Объектно-ориентированное программирование

Объекты: свойства и методы

Основной единицей в объектно-ориентированном программировании является ***программный объект***, который объединяет в себе как описывающие его данные (свойства), так и средства обработки этих данных (методы).

Если говорить образно, то объект — это существительное, свойства объекта — прилагательные, методы — глаголы.

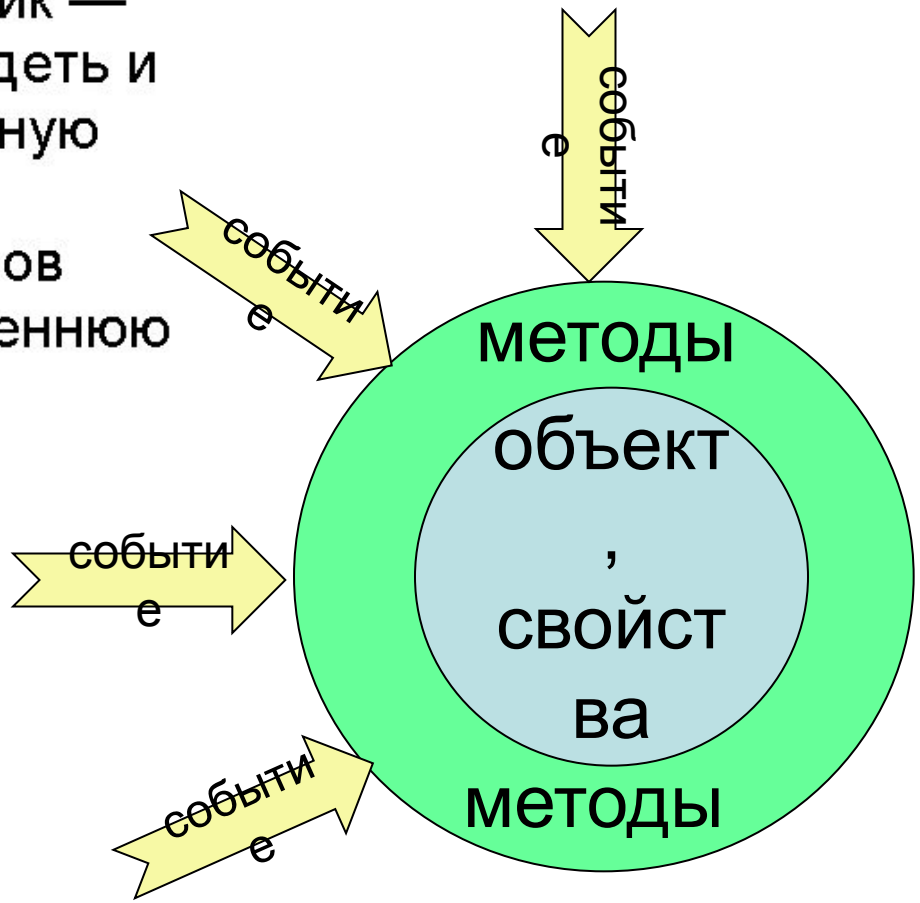
Классы объектов

Классы объектов являются шаблонами, определяющими наборы свойств, методов и событий, по которым создаются объекты.

Объект, созданный по шаблону класса, является ***экземпляром класса***. Различные экземпляры класса обладают одинаковым набором свойств, однако значения свойств у них могут быть различными.

Инкапсуляция

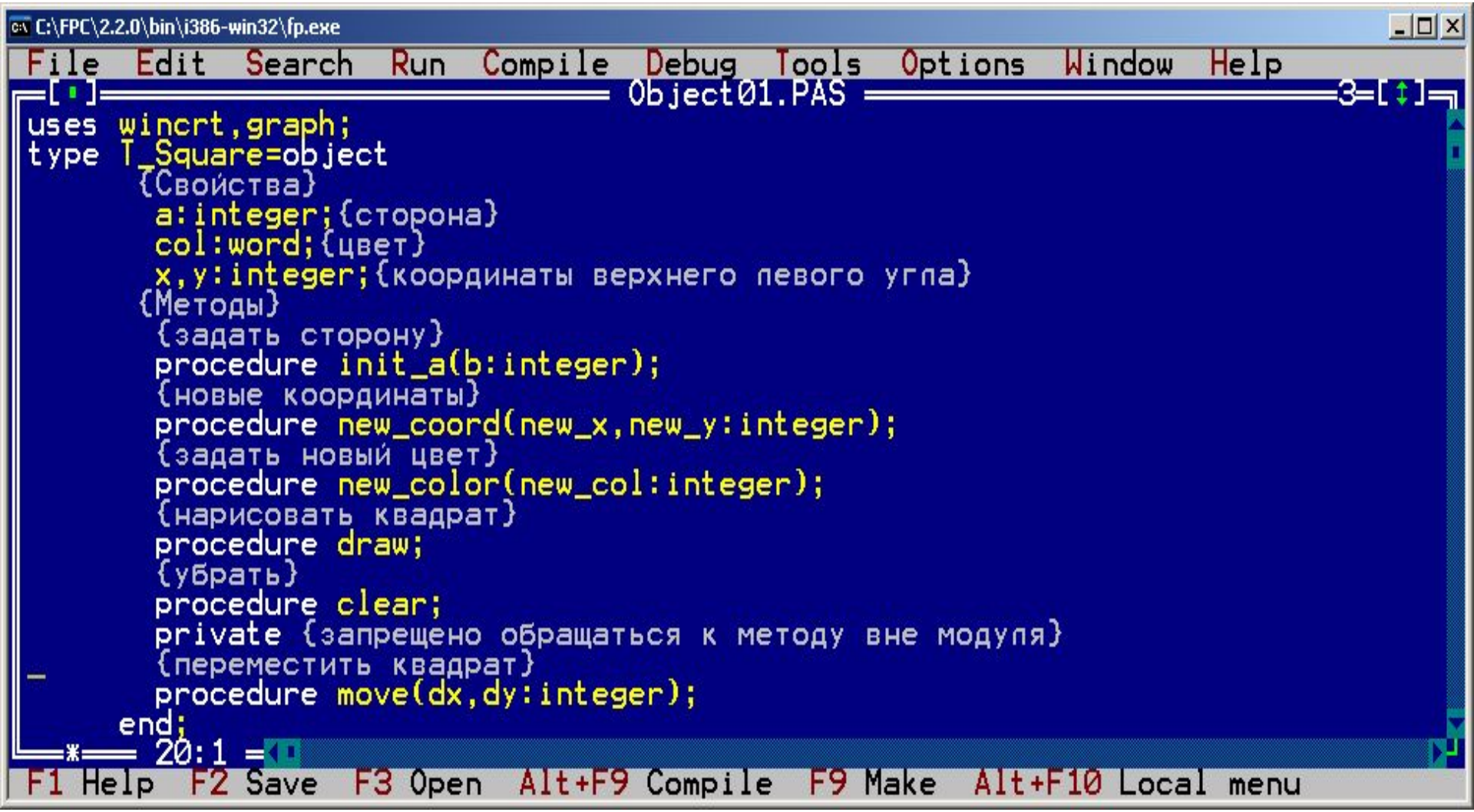
Инкапсуляция — это принцип, согласно которому любой класс должен рассматриваться как чёрный ящик — пользователь класса должен видеть и использовать только интерфейсную часть класса (т. е. список декларируемых свойств и методов класса) и не вникать в его внутреннюю реализацию.



Описание объектов

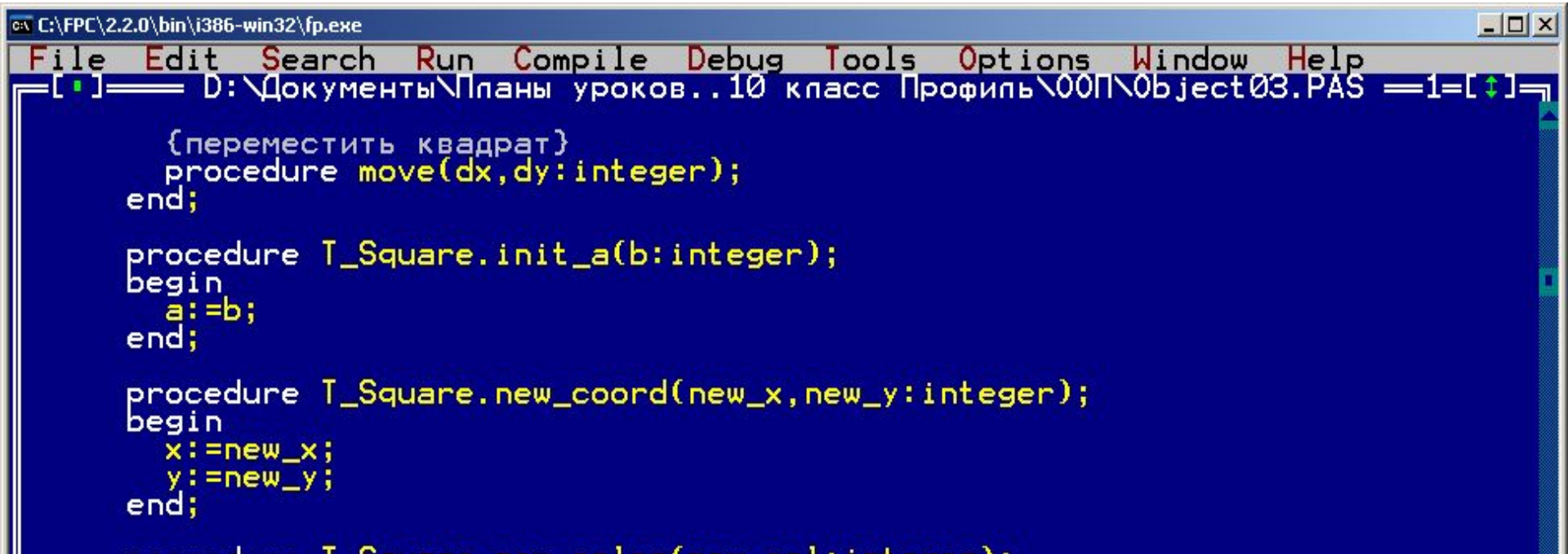
```
type <Имя объекта> = Object
       <свойство>:<тип>;
       ...
       <свойство>:<тип>;
       procedure <метод>;
       ...
       procedure <метод>;
     end;
```

Описание объектов



Описание объектов

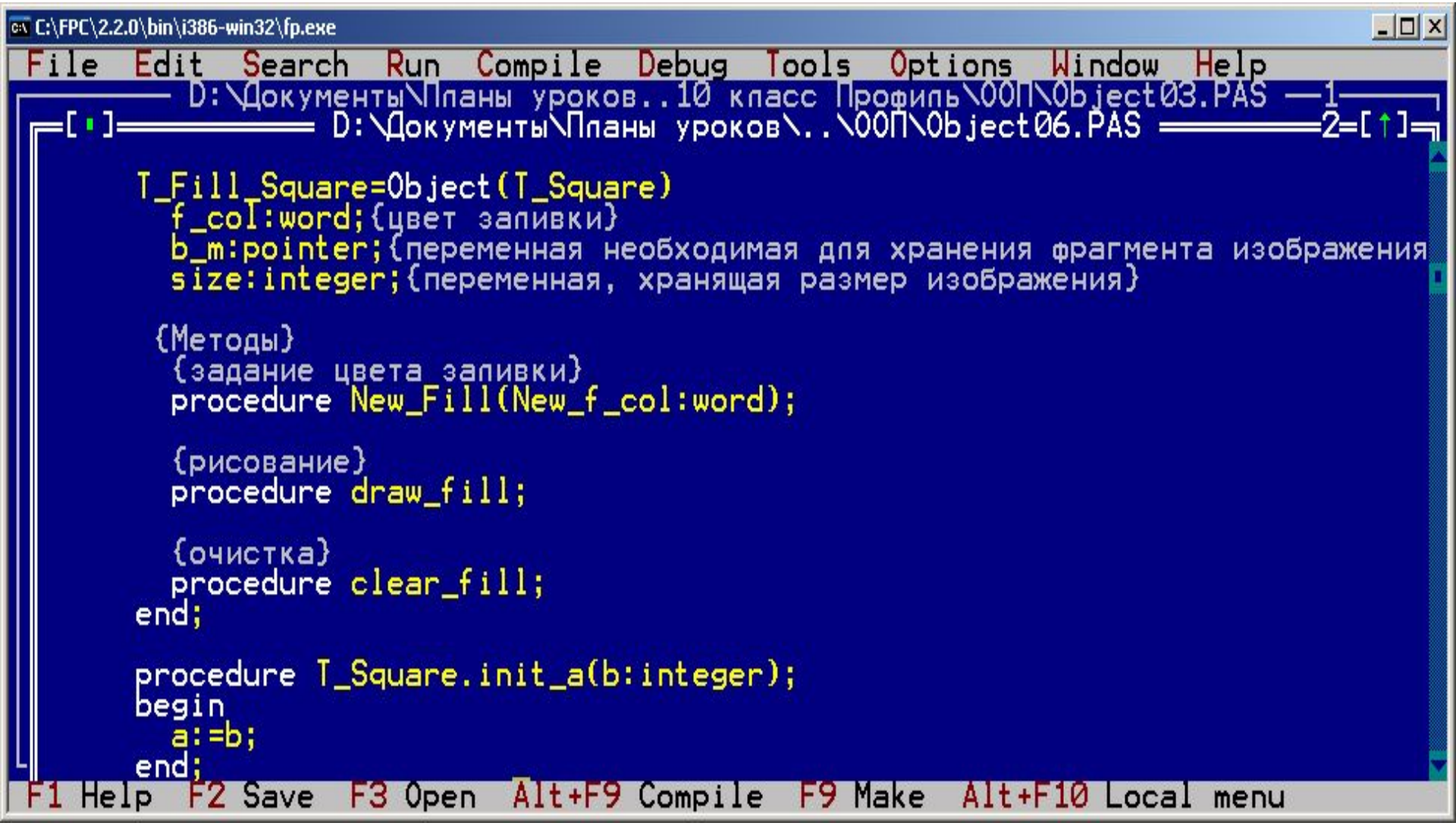
После декларирования свойств и методов объекта, все процедуры, связанные с методами должны быть описаны (используется точечная нотация).



Наследование

Наследованием называется возможность порождать один класс от другого с сохранением всех свойств и методов класса-предка (прародителя, иногда его называют суперклассом) и добавляя, при необходимости, новые свойства и методы. Набор классов, связанных отношением наследования, называют иерархией. Наследование призвано отобразить такое свойство реального мира, как иерархичность.

Наследование



Полиморфизм

Полиморфизмом называют явление, при котором функции (методу) с одним и тем же именем соответствует разный программный код (полиморфный код) в зависимости от того, объект какого класса используется при вызове данного метода. Полиморфизм обеспечивается тем, что в классе-потомке изменяют реализацию метода класса-предка с обязательным сохранением сигнатуры метода. Это обеспечивает сохранение неизменным интерфейса класса-предка и позволяет осуществить связывание имени метода в коде с разными классами — из объекта какого класса осуществляется вызов, из того класса и берётся метод с данным именем. Такой механизм называется динамическим (или поздним) связыванием — в отличие от статического (раннего) связывания, осуществляемого на этапе компиляции.