Дәріс 7

Тақырыбы: Ашық және жабық туынды кластар. Кластар және объектілер үшін кіру мүмкіндігінің ережелері. Абстракттық кластар және полиморфизм. Контейнерлік кластар. Полиморфизм түрлері және оны іске асыру. Параметрлік полиморфизм және нұсқалар. Функция нұсқалары. Кластар нұсқалары.

Лекция жоспары:

1. Объекттер полиморфизмі
2. Кеш байланысу механизмі
3. Мұрагерлік механизмі

Лекция мазмұны

1. Объекттер полиморфизмі

Объектігі бағытталған программалауды қолдану барысында объекттердің полиморфизмі қамтамасыз етіледі. Бұл термин нені білдіреді?

Объекттердің полиморфизмі мынаны білдіреді: **әртүрлі объекттерге жіберілген** *бір және сол бір хабарлама* **программаның орындалу этапында нақты қайсы объект осы хабарламаның қабылдаушысы болуына байланысты** *әртүрлі әрекеттердің* **орындалуына** (әртүрлі тәсілдердің шақырылуы)**әкелуі мүмкін.**

Басқаша айтқанда: қасиеті, қызметі, ішкі құрылымы бойынша әртүрлі объекттер мағынасы бойынша бірдей(програмисттің көзқарасы тұрғысынан) әрекеттерді табиғаты мен ішкі құрылымына байланысты әртүрлі орындауы мүмкін.

Егер хабарламаны жіберу **тәсілдің процедура сияқты шақырылуына** алмастырылған болса(Си, Паскаль тілдерінде), онда **полиморфизм** мынаны білдіреді: тәсіл шақыруы сақталған программа кодының сол және тек сол бір бөлігі кодтың орындалу кезінде қайсы класстың экземпляры әрекетті орындаушы боп табылуына байланысты әртүрлі тәсілдердің шақырылуына әкеледі.

Мысал.

Қазірше нақты бір тілдің синтаксисін қолданып жатпаймыз. Айталық, **P** – объект көрсеткіші болсын. **P** көрсеткіші програмаманың орындалу кезінде түрлі класстарға нұсқауы мүмкін. Айталық, ол параметр ретінде қайсыбір **ppp** процедураға берілсін. Хабарлама жіберу тілдің құралдарымен былайша жазылатын болсын P.Show; **ppp** процедурасы "паскалдік" стилде былайша сипатталсын

```
procedure ppp(P); begin
  P.Show; end;
```

ppp процедурасын шақыру кезінде, **P** ретінде түрлі класстардың көрсеткіші берілуі мүмкін. Полиморфизм әсері мынадан көрінеді: сол және сол бір код (**P.Show**) **Show** хабарламасына жауап ретінде процедура параметрі ретінде берілген P көрсеткіші қайсы классты нұсқауына байланысты түрлі тәсілдің шақырылуына әкеледі.

2. Кеш байланысу механизмі

Полиморфизмді жүзеге асыру үшін ОБП тілі **кеш байланысу** механизмін қолдауы қажет. Кеш байланысу механизмінің мәні мынада: жіберілген хабарламаға жауап қату үшін керекті нақты тәсілдің адресі компиляция этапында емес, прогрмманың орындалу барысында –

хабарламаны жіберу кезінде анықталады

Кеш байланысуды жүзеге асыру үшін объект тәсілі жөніндегі мәлімет оның өзінде бар болуы керектігі айқын. Әдетте, қайсыбір класстың экземплярында өз тәсілдерінің көрсеткіші бар болады немесе класстың барлық экземплярлары үшін ортақ осындай көрсеткіштер кестесіне сілтеме жасайды. Хабарлама жіберу компилятор арқылы осы мәліметті объекттің экземплярынан оқу және қажет адрес бойынша тәсілді шақыруға алмастырылады. Класстың барлық экземплялары үшін тәсілдер бірдей болғандықтан оларды компьтер жадысында бір мәрте сақтаған дұрыс.

Си++ және Турбо Паскаль тілдерінде кеш байланысу арқылы **virtual** кілттік сөзімен сипатталған тәсілдер шақырылады. Ал, Смолток тілінде кеш байланысу арқылы барлық тәсілдер шақырылады.

Осылайша, ОБП-ң идеяларын тиімді пайдалану үшін сол тілде объект, хабарлама, объект инкапсуляциясы, класс механизмі, мұрагерлік механизмі, полиморфизм терминдерінде программалау қолдау табу керек.

Егер жеке-жеке әрбір объект үшін интерфейсі, локал жадысының құрылымы, тәсілдері сипатталатын болса, онда программа өте үлкен болып кетеді. Сондықтан ОБП-ң барлық тілдерінде класстар механизмі енгізілген.

Класс **деп бірдей объекттердің, яғни,** *бірдей интерфейске, локал жадының құрылымына***(өрістердің бірдей жиыны)***, **және** мінез-құлыққа* **(жіберілген хабарламаларды өңдеу үшін бірдей тәсілдер қолданатын) ие объекттердің жиынын атайды.**

Программа жазу кезінде программист алдымен классты сипаттайды, яғни, ішіне енетін барлық объекттердің интерфейсін, ішкі құрылымын және тәсілдерін (мінез-құлқын) сипаттайды. Бұдан соң, нақты объектті сипаттаған уақытта оның берілген классқа тиістілігін білдіреді. Және мұндай объектті берілген класстың **экземпляры** деп айтады. Әрбір объект қайсыбір класстың экземпляры болып табылады

3. Мұрагерлік механизмі

Кластардың өзін сипаттауды қысқартып, ыңғайлы ету үшін ОПБ тілдеріне мұрагерлік механизмі ендірілген. Егер программада қайсыбір объекттердің класы сипатталған болса және оларға ұқсас (яғни, ұқсас, бірақ кейбір айырмашылығы бар, сипатталған класстың басқа жаңа түрі болып табылатын) объекттердің классын сипаттау қажет болса, сипатталып қойылған класстың қасиеттерін жаңа класс мұраға алатындығын, бұдан соң жаңа класста қандай айырмашылық бар екендігін ОБП тілінің құралдарымен көрсету керек.

Қасиеттерді мұраға алған классты **ұрпақ-класс** не **ішкі класс** деп атайды.

Өз кезегінде ішкікласты тудырған классты *аталық-класс* **не** *суперкласс* **деп атайды.**

Біреулік мұрагерлікте ұрпақ-класс тек бір аталық класстың қасиеттерін мұраға алады. Мұрагерліктің бұл түрі ОБП-ң барлық тілдерінде жүзеге асқан. Ал, кейбір тілдерде мысалы С++, көп мұрагерлік яғни, ұрпаққласста бірнеше аталық класстардың қасиетері бірігу механизмі де жүзеге асқан. *Мұрагерлік* – **бұл бар класстардың қасиеттерін жаңа класстарды сипаттағанда пайдалануға және жаңа классты түгел емес, тек бастапқы класстан айырмашылығын ғана сипаттауға мүмкіндік беретін ОБП**

тілінің механизмі

Бұл айырмашылықтар мыналар болуы мүмкін:

1. Интерфейстің өзгеруі (әдетте, бұл жаңа хабарламалар енгізу)
2. Локал жадының құрылымы өзгеруі (әдетте, жаңа өрістерді қосу)
3. Аталық класс құрамындағы тәсілдің хабарламаға жауап қатуындағы

өзгеріс

Мұрагерлік механизмі жаңа программалардың құрылуын тездетеді және объектіге бағытталған програмаларды оңай бейімделгіш етеді.

Программада сипатталған класстардың бірде-бір экземпляры болмауы мүмкін. Мұндай класстарды **абстрактілі** деп атайды. Әдетте мұндай класстар үлгі (шаблон) болып табылады (ең алдымен – интерфейс шаблоны). Осы үлгілер арқылы пограммадағы реал объекттердің классы құрылады(мұрагерлік арқылы).

Бақылау сұрақтары

1. Ашық және жабық туынды кластар?
2. Кластар және объектілер үшін кіру мүмкіндігінің ережелері?
3. Абстракттық кластар және полиморфизм. Контейнерлік кластар?
4. Полиморфизм түрлері және оны іске асыру?

Ұсынылатын әдебиеттер

1. Голуб А.И. Правила программирование на С и С++. М Бином, 1996.
2. Гослинг А., Арнольд К. Язык программирование Java. – Спб: Питер, 1997.

Дейтел Х.М., Дейтел П.Дж. Как программировать на С++. М.: Бином, 1998.