool seleccionarBeguda(int tipus)`
  - **Crida:** `b=mb.seleccionarBeguda(tipus);`
  - **Pre:** S'ha fet un reset de `mb`
  - **Post:** S'ha seleccionat a la màquina `mb` el tipus de beguda `tipus` (com a pas inicial del procés de compra d'una beguda) i `b=fals`  
 Si `tipus>4` o `tipus < 1`, no hi ha cap beguda seleccionada i `b=cert`  
 Si a `mb'` s'havia pagat ja una selecció de beguda prèvia (però aquesta beguda encara no s'havia subministrat), no es fa la nova selecció i `b=cert`  
 Codificació dels tipus: 1 (aigua), 2 (suc préssec), 3 (suc poma), 4 (suc taronja)
- `bool pagar(int ndeu, int nvint, int ncinqu, int nun)`
  - **Crida:** `b=mb.pagar(ndeudeu,nvint,ncinqu,nun);`
  - **Pre:** S'ha fet un reset de `mb`
  - **Post:** S'ha fet el pagament de la beguda seleccionada prèviament amb `mb.seleccionarBeguda(tipus)`. El pagament es compon de `ndeudeu` monedes de 10 cèntims, `nvint` monedes de 20 cèntims, `ncinqu` monedes de 50 cèntims i `nun` monedes de 1 euro. `b=fals`  
 Si no s'havia seleccionat prèviament una beguda, no s'accepten les monedes i `b=cert`  
 Si les monedes no completen l'import de la beguda seleccionada, no s'accepten i `b=cert`  
 Si les monedes introduïdes no caben a `mb`, no s'accepten i `b=cert`
- `bool subministrarBegudaICanvi(int& ndeu, int& nvint, int& ncinqu, int& nun, int& tipus)`
  - **Crida:** `b=mb.subministrarBegudaICanvi( ndeudeu, nvint, ncinqu, nun, tipus)`
  - **Pre:** S'ha fet un reset de `mb`
  - **Post:** S'ha subministrat la beguda `tipus` i s'ha retornat el canvi (la diferència entre el preu de la beguda i la quantitat que l'usuari havia pagat). El canvi es retorna en forma de `ndeudeu` monedes de 10 cèntims, `nvint` de 20, `ncinqu` de 50 i `nun` de 1 euro. `b=fals`.  
 Si a la màquina `mb` no hi ha prou monedes per subministrar el canvi, `b=cert` i es retorna a l'usuari la quantitat que havia pagat. Aquest retorn es fa amb les quantitats de monedes `ndeudeu`, `nvint`, `ncinqu` i `nun`, dels valors respectius.  
 Si no s'havia seleccionat prèviament una beguda o bé si no s'havia pagat encara, `b=cert`.

- `void reiniciarSeleccio(int& ndeu, int& nvint, int& ncinqu, int& nun)`
  - **Crida:** `mb.reiniciarSeleccio( ndeu, nvint, ncinqu, nun)`
  - **Pre:** S'ha fet un reset de `mb`
  - **Post:** S'anul·la la selecció feta d'una beguda (si se n'havia fet cap) i es retorna la quantitat pagada (si s'havia pagat alguna cosa)

## 2 La implementació de la classe

La implementació de la classe es reparteix en dos fitxers:

- *maquina.h*

Conté la representació de la classe, a la part privada.

(A més a més, la part pública enumera les capceleres de les operacions que la classe ofereix als seus usuaris, la qual cosa, no forma part de la representació sinó de l'especificació de la classe).

- *maquina.cpp*

Conté la implementació de les operacions de la classe. Aquesta implementació es fa en termes de la representació proposada a *maquina.h* i ha de respectar el contracte presentat a *maquina.txt*.

### 2.1 Fitxer *maquina.h*

```
const int NTIPUSB=4;
const int NTIPUSM=4;
const int NBEGUDES=100;
const int NMONEDES=100;

class Maquina{

    int begudes[NTIPUSB+1]; //indexem de begudes[1]...begudes[4]
    int monedes[NTIPUSM+1];
    float preu[NTIPUSB+1];
    int begudaSel;
    float qtatPagada;

    bool distribuirQtat(double qtat,int& ndeu,int& nvint, //Op. privada
                       int& ncinqu,int& nun);
public:

    void reset(float preu1, float preu2, float preu3, float preu4);
    bool reposarBegudes(int tipus, int qtat);
    bool reposarMonedes(int tipus, int qtat);
    bool seleccionarBeguda(int tipus);
    bool pagar (int ndeu, int nvint, int ncinqu, int nuneuro);
    bool administrarBegudaICanvi(int& ndeu, int& nvint, int& ncinqu,
                                 int& nun, int& tipus);
    void reiniciarSeleccio(int& ndeu, int& nvint, int& ncinqu,
                           int& nun);

    float getPreu(int tipus);
    int getBegudaSel();
};
```

## 2.2 Fitxer *maquina.cpp*

```

#include "maquina.h"
#include <math.h>
#include <iostream>

void Maquina::reset(float preu1, float preu2, float preu3, float preu4)
{
    int i;
    preu[1]=preu1;
    preu[2]=preu2;
    preu[3]=preu3;
    preu[4]=preu4;

    for (i=1;i<=NTIPUSB;i++){
        begudes[i]=0;
    }
    for (i=1;i<=NTIPUSM;i++){
        monedes[i]=0;
    }
    begudaSel=0;
    qtatPagada=0.0;
}

bool Maquina::reposarBegudes(int tipus, int qtat)
{
    if (tipus<0||tipus>NTIPUSB) return true;
    if (begudes[tipus]+qtat>NBEGUDES) return true;
    else begudes[tipus]=begudes[tipus]+qtat;
    return false;
}

bool Maquina::reposarMonedes(int tipus, int qtat)
{
    if (tipus<0||tipus>NTIPUSM) return true;
    if (monedes[tipus]+qtat>NMONEDES) return true;
    else monedes[tipus]=monedes[tipus]+qtat;
    return false;
}

bool Maquina::seleccionarBeguda(int tipus)
{
    if (tipus<0||tipus>NTIPUSB|| qtatPagada>0.0) return true;
    if (begudes[tipus]==0) return true;
    begudaSel=tipus;
    return false;
}

bool Maquina::pagar (int ndeu, int nvint, int ncinqu, int nuneuro)
{
    if (begudaSel==0||qtatPagada>0.0) return true;
    if ((monedes[1]+ndeu)>NMONEDES||
        (monedes[2]+nvint)>NMONEDES||
        (monedes[3]+ncinqu)>NMONEDES||
        (monedes[4]+nuneuro)>NMONEDES) return true;
}

```

```

qtatPagada=ndeus*0.10+nvint*0.20+ncinqu*0.50+nuneuro;

if (qtatPagada<preu[begudaSel]) {
    qtatPagada=0.0;
    return true;
}
monedes[1]=monedes[1]+ndeus;
monedes[2]=monedes[2]+nvint;
monedes[3]=monedes[3]+ncinqu;
monedes[4]=monedes[4]+nuneuro;
return false;
}

void Maquina::reiniciarSeleccio(int& ndeu, int& nvint, int& ncinqu,
                                int& nun)
{
    bool ok;

    begudaSel=0;
    if (qtatPagada!=0.0){
        ok=distribuirQtat(qtatPagada,ndeus,nvint,ncinqu,nun);
    }
    qtatPagada=0.0;
}

bool Maquina::subministrarBegudaICanvi(int& ndeu, int& nvint, int& ncinqu,
                                        int& nun, int& tipus)
{
    bool ok;
    float canvi;

    if (qtatPagada==0.0) return true;

    canvi=qtatPagada-preu[begudaSel];
    ok=distribuirQtat(canvi,ndeus,nvint,ncinqu,nun);

    if (ok){
        tipus=begudaSel;
        monedes[1]=monedes[1]-ndeus;
        monedes[2]=monedes[2]-nvint;
        monedes[3]=monedes[3]-ncinqu;
        monedes[4]=monedes[4]-nun;
        begudes[begudaSel]--;
        qtatPagada=0.0;
        begudaSel=0;

        return false;
    }
    else{
        ok=distribuirQtat(qtatPagada,ndeus,nvint,ncinqu,nun);
        qtatPagada=0.0;
        begudaSel=0;
        return true;
    }
}

int Maquina::getBegudaSel(){return begudaSel;}

```

```

float Maquina::getPreu(int tipus)
{
    if (tipus<=0||tipus>NTIPUSB) return 0.0;
    return preu[tipus];
}

/*****

OPERACIONS PRIVADES
*****/
/*****

bool Maquina::distribuirQtat(double q,int& ndeu,int& nvint,
                             int& ncinqu,int& nun)

ok= mb.distribuirQtat(double q,int& ndeu,int& nvint,
                      int& ncinqu,int& nun)

Distribueix la quantitat q entre monedes de 0.10, 0.20, 0.50
i 1.0. Aquesta distribucio la col.loca als parametres ndeu
(nombre de monedes de 0.10), nvint (id. de 0.20), ncinqu (0.50)
i nun (1) i ok=cert

Si no es possible fer aquesta distribucio tenint en compte el
nombre de monedes de cada tipus que hi ha a la maquina, ok=fals
*****/

bool Maquina::distribuirQtat(double q,int& ndeu,int& nvint,
                             int& ncinqu,int& nun)
{
    int qtat=(int)ceil(q*100);

    nun=qtat/100;
    if (monedes[4]<nun) nun=monedes[4];
    qtat=qtat-nun*100;

    ncinqu=qtat/50;
    if (monedes[3]<ncinqu) ncinqu=monedes[3];
    qtat=qtat-ncinqu*50;

    nvint=qtat/20;
    if (monedes[2]<nvint) nvint=monedes[2];
    qtat=qtat-nvint*20;

    ndeu=qtat/10;
    if (monedes[1]<ndeu) ndeu=monedes[1];
    qtat=qtat-ndeu*10;

    if (qtat<10) return true;
    else return false;
}

```

**Observació:** Noteu l'existència d'una operació privada (`distribuirQtat`) que facilita la implementació de les operacions públiques i que evita la repetició de codi. Noteu també que aquesta operació privada es posa en un lloc especial del fitxer (precedit d'un títol que fa referència a aquest fet) i que la seva especificació també es posa al fitxer *maquina.cpp* i no al fitxer que conté l'especificació de la classe (*contracte.txt*).

### 3 Ús de la classe

El fitxer *usuari.cpp* conté un programa que simula el funcionament d'una màquina de begudes. Aquest programa:

- Es limita a utilitzar les operacions de la interfície de la classe **Maquina**
- Mai no accedeix a la part privada de la classe. Per tant, el programa és independent de la representació de la classe i no s'ha de preocupar com s'ha desenvolupat aquesta representació.

**Fitxer** *usuari.cpp*:

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

void menuPagament(int& ndeu, int& nvint, int& ncinqu, int& nun);
void informarUsuariBegudaICanvi(int ndeu,int nvint,int ncinqu,
                                int nun,int tipus);
void informarUsuariCancelSeleccio(int ndeu,int nvint,int ncinqu,
                                   int nun);
void informarUsuariCancelSeleccio(int ndeu,int nvint,int ncinqu,
                                   int nun);
void omplirMaquina(Maquina& mb);
int mostrarMenu();

int main()
{
    Maquina mb;
    int ndeu,nvint,ncinqu,nun, opcio, tipus;
    bool sortir,b;

    mb.reset(1.0,1.0,1.0,1.0);

    omplirMaquina(mb);

    sortir=false;
    do{

        opcio=mostrarMenu();
        b=false;

        if (opcio>=1 && opcio<=4){ b=mb.seleccionarBeguda(opcio);}
        else if (opcio==5) {
            menuPagament(ndeou,nvint,ncinqu,nun);
            b=mb.pagar(ndeou,nvint,ncinqu,nun);
        }
        else if (opcio==6) {
            b=mb.subministrarBegudaICanvi(ndeou,nvint,ncinqu,nun,tipus);
            if (!b) informarUsuariBegudaICanvi(ndeou,nvint,ncinqu,nun,tipus);
        }
        else if (opcio==7){
            mb.reiniciarSeleccio(ndeou,nvint,ncinqu,nun);
            informarUsuariCancelSeleccio(ndeou,nvint,ncinqu,nun);
        }
        else if (opcio==8){
```

```

        cout<<"\n\n\nEl preu es "<<mb.getPreu(mb.getBegudaSel())<<endl;
    else {sortir=true;}

    if (b) cout<<"\n\n\n*****Error en la gestio de l'opcio triada"<<endl;
}while(!sortir);

return 0;
}

void omplirMaquina(Maquina& mb)
{
    mb.reposarMonedes(1,10);
    mb.reposarMonedes(2,10);
    mb.reposarMonedes(3,10);
    mb.reposarMonedes(4,10);

    mb.reposarBegudes(1,10);
    mb.reposarBegudes(2,10);
    mb.reposarBegudes(3,10);
    mb.reposarBegudes(4,10);
}

int mostrarMenu()
{
    int opcio;

    cout<<"\n\n\n\n*****"<<endl;
    cout<<"MAQUINA BEGUDES"<<endl;
    cout<<"*****\n\n"<<endl;
    cout<<"1: Seleccionar aigua"<<endl;
    cout<<"2: Seleccionar suc taronja"<<endl;
    cout<<"3: Seleccionar suc pinya"<<endl;
    cout<<"4: Seleccionar suc pera"<<endl;
    cout<<"5: Pagar (cal seleccio previa)"<<endl;
    cout<<"6: Obtenir beguda i canvi (cal pagament previ)"<<endl;
    cout<<"7: iniciar de nou compra beguda"<<endl;
    cout<<"8: Preu de la beguda seleccionada"<<endl;
    cout<<"9: Sortir"<<endl;

    cin>>opcio;
    return opcio;
}

void menuPagament(int& ndeu, int& nvint, int& ncinqu, int& nun)
{
    bool error=false;

    do{
        cout<<"Entra el nombre de monedes de 0.10 que uses per pagar"<<endl;
        cin>>ndeudeu;
        cout<<"Entra el nombre de monedes de 0.20 que uses per pagar"<<endl;
        cin>>nvint;
        cout<<"Entra el nombre de monedes de 0.50 que uses per pagar"<<endl;
        cin>>ncinqu;
        cout<<"Entra el nombre de monedes de 1.0 que uses per pagar"<<endl;
        cin>>nun;
        if (ndeudeu<0||nvint<0||ncinqu<0||nun<0){error=true;}
    }
}

```



```

    }while(error);
    cout<<"\n\n\n";
}

void informarUsuariBegudaICanvi(int ndeu,int nvint,int ncinqu,
                                int nun,int tipus)
{
    cout<<"\n\n\n"<<endl;
    cout<<"Et subministro la beguda de tipus "<<tipus<<endl;
    cout<<"Et subministro com a canvi "<<ndeu<<" monedes de 0.10"<<endl;
    cout<<"Et subministro com a canvi "<<nvint<<" monedes de 0.20"<<endl;
    cout<<"Et subministro com a canvi "<<ncinqu<<" monedes de 0.50"<<endl;
    cout<<"Et subministro com a canvi "<<nun<<" monedes de 1.0"<<endl;
}

void informarUsuariCancelSeleccio(int ndeu,int nvint,int ncinqu,
                                    int nun)
{
    cout<<"\n\n\n"<<endl;
    cout<<"S'ha cancel.lat la seleccio que havies fet "<<endl;
    cout<<"Et retorno la quantitat de diners que havies posat:"<<endl;
    cout<<ndeu<<" monedes de 0.10"<<endl;
    cout<<nvint<<" monedes de 0.20"<<endl;
    cout<<ncinqu<<" monedes de 0.50"<<endl;
    cout<<nun<<" monedes de 1.0"<<endl;
}

```

## 4 Procés de compilació

```

g++ -c usuari.cpp
g++ -c maquina.cpp
g++ maquina.o usuari.o -o usuari

```

Execució amb: ./usuari

## 5 Problemes addicionals

1. A l'especificació no s'han especificat les operacions `getBegudaSel()` i `getPreu()`. Especifica-les.
2. Especifica i implementa altres operacions que es podrien aplicar a la classe `Maquina`. Per exemple: obtenir la quantitat pagada, obtenir el nombre d'ampolles o monedes d'un determinat tipus, buidar les monedes...
3. (\*\*) Ampliar la funcionalitat de la maquina de manera que es pugui:
  - Afegir mes tipus de beguda i que s'accedeixi a les begudes per nom,
  - Afegir diferents valors de monedes i diferent nombre de monedes acceptades, sense haver de retocar el codi cada cop que es faci això.