Цэрэнбат Номин-Эрдэнэ

(20B1NUM0426)

ICSI201 - Объект хандлагат програмчлал

МУИС-ХШУИС

Мэдээллийн технологи

Лаборатори №8

Цэрэнбат Номин-Эрдэнэ

1. ОРШИЛ

Энэ лабораторийн ажлийн хүрээнд Обьект хандлагат С хэлний чухал хэсэг болох удамшил ямар харьцаа үүсгэдэг тухай болон бүрдэл харьцааг судалж, тайлбарлаж, ойлгож авна.

2. ЗОРИЛГО

- Лабораторийн ажлыг алдаагүй зөв хийж гүйцэтгэх.
- Division болон JobDescription классуудын хэд хэдэн объект байгуулах
- Employee классын хэд хэдэн объект байгуулж тус бүрийн Division, JobDescription –ийг тодорхойлох.
- Employee классын объект тус бүрд Spouse, Child – уудыг тохируулж өгөх
- Employee классын объект тус бүрийн бүх мэдээллийг хэвлэх.

3. ОНОЛЫН СУДАЛГАА

3.1 Удамшил ямар харьцаа үүсгэдэг вэ? Объект хандлагат программчлалд хэрхэн хэрэгжүүлдэг вэ?

Удамших болон удамшуулах харьцаанд орж буй классуудын хооронд тэр бол тэр эсвэл тийм төрлийх гэсэн хоёр харьцаа үүсгэдэг. Жишээлвэл тэр бол тэр гэдэг нь хүүхэд бол хүн харин тийм төрлийх гэдэгт нь машин бол тээврийн хэрэгсэлийн төрлийх гэх мэтчилэн.

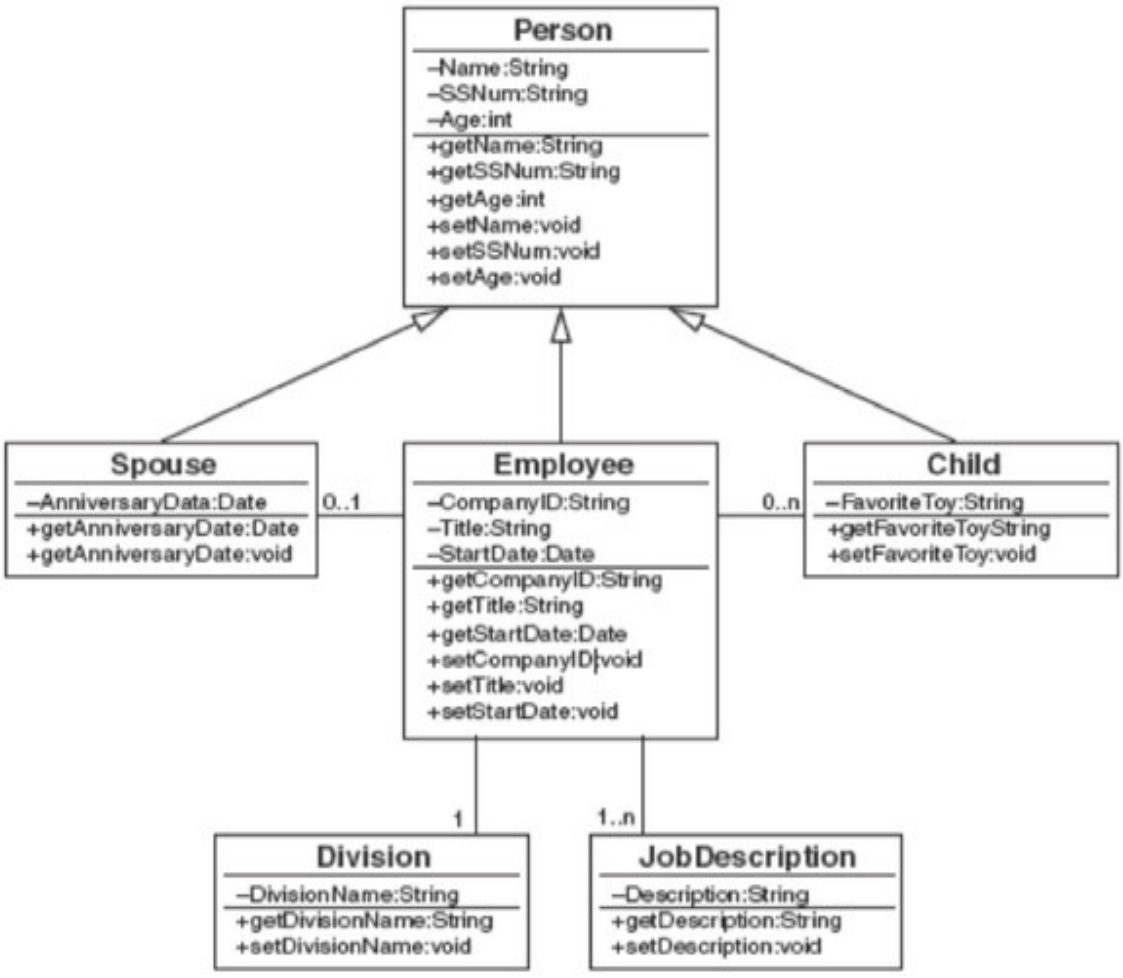
3.2 Бүрдэл харьцаа гэж юу вэ? Объект хандлагат программчлалд хэрхэн хэрэгжүүлдэг вэ?

Класын шинж нь өөр класых байж болох ба нарийн бүтэцтэй бүрдмэл классыг үүсгэхэд ашигладаг. Энэ нь тийм юмтай байх гэсэн харьцааг үсгэдэг.

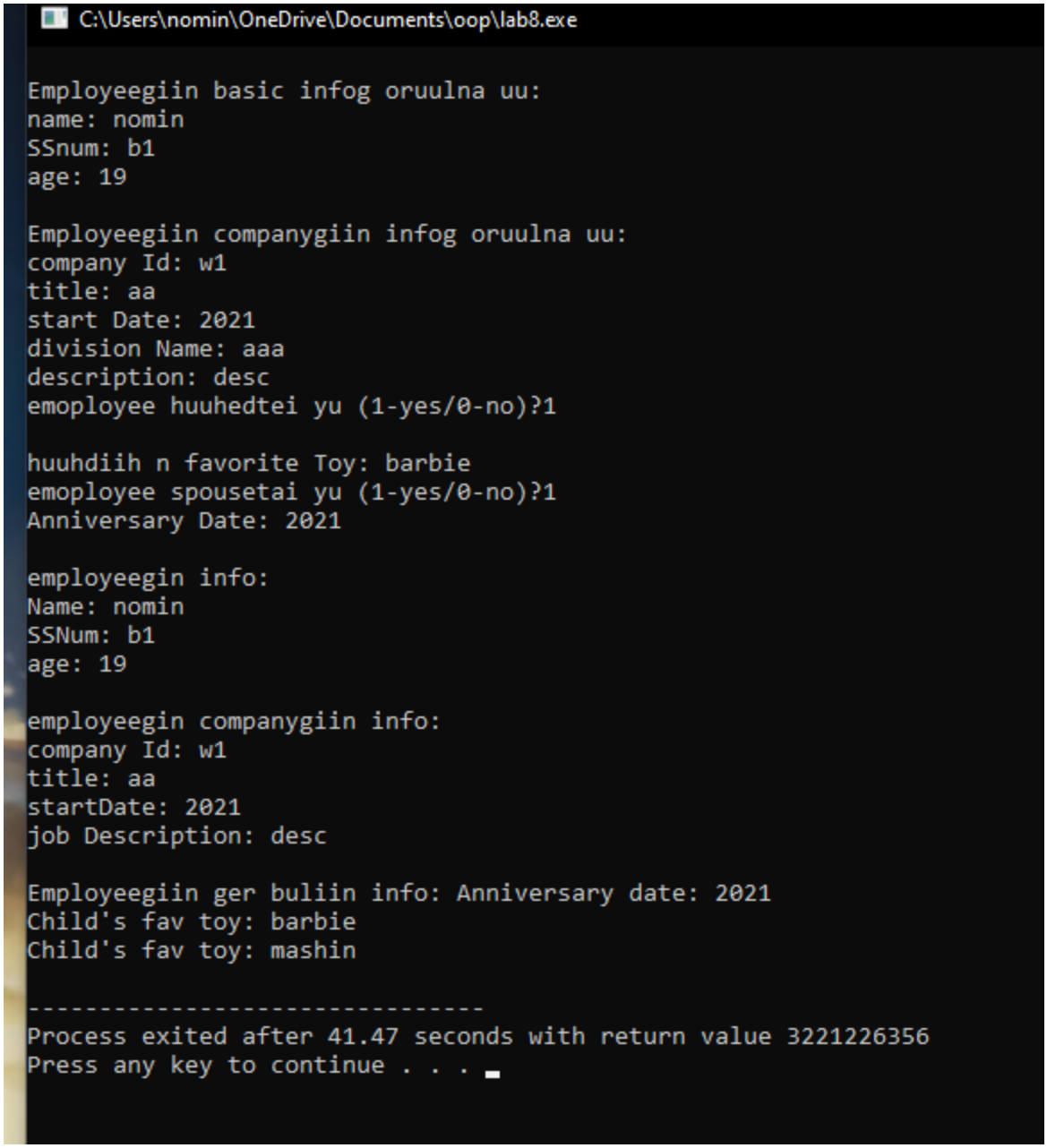
```
Class engine{
        Public:int c;
};
Class car {
        Public: char name[15];
        engine e1; //mashin ni motortoi baix estoi
        engine e2;
};
```

4. ХЭРЭГЖҮҮЛЭЛТ

1. Дараах зурагт харуулсан класс диаграмын дагуу классуудаг байгуул.
2. Division болон JobDescription классуудын хэд хэдэн объект байгуул
3. Employee классын хэд хэдэн объект байгуулж тус бүрийн Division, JobDescription –ийг зааж өг.
4. Employee классын объект тус бүрд Spouse, Child – уудыг тохируулж өг
5. Employee классын объект тус бүрийн бүх мэдээллийг хэвлэ.
6. 0..1, 0..n болон 1, 1..n харьцаануудыг зөв програмчилсан байх шаардлагатай.



Cardinality in a UML diagram.


C:\Users\nomin\OneDrive\Documents\oop\lab8.exe
Employeegiin basic infog oruulna uu:
name: nomin
SSnum: b1
age: 19
Employeegiin companygiin infog oruulna uu:
company Id: w1
title: aa
start Date: 2021
division Name: aaa
description: desc
emoployee huuhedtei yu (1-yes/0-no)?1
huuhdiih n favorite Toy: barbie
emoployee spousetai yu (1-yes/0-no)?1
Anniversary Date: 2021
employeegin info:
Name: nomin
SSNum: b1
age: 19
employeegin companygiin info:
company Id: w1
title: aa
startDate: 2021
job Description: desc
Employeegiin ger buliin info: Anniversary date: 2021
Child's fav toy: barbie
Child's fav toy: mashin
Process exited after 41.47 seconds with return value 3221226356
Press any key to continue . . .

lab8.cpp tes8t.cpp

```cpp
#include<iostream>
#include<iomanip>
#include<stdio.h>
#include<conio.h>
#include<string>
#include <cstring>
#include <vector>

using namespace std;

class person{
    public:
        char * name;
        char * SSNum;
        int age;


        person();
        ~person();
        void getName();
        void getSSNum();
        int getAge();
        void setName(char * a);
        void setSSNum(char * b);
        void setAge(int c);
        void printPersonInfo();
};

person::person(){
    name = new char;
    strcpy(name, "ner");
    SSNum = new char;
    strcpy(SSNum, " ");
    age = 28;
}

person::~person(){
    cout<<endl;
    delete name;
    delete SSNum;


}

void person::getName(){
    cout<<"name: ";
    cin>>name;
}
```

```
49  void person::getSSNum(){
50      cout<<"SSnum: ";
51      cin>>SSNum;
52  }
53
54  int person::getAge(){
55      cout<<"age: ";
56      cin>>age;
57  }
58
59
60  void person::setName(char * a){
61      name = new char[strlen(a)+1];
62      strcpy(name, a);
63  }
64  void person::setSSNum(char * b){
65      SSNum = new char[strlen(b)+1];
66      strcpy(SSNum, b);
67  }
68  void person::setAge(int c){
69      age = c;
70  }
71
72
73
74  void person::printPersonInfo() {
75      cout << "\n Name: " <<setw(8)<< name;
76      cout << "\n Ssnum: " <<setw(8)<< SSNum;
77      cout << "\n Age: " <<setw(8)<< age;
78
79  }
80
81
82
83  //---------SPOUSE------------//
84  class spouse : virtual public person {
85      public:
86          int anniversaryDate;
87
88
89          spouse(){
90          anniversaryDate = 2000;
91          }
92
93          int getAnniversaryDate();
94          void setAnniversaryDate(int ad);
95
```

```
94          void setAnniversaryDate(int ad);
95
96          void printSpouseInfo(){
97              cout<<"Anniversary date: "<<anniversaryDate<<endl;
98          }
99
100 };
101
102 int spouse::getAnniversaryDate(){
103     cout<<"Anniversary Date: ";
104     cin>>anniversaryDate;
105 }
106
107 void spouse::setAnniversaryDate(int ad){
108     anniversaryDate = ad;
109 }
110
111
112
113 //---------CHILD------------//
114
115 class child : virtual public person{
116     public:
117         char * favoriteToy;
118
119         child(){
120             favoriteToy = new char;
121             strcpy(favoriteToy, "mashin");
122         }
123         ~child(){
124             delete favoriteToy;
125         }
126         char getfavoriteToy();
127         void setfavoriteToy(char *ft);
128
129         void printChildInfo(){
130             cout<<"Child's fav toy: "<<favoriteToy<<endl;
131         }
132 };
133
134 char child::getfavoriteToy(){
135     cout<<"huuhdiih n favorite Toy: ";
136     cin>>favoriteToy;
137 }
138
```

```
139 void child::setfavoriteToy(char * ft){
140     favoriteToy = new char[strlen(ft)+1];
141     strcpy(favoriteToy, ft);
142 }
143
144
145
146 //---------DIVISION------------//
147 class division{
148     public:
149         char *divisionName;
150
151
152         division(){
153             divisionName = new char;
154             strcpy(divisionName, " ");
155         }
156         ~division(){
157             cout<<endl;
158             delete divisionName;
159         }
160         char getDivisionName();
161         void setDivisionName(char *div);
162
163 };
164
165
166 char division::getDivisionName(){
167     cout<<"division Name: ";
168     cin>>divisionName;
169 }
170
171 void division::setDivisionName(char *div){
172     divisionName = new char[strlen(div)+1];
173     strcpy(divisionName, div);
174 }
175
176
177
178 //---------JOB DESCRIPTION--------//
179 class jobDescription {
180     public:
181         char *description;
182
183
184         jobDescription(){
185             description = new char;
```

```
196
197 char jobDescription::getDescription(){
198     cout<<"description: ";
199     cin>>description;
200 }
201
202 void jobDescription::setDescription(char *jd){
203     description = new char[strlen(jd)+1];
204     strcpy(description, jd);
205 }
206
207
208 //--------EMPLOYEE-------------------------//
209
210 class employee :  public division, public jobDescription, public spouse, public child {
211     public:
212         char *companyId;
213         char *title;
214         int startDate;
215
216
217         employee(){
218             companyId = new char;
219             strcpy(companyId, "c1");
220             title = new char;
221             strcpy(title, "worker");
222             startDate = 2018;
223         }
224         ~employee(){
225
226             delete companyId;
227             delete title;
228
229         }
230         char getCompanyId(){
231             cout<<"company Id: ";
232             cin>>companyId;
233         }
234         char getTitle(){
235             cout<<"title: ";
236             cin>>title;
237         }
238         int getStartDate(){
239             cout<<"start Date: ";
240             cin>>startDate;
241         }
242
```

```
            strcpy(companyId, cId);
        }
        void setTitle(char * t){
            title = new char[strlen(t)+1];
            strcpy(title, t);
        }
        void setStartDate(int d){
            startDate = d;
        }



        void getEmployeeInfo(){
            cout<< "\nEmployeegiin basic infog oruulna uu: \n";
            getName();
            getSSNum();
            getAge();

            cout<< "\nEmployeegiin companygiin infog oruulna uu: \n";
            getCompanyId();
            getTitle();
            getStartDate();

            getDivisionName();
            getDescription();


        }

        void printEmployeeInfo(void){
            cout<<"\nemployeegin info: "<<endl;
            cout<<"Name: "<<name<<endl;
            cout<<"SSNum: "<<SSNum<<endl;
            cout<<"age: "<<age<<endl;

            cout<<"\nemployeegin companygiin info: "<<endl;

            cout<<"company Id: "<<companyId<<endl;
            cout<<"title: "<<title<<endl;
            cout<<"startDate: "<<startDate<<endl;
            cout<<"job Description: "<<description<<endl;


        }
};
```

```
291
292
293   //---------MAIN--------//
294
295   int main(){
296       int n, d;
297       int i;
298       employee p;
299       child ch1;
300       child ch2;
301       spouse s1;
302
303       vector<child> NChild;
304
305       p.getEmployeeInfo();
306       cout<<"emoployee huuhedtei yu (1-yes/0-no)?";
307       cin>>d;
308
309       if (d==1){
310           NChild.push_back( ch1 );
311           NChild.push_back( ch2 );
312           NChild[0].getfavoriteToy();
313
314
315       }else{
316           cout<<"employee huuhedgui";
317       }
318
319       cout<<"emoployee spousetai yu (1-yes/0-no)?";
320       cin>>d;
321       if (d==1){
322           s1.getAnniversaryDate();
323       }else{
324           cout<<"employee spousegui\n";
325       }
326
327
328       p.printEmployeeInfo();
329       cout<< "\nEmployeegiin ger buliin info: ";
330       s1.printSpouseInfo();
331
332
333       NChild[0].printChildInfo();
334       NChild[1].printChildInfo();
335
336   }
```

5. ХАВСРАЛТ

```
#include<iostream>
#include<iomanip>
#include<stdio.h>
#include<conio.h>
#include<string>
```

```
#include <cstring>
#include <vector>

using namespace std;

class person{
	public:
		char * name;
		char * SSNum;
		int age;


		person();
		~person();
		void getName();
		void getSSNum();
		int getAge();
		void setName(char * a);
		void setSSNum(char * b);
		void setAge(int c);
		void printPersonInfo();
};

person::person(){
	name = new char;
	strcpy(name, "ner");
	SSNum = new char;
	strcpy(SSNum, " ");
	age = 28;
}

person::~person(){
	cout<<endl;
	delete name;
	delete SSNum;

}

void person::getName(){
	cout<<"name: ";
	cin>>name;
}

void person::getSSNum(){
	cout<<"SSnum: ";
	cin>>SSNum;
```

```
}

int person::getAge(){
	cout<<"age: ";
	cin>>age;
}


void person::setName(char * a){
	name = new char[strlen(a)+1];
	strcpy(name, a);
}
void person::setSSNum(char * b){
	SSNum = new char[strlen(b)+1];
	strcpy(SSNum, b);
}
void person::setAge(int c){
	age = c;
}



void person::printPersonInfo() {
	cout << "\n Name: " <<setw(8)<< name;
	cout << "\n Ssnum: " <<setw(8)<< SSNum;
	cout << "\n Age: " <<setw(8)<< age;

}



//---------SPOUSE------------//
class spouse : virtual public person {
	public:
		int anniversaryDate;


		spouse(){
		anniversaryDate = 2000;
		}

		int getAnniversaryDate();
		void setAnniversaryDate(int ad);

		void printSpouseInfo(){
			cout<<"Anniversary date: "<<anniversaryDate<<endl;
```

```
                }

};

int spouse::getAnniversaryDate(){
        cout<<"Anniversary Date: ";
        cin>>anniversaryDate;
}

void spouse::setAnniversaryDate(int ad){
        anniversaryDate = ad;
}



//---------CHILD------------//

class child : virtual public person{
        public:
                char * favoriteToy;

                child(){
                        favoriteToy = new char;
                        strcpy(favoriteToy, "mashin");
                }
                ~child(){
                        delete favoriteToy;
                }
                char getfavoriteToy();
                void setfavoriteToy(char *ft);

                void printChildInfo(){
                        cout<<"Child's fav toy: "<<favoriteToy<<endl;
                }
};

char child::getfavoriteToy(){
        cout<<"huuhdiih n favorite Toy: ";
        cin>>favoriteToy;
}

void child::setfavoriteToy(char * ft){
        favoriteToy = new char[strlen(ft)+1];
        strcpy(favoriteToy, ft);
}
```

```
//---------DIVISION------------//
class division{
	public:
		char *divisionName;


		division(){
			divisionName = new char;
			strcpy(divisionName, " ");
		}
		~division(){
			cout<<endl;
			delete divisionName;
		}
		char getDivisionName();
		void setDivisionName(char *div);

};


char division::getDivisionName(){
	cout<<"division Name: ";
	cin>>divisionName;
}

void division::setDivisionName(char *div){
	divisionName = new char[strlen(div)+1];
	strcpy(divisionName, div);
}



//---------JOB DESCRIPTION--------//
class jobDescription {
	public:
		char *description;


		jobDescription(){
			description = new char;
			strcpy(description, "intern");
		}
		~jobDescription(){
			delete description;
```

```
		}
		char getDescription();

		void setDescription(char *jd);

};

char jobDescription::getDescription(){
	cout<<"description: ";
	cin>>description;
}

void jobDescription::setDescription(char *jd){
	description = new char[strlen(jd)+1];
	strcpy(description, jd);
}


//--------EMPLOYEE-------------------------//

class employee :  public division, public jobDescription, public spouse, public child {
	public:
		char *companyId;
		char *title;
		int startDate;


		employee(){
			companyId = new char;
			strcpy(companyId, "c1");
			title = new char;
			strcpy(title, "worker");
			startDate = 2018;
		}
		~employee(){

			delete companyId;
			delete title;

		}
		char getCompanyId(){
			cout<<"company Id: ";
			cin>>companyId;
		}
		char getTitle(){
			cout<<"title: ";
```

```
        cin>>title;
}
int getStartDate(){
        cout<<"start Date: ";
        cin>>startDate;
}

char setCompanyId(char * cid){
        companyId = new char[strlen(cid)+1];
        strcpy(companyId, cid);
}
void setTitle(char * t){
        title = new char[strlen(t)+1];
        strcpy(title, t);
}
void setStartDate(int d){
        startDate = d;
}



void getEmployeeInfo(){
        cout<< "\nEmployeegiin basic infog oruulna uu: \n";
        getName();
        getSSNum();
        getAge();

        cout<< "\nEmployeegiin companygiin infog oruulna uu: \n";
        getCompanyId();
        getTitle();
        getStartDate();

        getDivisionName();
        getDescription();


}

void printEmployeeInfo(void){
        cout<<"\nemployeegin info: "<<endl;
        cout<<"Name: "<<name<<endl;
        cout<<"SSNum: "<<SSNum<<endl;
        cout<<"age: "<<age<<endl;

        cout<<"\nemployeegin companygiin info: "<<endl;
```

```
                        cout<<"company Id: "<<companyId<<endl;
                        cout<<"title: "<<title<<endl;
                        cout<<"startDate: "<<startDate<<endl;
                        cout<<"job Description: "<<description<<endl;


                }
};



//---------MAIN--------//

int main(){
        int n, d;
        int i;
        employee p;
        child ch1;
        child ch2;
        spouse s1;

        vector<child> NChild;

        p.getEmployeeInfo();
        cout<<"emoployee huuhedtei yu (1-yes/0-no)?";
        cin>>d;

        if (d==1){
                NChild.push_back( ch1 );
                NChild.push_back( ch2 );
                NChild[0].getfavoriteToy();


        }else{
                cout<<"employee huuhedgui";
        }

        cout<<"emoployee spousetai yu (1-yes/0-no)?";
        cin>>d;
        if (d==1){
                s1.getAnniversaryDate();
        }else{
                cout<<"employee spousegui\n";
        }
```

```
        p.printEmployeeInfo();
        cout<< "\nEmployeegiin ger buliin info: ";
        s1.printSpouseInfo();


        NChild[0].printChildInfo();
        NChild[1].printChildInfo();

}
```