

PROIECT LA PROGRAMARE ORIENTATA PE OBIECTE

Trupoiu Razvan

Grupa 10204

Cuprins:

Capitolul 1

Notiuni introductive

Capitolul 2

Analiza si proiectarea orientata pe obiecte a sistemului informatic – modelul OMT

Capitolul 3

Implementarea si testarea sistemului informatic

Capitolul 4

Concluzie. Bibliografie

Capitolul 1

-Notiuni introductive-

Numele lantului nostru de restaurante este „All in One”, fiecare cu o anumita tema. Acesta contine mai multe ramuri cu diferite specialitati. Aveti de ales dintre o masa specializata in mancare italieneasca, una in mancare romana, dar si una in mancare japoneza. Am ales acest lant datorita diversitatii alimentatiei. Fiecare om prefera un anumit tip de mancare, asa ca am decis sa le satisfacem nevoile tuturor, dupa bunul plac.

Aplicatie:

Sa se implementeze clasa de baza „restaurant”, avand ca date membru nume, nr. clienti, nr. angajati si nr. locuri. Ca metode se vor folosi: constructor cu parametrii, un destructor, o functie membru pentru a afisa datele membru, o functie de tip set pt setarea numelui, nr. de clienti si a nr. de angajati si cate o functie de tip get pentru returnarea acestora.

Din clasa de baza „restaurant” deriveaza clasa „restaurant cu specialitati romanesti”, avand ca date membru nume, nr. clienti, nr. angajati si nr. locuri. Se vor folosi urmatoarele metode: constructor cu parametrii, destructor, o functie membru pentru redefinirea functiei de afisare din clasa de baza.

Tot din clasa de baza deriveaza si clasa „restaurant cu specialitati japoneze”, avand ca date membru numele. Se folosesc metodele: constructor cu parametrii, destructor, o functie membru pentru redefinirea functiei de afisare din clasa de baza.

Din clasa „restaurant” deriveaza clasa abstracta „restaurant cu specialitati italienesti”, fara date membru. Ca metode, se folosesc: constructor implicit, o functie virtuala pura pentru afisarea clasei abstracte.

Din clasa „restaurant cu specialitati italienesti” deriva clasa „restaurant cu specialitati italienesti de spaghetti”, avand ca date membru nume, nr. clienti, nr. angajati si nr. locuri. Se folosesc urmatoarele metode: constructor implicit, constructor cu parametrii, constructor de copiere, o functie membru pentru a redefini functia virtuala din clasa abstracta.

Din clasa „restaurant cu specialitati italienesti” deriva clasa „restaurant cu specialitati italienesti de pizza”, cu datele membru nume, nr. clienti, nr. angajati si nr. locuri. Metode: constructor implicit, constructor cu parametrii, constructor de copiere, o functie membru pentru a redefini functia virtuala din clasa abstracta.

Capitolul 2

-Analiza si proiectarea orientata pe obiecte a sistemului informatic-

1) OMT – Object Modelling Technique

Restaurant_cu_special itati_romanesti
char nume_restaurant[10]; int nr_angajati(); int nr_locuri(); int nr_clienti();
rest_ro(char*,int,int,in t,char*,int,int,int); ~rest_ro(); void afisare();

Restaurant
char nume_restaurant[10]; int nr_locuri(); int nr_clienti(); int nr_angajati();
rest(char*,int,int,int); ~rest(); void set_nume(char*); char*get_nume(); void set_nr_cl(int); int get_nr_cl(); void set_nr_ang(int); int get_nr_ang(); void afisare();

Restaurant_cu_special itati_japoneze
char nume_restaurant[10]; int nr_angajati(); int nr_locuri(); int nr_clienti();
rest_jap(char*,int,int,i nt,char*); ~rest_jap(); void afisare();

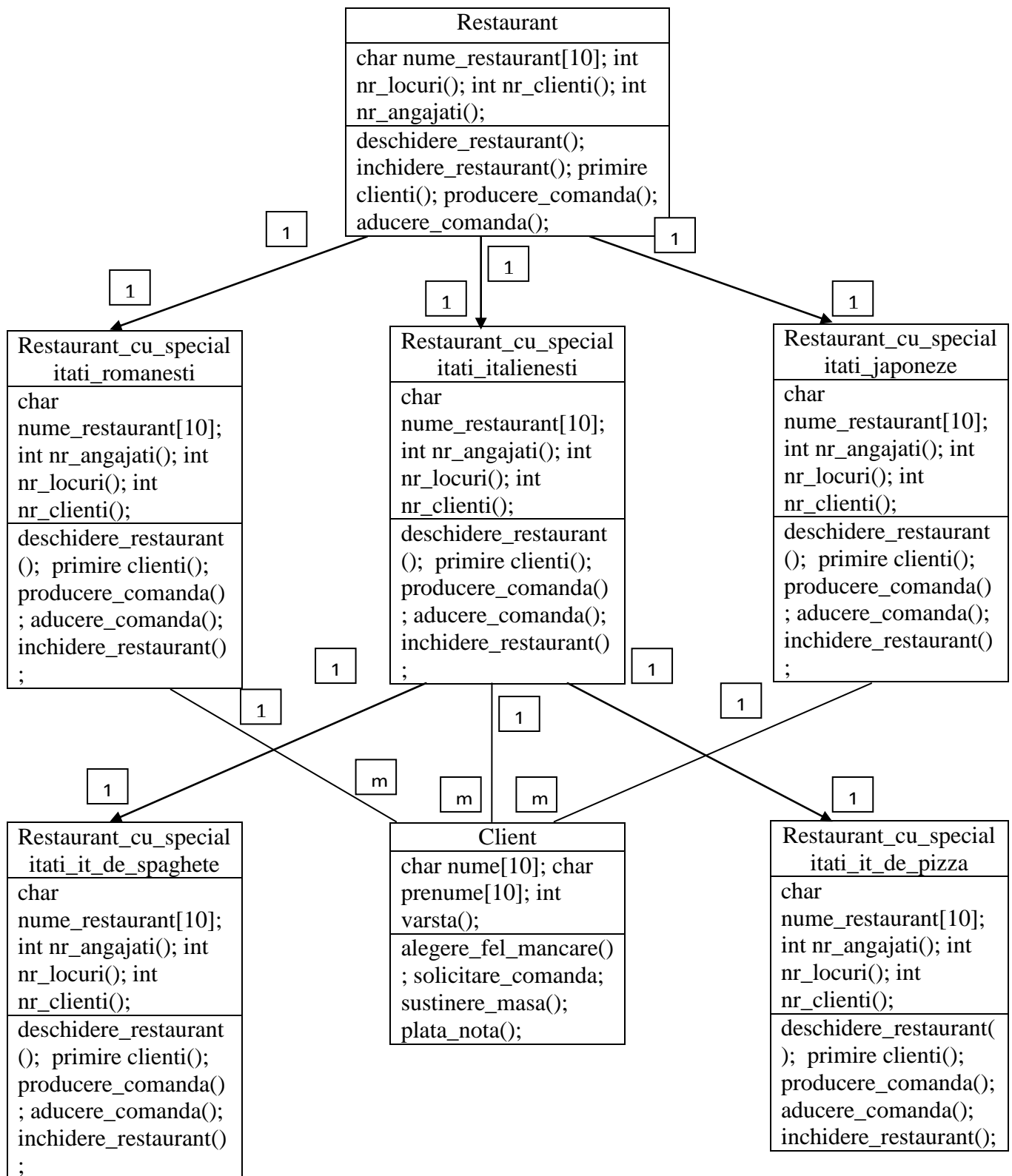
Restaurant_cu_special itati_it_de_spaghete
char nume_restaurant[10]; int nr_angajati(); int nr_locuri(); int nr_clienti();
rest_it_spg(); rest_it_spg(char,int,in t,int); rest_it_spg(rest_it_sp g&); virtual void afisare();

Restaurant_cu_special itati_italienesti
char nume_restaurant[10]; int nr_angajati(); int nr_locuri(); int nr_clienti();
rest_it(); virtual void afisare();

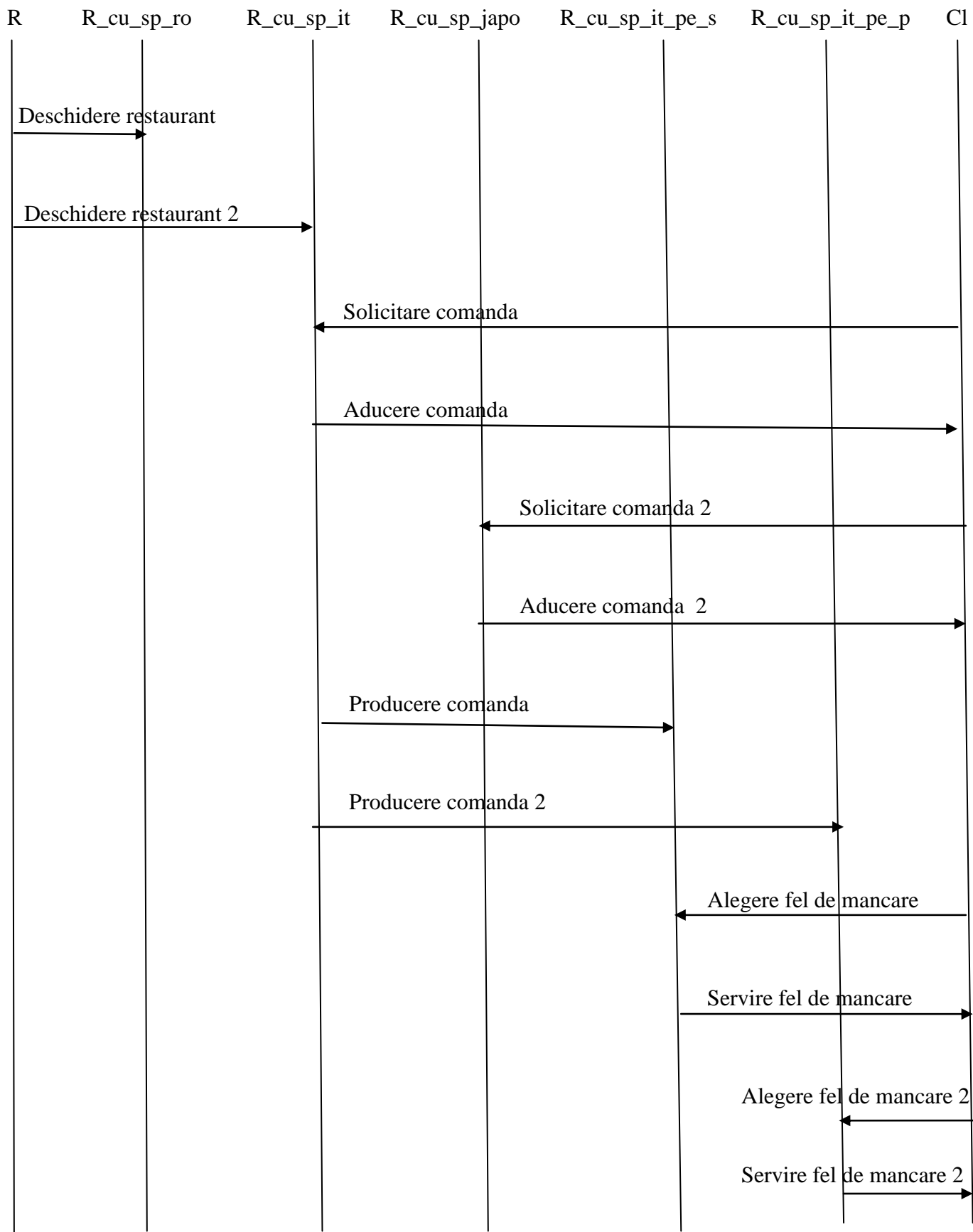
Restaurant_cu_special itati_it_de_pizza
char nume_restaurant[10]; int nr_angajati(); int nr_locuri(); int nr_clienti();
rest_it_pz(); rest_it_pz(char,int,int, int); rest_it_pz(rest_it_pz&); virtual void afisare();

Client
char nume[10]; char prenume[10]; int varsta();
client(); client(char, char, int); client(client&); friend char aleg_fel_mancare(); friend char solicitare_comanda();

2) Diagrama obiectelor

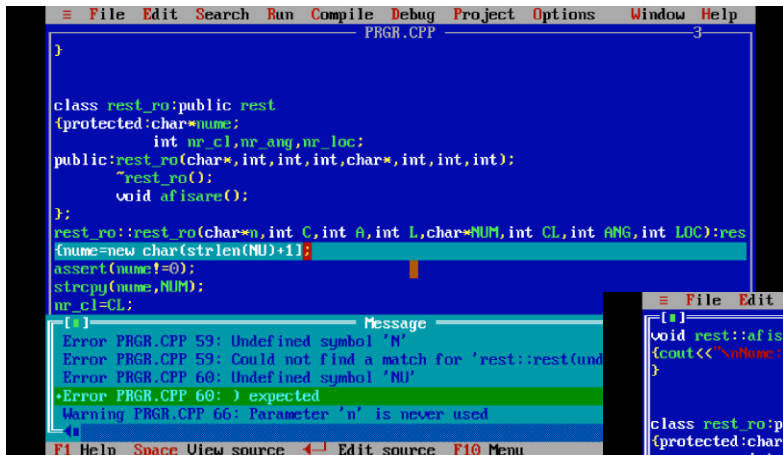


3) Diagrama succesiunii evenimentelor



Capitolul 3

-Implementarea si testarea sistemului informatic-



```
PRGR.CPP
3

class rest_ro:public rest
{protected:char*name;
  int nr_cl,nr_ang,nr_loc;
public:rest_ro(char*,int,int,int,char*,int,int,int):
  ~rest_ro();
  void afisare();
};

rest_ro::rest_ro(char*n,int C,int A,int L,char*NUM,int CL,int ANG,int LOC):res
{name=new char(strlen(NUM)+1);
assert(name!=0);
strcpy(name,NUM);
nr_cl=CL;
nr_ang=ANG;
nr_loc=LOC;
}
```

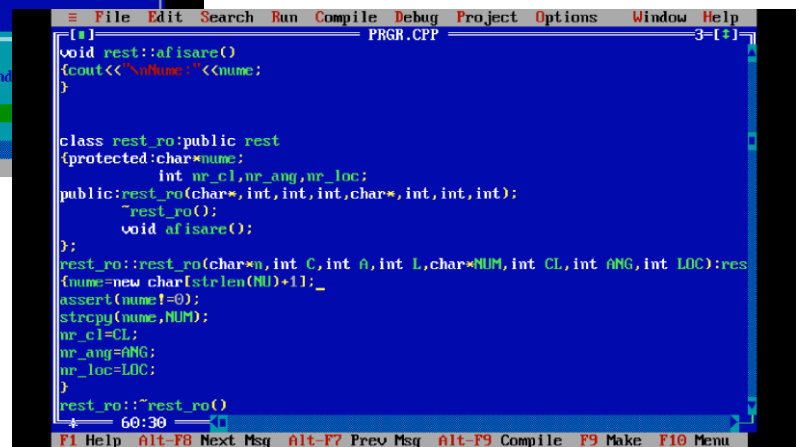
Message

- Error PRGR.CPP 59: Undefined symbol 'N'
- Error PRGR.CPP 59: Could not find a match for 'rest::rest'
- Error PRGR.CPP 60: Undefined symbol 'NU'
- Error PRGR.CPP 60:) expected
- Warning PRGR.CPP 66: Parameter 'n' is never used

F1 Help Space View source Edit source F10 Menu

Eroare: Paranteza lipsa din constructorul cu parametrii din clasa restaurantul cu specialitati romanesti

Rezolvare eroare: Deschiderea parantezei patrate „[” la constructorul de parametrii din clasa restaurantul cu specialitati romanesti



```
PRGR.CPP
3-[+]

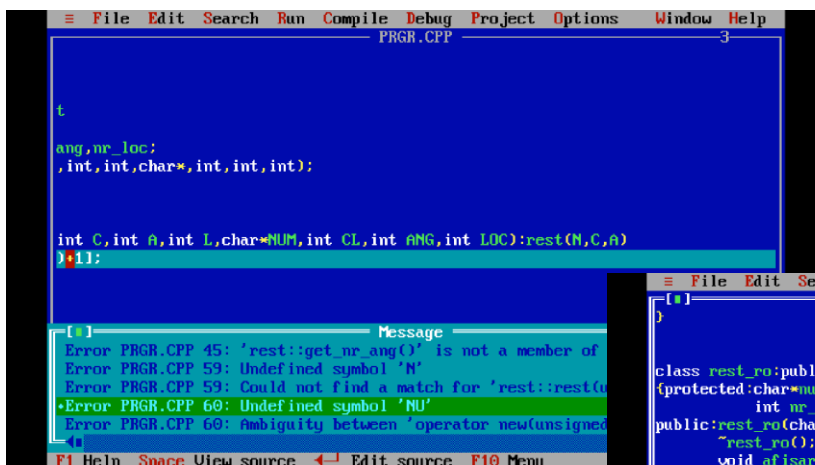
void rest::afisare()
{cout<<"nume:"<<name;
}

class rest_ro:public rest
{protected:char*name;
  int nr_cl,nr_ang,nr_loc;
public:rest_ro(char*,int,int,int,char*,int,int,int):
  ~rest_ro();
  void afisare();
};

rest_ro::rest_ro(char*n,int C,int A,int L,char*NUM,int CL,int ANG,int LOC):res
{name=new char(strlen(NUM)+1);
assert(name!=0);
strcpy(name,NUM);
nr_cl=CL;
nr_ang=ANG;
nr_loc=LOC;
}

rest_ro::~rest_ro()
{delete name;
}
```

F1 Help Alt-F8 Next Msg Alt-F7 Prev Msg Alt-F9 Compile F9 Make F10 Menu



```
PRGR.CPP
3

t
ang,nr_loc;
,int,int,char*,int,int,int):

int C,int A,int L,char*NUM,int CL,int ANG,int LOC):rest(N,C,A)
):11;
```

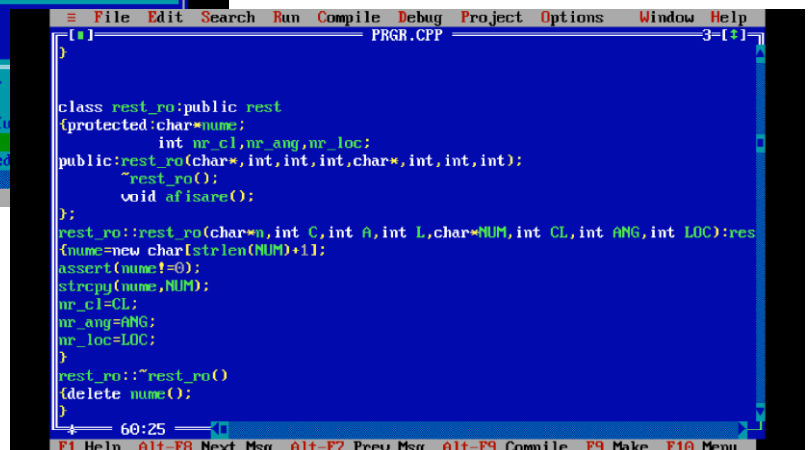
Message

- Error PRGR.CPP 45: 'rest::get_nr_ang()' is not a member of 'rest'
- Error PRGR.CPP 59: Undefined symbol 'N'
- Error PRGR.CPP 59: Could not find a match for 'rest::rest'
- Error PRGR.CPP 60: Undefined symbol 'NU'
- Error PRGR.CPP 60: Ambiguity between 'operator new(unsigned int)' and 'operator new(unsigned int, const char*)'

F1 Help Space View source Edit source F10 Menu

Eroare: Declararea gresita a numelui cu „NU”, in loc de „NUM”

Rezolvare eroare: Redenumirea din „NU” in „NUM”



```
PRGR.CPP
3-[+]

class rest_ro:public rest
{protected:char*name;
  int nr_cl,nr_ang,nr_loc;
public:rest_ro(char*,int,int,int,char*,int,int,int):
  ~rest_ro();
  void afisare();
};

rest_ro::rest_ro(char*n,int C,int A,int L,char*NUM,int CL,int ANG,int LOC):res
{name=new char(strlen(NUM)+1);
assert(name!=0);
strcpy(name,NUM);
nr_cl=CL;
nr_ang=ANG;
nr_loc=LOC;
}

rest_ro::~rest_ro()
{delete name;
}
```

F1 Help Alt-F8 Next Msg Alt-F7 Prev Msg Alt-F9 Compile F9 Make F10 Menu

```

class rest_ro:public rest
{protected:char*nume;
    int nr_cl,nr_ang,nr_loc;
public:rest_ro(char*,int,int,int,char*,int,int,int):
    ~rest_ro();
    void afisare();
};
rest_ro::rest_ro(char*n,int C,int A,int L,char*NUM,int CL,int ANG,int LOC):res
(nume=new char[strlen(NUM)+1];
assert(nume!=0);
strcpy(nume,NUM);
nr_cl=CL;

```

Message

Error PRGR.CPP 45: 'rest::get_nr_ang()' is not a member of 'rest'

Error PRGR.CPP 59: Undefined symbol 'N'

Error PRGR.CPP 59: Could not find a match for 'rest::rest()'

Warning PRGR.CPP 66: Parameter 'n' is never used

Warning PRGR.CPP 66: Parameter 'C' is never used

Eroare: Variabila gresita. „n” in loc de „NUM”

Rezolvare eroare: Redenumirea din „n” in „N”

```

class rest_ro:public rest
{protected:char*nume;
    int nr_cl,nr_ang,nr_loc;
public:rest_ro(char*,int,int,int,char*,int,int,int):
    ~rest_ro();
    void afisare();
};
rest_ro::rest_ro(char*N,int C,int A,int L,char*NUM,int CL,int ANG,int LOC):res
(nume=new char[strlen(NUM)+1];
assert(nume!=0);
strcpy(nume,NUM);
nr_cl=CL;
nr_ang=ANG;
nr_loc=LOC;
}
rest_ro::~rest_ro()
{delete nume;
}

```

```

nr_cl=Y.nr_cl;
nr_ang=Y.nr_ang;
nr_loc=Y.nr_loc;
}
void rest_it_pz::afisare()
{cout<<"Nefect it cu pizza:";
}
void main()
{rest ob1("All in one","3000","300","5000");
ob1.afisare();
rest_ro ob2("All in one","3000","300","5000","Cetate","80","20","100");
ob2.afisare();
rest_jap ob3("All in one","3000","300","5000","Kanto");
}

```

Message

Error PRGR.CPP 181: Multiple declaration for 'ob1'

Error PRGR.CPP 181: Could not find a match for 'rest'

Error PRGR.CPP 182: Cannot convert 'rest*' to 'rest'

Error PRGR.CPP 184: Multiple declaration for 'ob2'

Error PRGR.CPP 184: Could not find a match for 'rest'

Eroare: Declararea obiectului 1 si 2 de 2 ori

Rezolvare eroare: Denumirea obiectelor 1 si 2 cu 4 si 5

```

void rest_it_pz::afisare()
{cout<<"Nefect it cu pizza:";
}
void main()
{rest ob1("All in one","3000","300","5000");
ob1.afisare();
rest_ro ob2("All in one","3000","300","5000","Cetate","80","20","100");
ob2.afisare();
rest_jap ob3("All in one","3000","300","5000","Kanto Sushii","150","30","200");
ob3.afisare();
rest_it_ptr ptr;
rest_it_spg ob4(3);
ptr=&ob4;
ob4.afisare();
rest_it_pz ob5(7,9);
ptr=&ob5;
ob5.afisare();
}

```


Capitolul 4

-Bibliografie-

Bibliografie:

- Cursul de programare orientata pe obiecte
- Laboratorul de programare orientata pe obiecte