Дәріс 8

Тақырыбы: Төтенше жағдайларды өңдеу. Ерекшіліктің генерациясы. Ағындарды енгізу-шығару класстары. Синхронды және кері синхронды ағындар.

Лекция жоспары:

1. Программалық жабдықтарды жөндеу. Қателердің классификациясы
2. Программаны дұрыстау
3. Ерекше жағдайлар

Лекция мазмұны

1. Программалық жабдықтарды жөндеу. Қателердің классификациясы

Программалық жабдықты құру кезіндегі маңызды кезеңдердің бірі – программаны жөндеу кезеңі. Программаны жөндеу (Debugging -отладка) кезінде, программадағы қателер табылып, бөліп алынып жөнделеді.

Программаны жөндеу үшін арнайы жөндеуші- программалар (отладчиктер) қолданылады. Программалау жүйелерінде кіріктірілген жөндеуші- программалар болады. Олар программистке программаны бақылап отыру мүмкіндігін береді, яғни қажет болған кезде тоқтату , қайта жүктеу, қадамдап орындау және т.б. сияқты әрекеттерді орындауды ұйымдастырады.

Программист өзінің құрған қосымшасы орындалған кезде болуы мүмкін қателерді анықтап, ол қателер бола қалған жағдайда программаның қалай жұмыс жасауы керек екенін алдын-ала қамтамасыз етуі тиіс. Жалпы программалау кезінде жіберілетін қателерді келесі топтарға бөледі: синтаксистік қателер, логикалық қателер және динамикалық қателер.

*Синтаксистік қателер*ге программа мәтінін теру кезінде операторлардың қате жазылуы, операторларды айыру белгілерінің қойылмауы, программа соңының көрсетілмеуі және т.б. жатады. Әдеттте синтаксистік қателерді анықтау компилятордың қызметіне жатады, яғни программа синтаксистік қатесі жөнделмейінше компиляциядан өтпейді.

Логикалық қателер, есеп алгоритмінің дұрыс құрылмауынан болады. Логикалық қатесі бар программалар түсініксіз жұмыс жасайды, мысалы, цикл алгоритмінде циклдан шығу шарты дұрыс құрылмаған болса, онда программа ешбір тоқтамастан қайталанып, нәтиже бермей жұмыс жасауы мүмкін, сол сияқты, есептеу алгоритмдерінде көбейтіндінің бастапқы мәнін нольге тең деп алғанда нәтижеде үнемі ноль шығуы мүмкін және т.б. . Мұндай қателерді программаны тестілеу, яғни әртүрлі мәндер үшін орындап көру арқылы табады.

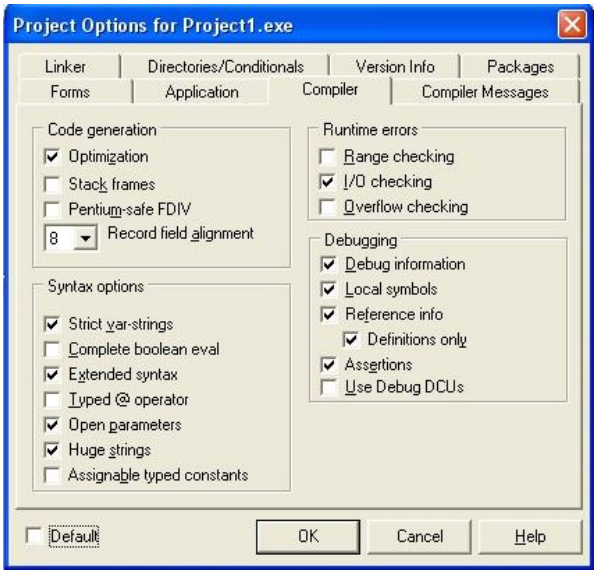
Динамикалық қателер бұл- программаның орындалуы кезінде пайда, болып оның орындалу тәртібінің бұзылуына немесе нәтижесіз тоқтап қалуына әкеліп соқтыратын қателер. Динамикалық қателерді немесе «орындау уақыты кезіндегі қателер» («ошибка времени выполнения», Runtime errors) деп те атайды. Динамикалық қателерге, мысалы, есептеу кезінде бөлшек бөлімінің нольге тең болуы, түбір астында

теріс сан кездесіп қалуы, жады ресурстарының жетпей қалуы, программада көрсетілген маршрут бойынша файлдың табылмай қалуы, принтерде қағаздың бітіп қалуы және т.б. көптеген нәрселер жатады. Қосымшалардағы осындай динамикалық қателерге байланысты болатын жағдайларды «ерекше жағдайлар» деп атап, және олармен жұмыс жасау үшін программалау тілдерінде «ерекше жағдайларды өңдеу» түсінігі енгізілген.

2. Программаны дұрыстау

Программа құруда семантикалық (мағыналық), синтаксистік және алгоритмдік қателер жиі кездеседі. Программаны компиляциялау кезінде синтаксистік қателерді компилятор өзі табады да, машина жұмысын тоқтатады. Ал, 15 орнына 25 енгізілген сияқты қатені компилятор еске алмай, программаның орындалуы аяқталған кезде қате нәтиже шығады. Программада қате жазылған оператор сияқты алгоритмдік қате де программаның орындалу нәтижесін дұрыс көрсетпейтіні сөзсіз. Сондықтан программаны құрып болған соң он мұқият қайта тексеріп шығуы керек.

Программаның синтаксистік түрде жазылуы дұрыстығын компилятордың өзі қадағалап отыруы және терезесін экранда көрсету үшін әдетте Delphi орнатылған соң ол күйге келтіріліп қойылады. Оның бірі – **Project – Options (проект –күйге келтіру)** командасын беріп, ашылған терезенің *Compiler*_ бетіндегі жазуларды 2.21 – суреттегі сияқты орнатып қою керек.



Сурет 6 - Компиляторды күйге келтіру терезесі

3. Ерекше жағдайлар

Программаға нолге бөлу және т.с.с. орындалмайтын қате әрекеттердің кездейсоқ енгізілуі де мүмкін. Олар *ерекше жағдайлар* делінеді. Ерекше жағдайларды өңдеу үшін Delphi-ге стандартты өңдеуіштер класы енгізілген.

Өңдеуіштер атаулары Е әріпінен басталады:

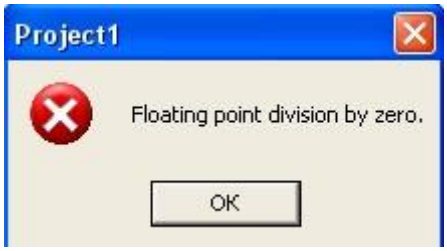
EZeroDrivide – нөлге бөлу;

EIntError – бүтін сандармен жұмыс кезінде кездесетін қателер; EConvertErrror – мәннің типін түрлендіру қатесі, т.б.

Мысалы, мынадай программа үзіндісін

```
Begin
Y:=x/k;
ShowMessage(Floattostr(y));
End;
```

Орындау кезінде нөлге бөлуге болмайтыны жазылған суреттегі хабар шығады(k=0 болған кезде).



Сурет 7 - Ерекше жағдайды хабарлау терезесі

Ерекше жағдайды өңдеу үшін Delphi ортасында блокты қорғау механизмі қарастырылған.Оның жазылу түрі:

```
Try
        <операторлар>;
except
<ерекше жағдайларды өңдеуіштер> else
<операторлар>
end;
```

Блокты қорғау блогы **Try** (байқап көру) қызметші сөзінен басталып, **end** сөзімен аяқталады.Мұнда алдымен **try**..**except** бөлімінің операторлары орындалады. Егер операторлар ерекшеліксіз орындалса, қорғалған блоктың жұмысы осымен аяқталып, басқару **end** операторынан соң жазылған операторға беріледі.Егер ерекшелік бар болса, басқару **except** - тен соң жазылған бөлімге беріледі(блокқа **else** бөлімін енгізу міндетті емес).

1 – мысал. h=1 қадам бойынша [-2;2 кесіндісінде y=2x/(x+1) функциясының мәндерін есептеу керек.

```
Procedure Tform1. Button1Click(sender:Tobject);
Var y:real;x:integer;y1:string;
Begin
```

```
 Try
        For x:=-2 to 2 do
       Begin
Y:=2*x/(x+1); y1:=FloatTo Str(y);
ShowMessage(y1);
End;
Except
        On EzeroDivide do
Begin ShowMessage(_Нөлге бөлу мүмкін емес‘);
Close;
End;
End;
```

Программаны іске қосқан кезде, алдымен суретте көрсетілген терезе шығады. Ал программа қайта қосылса типті түрлендіру қате сөйлемі жазылған шығару терезесі көрініп, программа жұмысы тоқтатылады.

2-мысал. _Ermek1‘ файлы дискіде жазулы болсын. Оны ашу операторын try блогын енгізіп жазу керек. Программа үзіндісі:

```
Var F:TextFile;
Begin
AssignFile(F, ‘Ermek1‘);
Try
        Reset(F);
Except
        //файл табылған жоқ
ShowMessage (_Файлды ашу мүмкін емес‘);
End;
End;
```

Мұнда Reset операторы бойынша файл ашылады, егер ол жоқ болса, try блогының екінші бөлімі орындалады(файлдар жөнінде төртінші тарауды қараңыз).

Бақылау сұрақтары

1. Төтенше жағдайларды өңдеу?
2. Ерекшіліктің генерациясы?
3. Ағындарды енгізу-шығару класстары?
4. Синхронды және кері синхронды ағындар?

Ұсынылатын әдебиеттер

1. Гамма Э. Хелм Р., Джонсон Р., Влиссидес Дж. Приемы объектно-ориентированного проектирование. Паттерны проектирования. – СПб: Питер, 2001.
2. Ишкова Э.А. С++ начала программирования. – М.:Бином, 2001.
3. Кетков Ю., Кетков А. Практика программирования: Visual Basic, C++ Builder, Delphi.

Прата С. Язык программирования С++. Лекции и упражнения. – Киев, 2001