

32. Объектно-ориентированное программирование

Автор: Александр
26.08.2014 15:28

Исторически сложилось так, что программирование возникло и развивалось как процедурное программирование, которое предполагает, что основой программы является алгоритм, процедура обработки данных.

Объектно-ориентированное программирование (ООП) - это методика разработки программ, в основе которой лежит понятие «объект». Объект - это некоторая структура, соответствующая объекту реального мира, его поведению. Задача, решаемая с использованием методики ООП, описывается в терминах объектов и операций над ними, а программа при таком подходе представляет собой набор объектов и связей между ними.

Класс

Классический язык Pascal позволяет программисту определять свои собственные сложные типы данных - записи (records). Object Pascal, поддерживая концепцию объектно-ориентированного программирования, дает возможность определять классы. Класс - это сложная структура, включающая, помимо описания данных, описание процедур и функций, которые могут быть выполнены над представителем класса - объектом.

Вот пример объявления простого класса:

```
TPerson = class
```

```
private
```

```
fname: string[15]; faddress: string[35];
```

32. Объектно-ориентированное программирование

Автор: Александр
26.08.2014 15:28

```
public
```

```
procedure Show;
```

```
end;
```

Данные класса называются полями, процедуры и функции - методами.

В Приведенном Примере TPerson - это имя класса, fname и faddress - имена полей, show - имя метода.

Объект

Объекты как представители класса объявляются в программе в разделе var, например:

```
var
```

```
student: TPerson; professor: TPerson;
```

Примечание. В Object Pascal объект - это динамическая структура. Переменная-объект содержит не данные, а ссылку на данные объекта. Поэтому программист должен позаботиться о выделении памяти для этих данных.

32. Объектно-ориентированное программирование

Автор: Александр
26.08.2014 15:28

Выделение памяти осуществляется при помощи специального метода класса - конструктора, которому обычно присваивают имя `Create` (создать). Для того чтобы подчеркнуть особую роль и поведение конструктора, в описании класса вместо слова `procedure` используется слово `constructor`.

Выделение памяти для данных объекта происходит путем присваивания значения результата применения метода-конструктора к типу (классу) объекта. Например, после выполнения инструкции

```
professor := TPerson.Create;
```

выделяется необходимая память для данных объекта `professor`.

Помимо выделения памяти, конструктор, как правило, решает задачу присваивания полям объекта начальных значений, т. е. осуществляет инициализацию объекта. Ниже приведен пример реализации конструктора для объекта `TPerson`:

```
constructor TPerson.Create;
```

```
begin
```

32. Объектно-ориентированное программирование

Автор: Александр
26.08.2014 15:28

```
fname := ";
```

```
faddress := ";
```

```
end.
```

Реализация конструктора несколько необычна. Во-первых, в теле конструктора нет привычных инструкций `New`, обеспечивающих выделение динамической памяти (всю необходимую работу по выделению памяти выполняет компилятор). Во-вторых, формально конструктор не возвращает значения, хотя в программе обращение к конструктору осуществляется как к методу-функции.

После объявления и инициализации объект можно использовать, например, установить значение поля объекта. Доступ к полю объекта осуществляется указанием имени объекта и имени поля, которые отделяются друг от друга точкой. Хотя объект является ссылкой, правило доступа к данным с помощью ссылки, согласно которому после имени переменной, являющейся ссылкой, надо ставить значок `^`, на объекты не распространяется.

Если в программе какой-либо объект больше не используется, то можно освободить память, занимаемую полями данного объекта. Для выполнения этого действия используют метод-деструктор `Free`. Например, для того, чтобы освободить память, занимаемую полями объекта `professor`.

Метод

Методы класса (процедуры и функции, объявление которых включено в описание класса) выполняют действия над объектами класса. Для того чтобы метод был

32. Объектно-ориентированное программирование

Автор: Александр
26.08.2014 15:28

выполнен, необходимо указать имя объекта и имя метода, отделив одно имя от другого точкой. Например, инструкция

```
professor. Show;
```

вызывает применение метода show к объекту professor. Фактически инструкция применения метода к объекту - это специфический способ записи инструкции вызова процедуры.

Методы класса определяются в программе точно так же, как и обычные процедуры и функции, за исключением того, что имя процедуры или функции, являющейся методом, состоит из двух частей: имени класса, к которому принадлежит метод, и имени метода. Имя класса от имени метода отделяется точкой.

Инкапсуляция и свойства объекта

Под инкапсуляцией понимается скрытие полей объекта с целью обеспечения доступа к ним только посредством методов класса.

В языке Object Pascal ограничение доступа к полям объекта реализуется при помощи свойств объекта. Свойство объекта характеризуется полем, сохраняющим значение свойства, и двумя методами, обеспечивающими доступ к полю свойства. Метод

32. Объектно-ориентированное программирование

Автор: Александр
26.08.2014 15:28

установки значения свойства называется методом записи свойства (`write`), а метод получения значения свойства - методом чтения свойства (`read`).

В описании класса перед именем свойства записывают слово `property` (свойство). После имени свойства указывается его тип, затем - имена методов, обеспечивающих доступ к значению свойства. После слова `read` указывается имя метода, обеспечивающего чтение свойства, после слова `write` - имя метода, отвечающего за запись свойства.

Внешне применение свойств в программе ничем не отличается от использования полей объекта. Однако между свойством и полем объекта существует принципиальное отличие: при присвоении и чтении значения свойства автоматически вызывается процедура, которая выполняет некоторую работу.

Наследование

Концепция объектно-ориентированного программирования предполагает возможность определять новые классы посредством добавления полей, свойств и методов к уже существующим классам. Такой механизм получения новых классов называется порождением. При этом новый, порожденный класс (потомок) наследует свойства и методы своего базового, родительского класса.

Директивы `protected` и `private`

Помимо объявления элементов класса (полей, методов, свойств) описание класса, как правило, содержит директивы `protected` (защищенный) и `private` (закрытый), которые устанавливают степень видимости элементов класса в программе.

32. Объектно-ориентированное программирование

Автор: Александр
26.08.2014 15:28

Элементы класса, объявленные в секции `protected`, доступны только в порожденных от него классах. Область видимости элементов класса этой секции не ограничивается модулем, в котором находится описание класса. Обычно в секцию `protected` помещают описание методов класса.

Элементы класса, объявленные в секции `private`, видимы только внутри модуля. Эти элементы не доступны за пределами модуля, даже в производных классах. Обычно в секцию `private` помещают описание полей класса, а методы, обеспечивающие доступ к этим полям, помещают в секцию `protected`.

Иногда нужно полностью скрыть элементы класса. В этом случае определение класса следует поместить в отдельный модуль, а в программу, которая использует объекты этого класса, поместить ссылку на модуль.

Полиморфизм

Полиморфизм - это возможность использования одинаковых имен для методов, входящих в различные классы. Концепция полиморфизма обеспечивает в случае применения метода к объекту использование именно того метода, который соответствует классу объекта.