#include <iostream>

#include <cstring>

#include <cstdlib>

using namespace std;

#include "clases.h"

const char\* AR\_ReservasResto="reservasresto.dat";

void cargarCadena(char \*pal, int tam){

  int i;

  fflush(stdin);

  for(i=0;i<tam;i++){

      pal[i]=cin.get();

      if(pal[i]=='\n') break;

      }

  pal[i]='\0';

  fflush(stdin);

}

class ReservasResto{

    private:

    int \_codigoRestaurante;

    char \_nombre[30];

    int \_cantReserv;

    bool \_estado;

    public:

    void Mostrar();

    void Cargar();

    int leerDeDisco(int pos);

    bool grabarEnDisco();

    //GETTERS

    int getCodigoRestaurante(){return \_codigoRestaurante;}

    const char \*getNombre(){return \_nombre;}

    int getCantReserv(){return \_cantReserv;}

    bool getEstado(){return \_estado;}

    //SETTERS

    void setCodigoRestaurante(int cod){\_codigoRestaurante=cod;}

    void setNombre(const char \*nom){strcpy(\_nombre, nom);}

    void setCantReserv(int cant){\_cantReserv=cant;}

    void setEstado(bool e){\_estado=e;}

};

void ReservasResto::Cargar(){

    char texto[30];

    int aux;

    cout<<"NOMBRE: ";

    cargarCadena(texto, 29);

    setNombre(texto);

    cout<<"CODIGO: ";

    cin>>aux;

    setCodigoRestaurante(aux);

    cout<<"CANTIDAD RESERVAS: ";

    cin>>aux;

    setCantReserv(aux);

    setEstado(true);

}

void ReservasResto::Mostrar(){

    cout<<"NOMBRE: ";

    cout<<getNombre()<<endl;

    cout<<"CODIGO: ";

    cout<<getCodigoRestaurante()<<endl;

    cout<<"CANTIDAD RESERVAS: ";

    cout<<getCantReserv()<<endl;

}

/\*

devuelve -1 si no lo pudo abrir archivo

carga en el obj de la cls la info que leyo y devuelve 1

Si no pudo leer devuelve 0

\*/

int ReservasResto::leerDeDisco(int pos){

    FILE \*pReser;

    int leyo;

    pReser=fopen(AR\_ReservasResto,"rb");

    if (pReser==NULL){return -1;}

    fseek(pReser, sizeof(ReservasResto)\*pos,SEEK\_SET);

    leyo=fread(this ,sizeof(ReservasResto),1,pReser);

    fclose(pReser);

    return leyo;

}

/\*

devuelve falso si NO pudo grabar

y verdadero si pudo

\*/

bool Cliente::grabarEnDisco(){

    FILE \*pCli;

    bool escribio;

    pCli=fopen(AR\_ReservasResto,"ab");

    if (pCli==NULL){return false;}

    escribio=fwrite(this ,sizeof(ReservasResto),1,pCli);

    fclose(pCli);

    return escribio;

}

void Punto1(){

    ReservasResto nuevoObj;

    Restaurante resto;

    Reserva reserva;

    int posI=0;

    int posJ=0;

    int contRes=0;

    while (resto.leerDeDisco(posI)==true) {

        if (resto.getTipoRestaurante()==3) {

            while (reserva.leerDeDisco(posJ)==true) {

                if (resto.getCodigoRestaurante()==reserva.getCodigoRestaurante()) {

                    contRes++;

                }

                posJ++;

            }

            if (contRes<30) {

                nuevoObj.setCodigoRestaurante(resto.getCodigoRestaurante());

                nuevoObj.setNombre(resto.getNombre());

                nuevoObj.setCantReserv(contRes);

                nuevoObj.grabarEnDisco();

            }

        }

        contRes=0;

        posI++;

    }

}

int Punto2(){

    int vCate[10];

    Restaurante resto;

    int pos=0;

    int cateMax;

    while (resto.leerDeDisco(pos)==true) {

        vCate[resto.getTipoRestaurante()-1]++;

        pos++;

    }

    cateMax=vCate[0];

    for (int i=1; i<10; i++) {

        if (vCate[i]>cateMax) {

            cateMax=vCate[i];

        }

    }

    return cateMax;

}

int main () {

    int maximo;

    Punto1();

    maximo=Punto2();

    cout<<"LA CATEGORIA CON MAS RESTAURANTES ES LA CATEGORIA: "<<maximo<<endl;

    system("pause");

    return 0;

}

**PUNTO 3:**

class Animal{

protected:

int patas;

public:

int getPatas();

void setPatas();

};

class Felino: public Animal{

private:

int patas;

int peso;

public:

int getPeso();

void setPeso();

};

**PUNTO 4:**

Los constructores se ejecutan automáticamente (sin necesidad de llamarlos) al instanciar un objeto de una clase y llevan el mismo nombre de la clase.

Pueden ser útiles para:

- asignar valores a los atributos del objeto al momento de la creación del objeto.

- pedir valores en el momento de ser creado el objeto, ya sea con un parámetro, ninguno o varios.

Eso nos lleva a comentar también que podemos tener varios constructores en una clase pasándole diferente cantidad de parámetros (sobrecarga).