Класове

Клас – съвкупност от данни, характеристики и методи. Данните се нар. още полета. Методите представляват процедури или функции. Характеристиките са данни, достъпът до които се осъществява посредством методите. Когато се задели памет за класа се създава инстанция или обект. В Delphi се работи с динамично заделяне на памет т.е. при създаване на обекта се заделя памет само за указател към него. За да се зададе обекта в смисъл да се задели памет за неговите данни (полета), трябва да се извика конструктор, а за да се освободи памет се извиква деструктор. Конструктора и деструктора са методи със специална употреба.

За всеки обект се заделя собствено копие на данните. Декларирането на класа се осъществява в раздела type т.е. там където се декларират всички други данни. Работа с анонимни типове не се допуска.

Type

=class[()]

end;

Името на класа е просто. Родител – име се в предвид родителския клас. Родителят може да бъде един единствен или да липсва т.е. няма множествена наследственост. Ако липсва влиза в сила родителят по подразбиране – TObject. Членове – полета, методи и характеристики специфични за този клас. Освен това класът има достъп и до родителските такива в зависимост от видимостта им. Допуска се създаване на наследник, който няма специфични членове. Декларацията задължително е с родител:

=class().

Type

TlistColumes=class(Tcollection)

Private

Fowner:FcustomListView;

Function GetItem(index:Integer):TlistColumes;

Procedure SetItem(Indec:integer; Value:TlistColumes);

Protected

Function GetOwner: Tpersistent,Override;

Procedure Update(Item: Tcollection.Item);

Public

Constructor Create(AOwner: TCustomListView);

Function Add:TlistColumes;

End;

Обектът се създава в раздела Var :

Var

:;

Delphi дефинира вместо нас родител по подразбиране TObject, който създава основни методи и конструктор. Делфи дефинира и TClass псевдоним на TObject. TClass се дефинира така :

TClass=class of TObject;

Деклариране на обект. При тази декларация вместо думичката class стой думата object. За тях казаните до момента превила не важат. Няма родител по подразбиране, няма конструктор, деструктор и методи, които се наследяват. Такъв обект не може да има характеристики, както и published членове. Памет за този обект се заделя с new, и освобождава с dispose. С такъв обект се работи все едно той е указател или запис, за който се заделя памет.

Съвместимост при присвояване

Класовете са съвместими при присвояване само в следния вид :

:=; Отляво е променливата от базовия клас. Същото правило важи и при предаване на параметри – формалния параметър трябва да бъде базов клас или родителски клас.

Видимост на членовете на класа

Съществуват 4 раздела :

- private – достъпа е ограничен в рамките на текущия файл; всички членове са видими във файла;

- protected – видимостта е в рамките на текущия файл и всички наследници независимо къде са дефинирани;

- public – общо достъпни членове;

- published – общо достъпни за които компилатора дефинира RunTimeInformation RTI, благодарение на нея тези данни могат да присъстват в object inspector.

Тези членове са достъпни по време на проектиране на програмата. Ключова дума може да предхожда члена (private). Ако няма записана ключова дума в списъка, елемента който е първи се подразбира published при включена {$M+} иначе се подразбира public.

Реда, в който се изреждат разделите е произволен, като някои раздели могат да отсъстват. В рамките на един раздел се спазва последователност. Първо полетата се описват, методите и накрая характеристиките. Раздела може да се прекъсва и след това да се продължава. Има и изисквания кое може да бъде published – подредените данни, дискретните данни. Данни тип string, real данни без real48, данни class, данни interface, указатели към методи, множества от базов тип, които обаче се събират в byte дума или двоична дума. Всички методи могат да бъдат published, но е забранено да има overload.