Лекция 2

КЛАСТАР: СИНТАКСИС НЕГІЗДЕРІ, ДЕКЛАРАЦИЯ ЖӘНЕ АНЫҚТАМАЛАР

Класс – C++ тілі абстракциясының негізгі бірлігі және мәліметтер құрылымының тек мәліметтер ғана емес сонымен қатар функциялардан тұратын кеңейтілген түсінігі болып табылады. Кластарда class кілттік сөзін қолданып хабарлайды:

```
class class name {
  access specifier 1:
member1;
  access specifier 2:
    member2;
  ...
  } object_names;
```

Мұндағы class_name ***кластың идентификаторы***, object_names – осы ***класс объектілерінің тізімі***. Программа денесі мәліметтердің хабарламасы және функциялардың хабарламасы және қосымша қол жетімділік спецификаторлары элементтерінен тұрады. Қол жетімділік спецификаторы келесі үш кілттік сөзден тұратын: private, public немесе protected өрнек болып табылады. Бұл спецификаторлар оларды хабарлап болған соң элементтердің қолжетімділік құқын өзгертеді:

– ***private*** класының мүшелері осы кластың ішіндегі элементтерге ғана немесе оның friends класына қол жетімді болады;

– ***protected*** класының мүшелері осы кластың ішіндегі элементтерге, оның friends класына, сонымен қатар оның туынды кластарының элементтеріне қол жетімді болады;

– ***public*** егер объект көрініп тұрса, кез келген кластың элементтеріне қол жетімді болады.

Келесі листинг BankAccount класын хабарлау мысалы болып табылады:

```
1:  classBankAccount{
2:
3:  private:
4:  double sum;
5:  string name;
6:  public:
7:  BankAccount(stringnm):name(nm),sum(0){}
8:  double balance(){returnsum;}
9:  void deposit(doubleamount){sum+=amount;}
10: void withdraw(doubleamount){sum-=amount;}
11: string getName(){returnname;}
12: };
```

3-жолдан бастап 7-жолдың модификаторларына дейінгі кластың барлық мүшелеріне private қолжетімділігі орнатылған.

CONSTRUCTORS (ҚҰРАУШЫЛАР)

Конструктор кластың жаңа көшірмесін құрайды және баптайды. C++ тілінде кластар бірнеше құраушылардан тұрады. ***Бұл класс көшірмесіне параметр типтері мен мәндерін қоса алғанда өзгерістер енгізуге мүмкіндік береді және олар әрбір құраушыда болуы мүмкін***.

Төменде қосымша құраушылар енгізілген BankAccount класы келтірілген. Берілген кластың мүшелері үшін құрастырғыштарда initializer lists-ты қолдану – бастапқы шарттарын анықтаудың ыңғайлы тәсілі. Initializer lists құрастырғыштан кейін жазылатын үтірмен ерекшеленген. 8-жолда қос нүктеден бос жүйелі жақшаға дейін initializer тізімі енгізілген. Initializer listname мәліметтер мүшесінің nm параметрінің шартына тең бастапқы шартына private орнатады. Сонымен қатар ол мәліметтердің sum бастапқы параметрін нөлге тең етіп орнатады.

Initializer lists and multiple constructors

```
1:  class BankAccount{
2:
3:  private:
4:  double sum;
5:  string name;
6:
7:  public:
8:  BankAccount(stringnm):name(nm),sum(0){}
9:  BankAccount(stringnm,double bal):
10: name(nm),sum(bal){}
11:
12: double balance(){returnsum;}
13: void deposit(doubleamount){sum+=amount;}
14: void withdraw(doubleamount){sum-=amount;}
15: string getName(){returnname;}
16: };
```

C++ программалау тілі объект декларациясынан тұратын код жолы орындалатын объект мысалын суреттейді. Объект көшірмесі класс құрастырғышының орындалуын қамтиды. 3-листинг BankAccount-тің әртүрлі екі объектісін хабарлайды

Объектіні баптау

```
1:  BankAccountaccount1("checking");
2:  BankAccountaccount2("savings",200);
3:
4:  account2.withdraw(100);
5:  account1.deposit(100);
```

DESTRUCTOR (ДЕСТРУКТОР)

Деструктор – объект жұмысының дұрыс аяқталуын жүзеге асыруға және жадыда орын босатуға арналған арнайы функция. Деструктордың құрастырғыштан айырмашылығы класта тек бір рет қана болады. Мысалы, мәліметтер қорында объект жұмысын аяқтағаннан соң деструктор мәліметтер қорымен жұмысын аяқтайды. Деструктордың синтаксисін келесі түрде бейнелеуге болады:

```
1:  ~BankAccount(){
2:  if(balance()<0){
3:  cout<<"Warning: negative balance!"<<endl;
4:  }
5:  }
```

DECLARATIONVS. DEFINITION

Кластар спецификациясында «***definition***» (***анықтама***) термині функцияда қолданылған болатын. Біз функцияны «анықтағанда» жүйелі жақша ішінде жазылған код арқылы функцияның жұмысын анықтаймыз. Бір жағынан функцияның «декларациясы»

тек функцияның интерфейсін анықтайды. Бұл ***интерфейс*** функцияның атауынан, қайтарымды типінен, параметрлерінің тізімінен тұрады.

Келесі листинг декларация және функцияның орта мәнін анықтауды бейнелейді.

Декларация және анықтама

```
1:  #include<iostream>
2:  #include<cstdlib>
3:
4:  usingnamespacestd;
5:
6:  // function declaration
7:  doubleaverage(int,int);
8:
9:  intmain(intargc,char*argv[]){
10:
11: cout<<average(10,2)<<endl;
12: returnEXIT_SUCCESS;
13: }
14:
15: // function definition
16: doubleaverage(inttotal,intcount){
17: return(total/count);
18: }
```

Функцияның декларациясы мен анықтамасы көптеген аспектілер бойына ерекшеленеді. Жоғарыда келтірілген мысал бойынша функцияның орта мәнін анықтайтын декларация жүйелі жақша емес, нүктелі үтір арқылы белгіленіп тұр. Функцияның декларациясы параметрлер үшін айнымалылардың атауларын енгізбейді. Функцияның анықтамасы оны бас функцияда қолданып болған соң анықталады. Мұны тек әлі анықталмаған класс немесе функция үшін қолдануға болады.

Класс сыртындағы класс мүшесінің функциясы

```
:   classBankAccount{
2:  private:
3:  doublesum;
4:  stringname;
5:
6:  public:
7:  BankAccount(stringnm):name(nm),sum(0){}
8:  BankAccount(stringnm,doublebal):
9:  name(nm),sum(bal){}
10:
11: doublebalance();
12: voiddeposit(double);
13: voidwithdraw(double);
14: stringgetName();
15: };
16:
17: doubleBankAccount::balance(){
18: returnsum;
19: }
20: voidBankAccount::deposit(doubleamount){
21: sum+=amount;
22: }
23: voidBankAccount::withdraw(doubleamount){
24: sum-=amount;
25: }
26: stringBankAccount::getName(){
27: returnname;
28: }
```

C++ тілі функция мүшелерінің анықтамасын класс анықтамасына қосуы мүмкін. Сонымен қатар класс мүшелерін, бір немесе бірнеше функция анықтамасын класс анықтамасының сыртында хабарлауға болады. Егер класс мүшелерінің функциясының атауы класс анықтамасының сыртында болса, онда программалаушы класс мүшелерінің функциясының атауын толығымен біліктендіруі тиіс. Ол үшін ***scope resolution operator*** (::) қолданылады.

Мысалы, класс анықтамасының сыртындағы класс мүшесінің функциясы төмендегідей анықталады. Класс анықтамасының сыртына класс мүшелері функциясының анықтамасын орнатудың өзіндік ерекшеліктері бар. Біріншіден, бұл класс өлшемін азайтады, оқуға ыңғайлы, функцияның анықтамасын басқа жақта болғандықтан, класты бүкіл интерфейс ретінде көруге болады.