

Лекция 4

**Основы
объектно-ориентированного
программирования**

- 1. Парадигма ООП**
- 2. Основные понятия ООП**
- 3. Базовые принципы ООП**

1. Парадигма ООП

Парадигма программирования — это система идей и понятий, определяющих стиль написания компьютерных программ, а также образ мышления программиста.

Процедурно-ориентированное программирование
Основа – процедурная организация структуры программы

Объектно-ориентированное программирование (в дальнейшем **ООП**) — парадигма программирования, в которой основными концепциями являются понятия **объектов** и **классов**.

1. Парадигма ООП

Семантика и прагматика программы

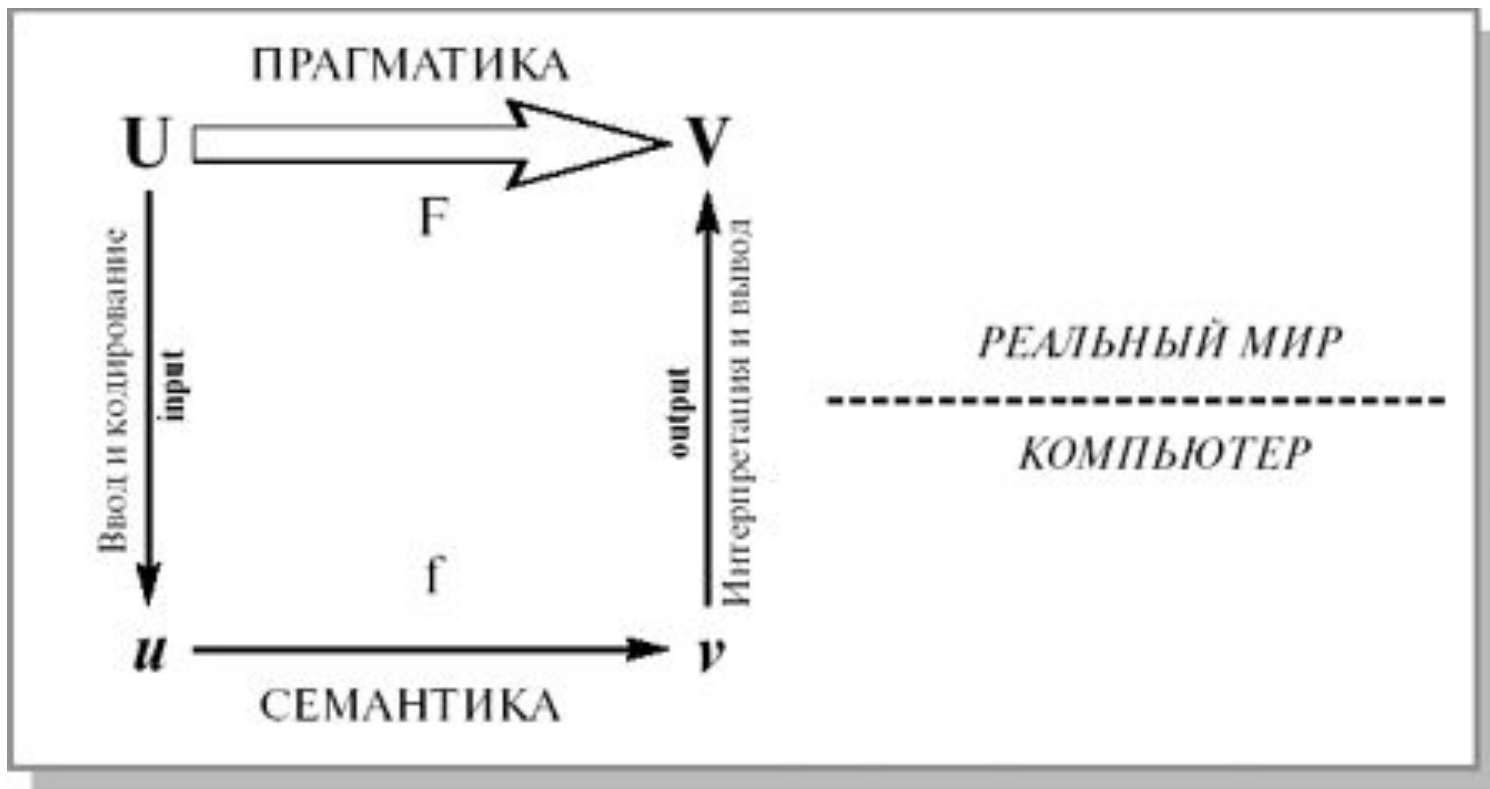


Рис. 4.1

2. Основные понятия ООП

2.1. Объекты

По определению будем называть **объектом** понятие, абстракцию или любой предмет с четко очерченными границами, имеющий смысл в контексте рассматриваемой прикладной проблемы.

Введение объектов преследует две цели:

- понимание прикладной задачи (проблемы);
- создание основы для реализации на компьютере.

Примеры объектов: стол, банк , Петр Карпенко, дело № 7461, сберкнижка и т.д.

Каждый объект **имеет состояние**, обладает четко определенным **поведением** и уникальной **идентичностью**.

2. Основные понятия ООП

2.1. Объекты

Состояние (state) - совокупный результат поведения объекта: одно из стабильных условий, в которых объект может существовать, охарактеризованных количественно;

В любой момент времени состояние объекта включает в себя **перечень свойств объекта** (обычно статический) и **текущие значения этих свойств** (обычно динамические).

2. Основные понятия ООП

2.1. Объекты

Поведение (behavior) - действия и реакции объекта, выраженные в терминах передачи сообщений и изменения состояния ; видимая извне и воспроизводимая активность объекта.

Программа, написанная с использованием ООП, обычно состоит из множества объектов, и все эти объекты взаимодействуют между собой. Обычно говорят, что взаимодействие между объектами в программе происходит посредством передачи сообщений между ними.

Например, возможные действия с некоторым файлом операционной системы ПК.

Уникальность (identity) - свойство объекта; то, что отличает его от других объектов

2. Основные понятия ООП

2.2. Классы

Формально **класс** - это шаблон поведения объектов определенного типа с заданными параметрами, определяющими состояние.

Все экземпляры одного класса (объекты, порожденные от одного класса) имеют один и тот же набор свойств и общее поведение, то есть одинаково реагируют на одинаковые сообщения.

В соответствии с **UML** класс графически изображается в виде прямоугольника, состоящего из трех частей.

В **верхней части** помещается название класса, в **средней** - свойства объектов класса, в **нижней** - действия, которые можно выполнять с объектами данного класса (методы).

Каждый класс также может иметь специальные методы, которые автоматически вызываются при создании и уничтожении объектов этого класса:

- **конструктор (constructor)** - выполняется при создании объектов ;
- **деструктор (destructor)** - выполняется при уничтожении объектов.

2. Основные понятия ООП

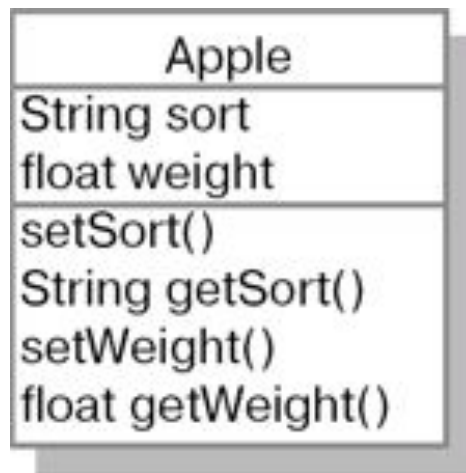


Рис. 4.2

Типы отношений между классами

- агрегация (Aggregation);
- ассоциация (Association);
- наследование (Inheritance);
- метаклассы (Metaclass).

2. Основные понятия ООП

Типы отношений между классами

Агрегация

Отношение между классами типа "содержит" (contain) или "состоит из" называется агрегацией, или включением. Например, если аквариум наполнен водой и в нем плавают рыбки, то можно сказать, что аквариум агрегирует в себе воду и рыбок.

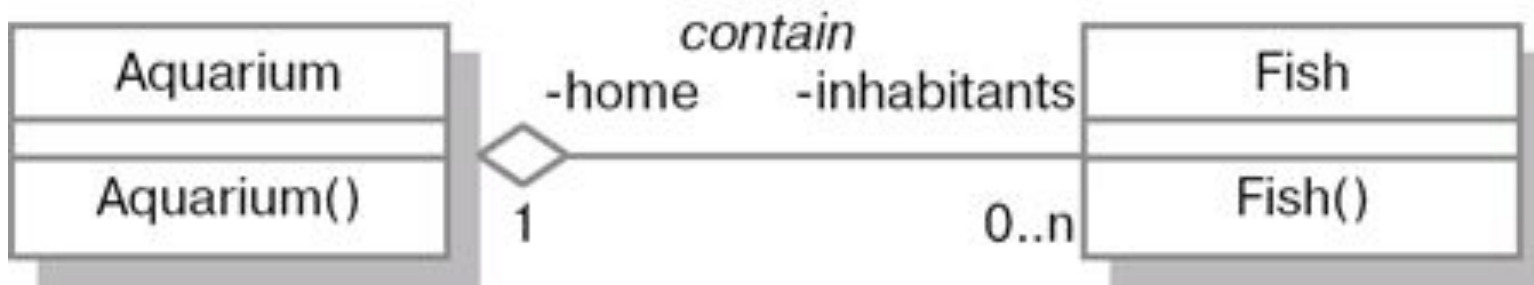


Рис. 4.3

2. Основные понятия ООП

Типы отношений между классами

Ассоциация

Если объекты одного класса ссылаются на один или более объектов другого класса, но ни в ту, ни в другую сторону отношение между объектами не носит характера "владения", или контейнеризации, такое отношение называют ассоциацией (association). Отношение ассоциации изображается так же, как и отношение агрегации, но линия, связывающая классы, - простая, без ромбика.

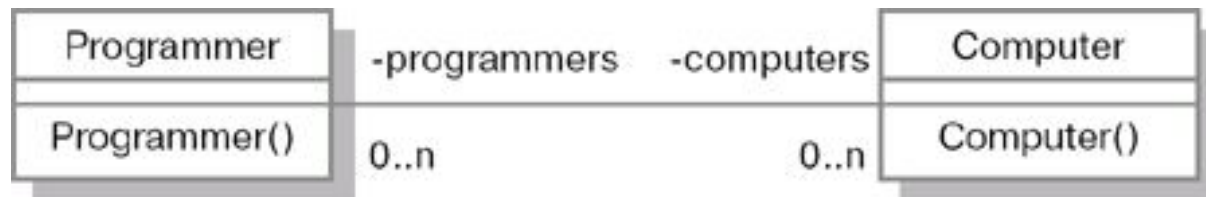


Рис. 4.4

3. Базовые принципы ООП

ООП - это методология программирования, опирающаяся на три базовых принципа:

- инкапсуляцию,
- наследование,
- полиморфизм

3. Базовые принципы ООП

3.1. Инкапсуляция

Инкапсуляция (encapsulation) - это сокрытие реализации класса и отделение его внутреннего представления от внешнего (интерфейса).

При использовании объектно-ориентированного подхода не принято применять прямой доступ к свойствам какого-либо класса из методов других классов. Для доступа к свойствам класса принято задействовать специальные методы этого класса для получения и изменения его свойств.

3. Базовые принципы ООП

3.2. Наследование

Наследование (inheritance) - это отношение между классами, при котором класс использует структуру или поведение другого класса (одиночное наследование), или других (множественное наследование) классов.

Наследование вводит иерархию "общее/частное", в которой подкласс наследует от одного или нескольких более общих суперклассов. Подклассы обычно дополняют или переопределяют унаследованную структуру и поведение.

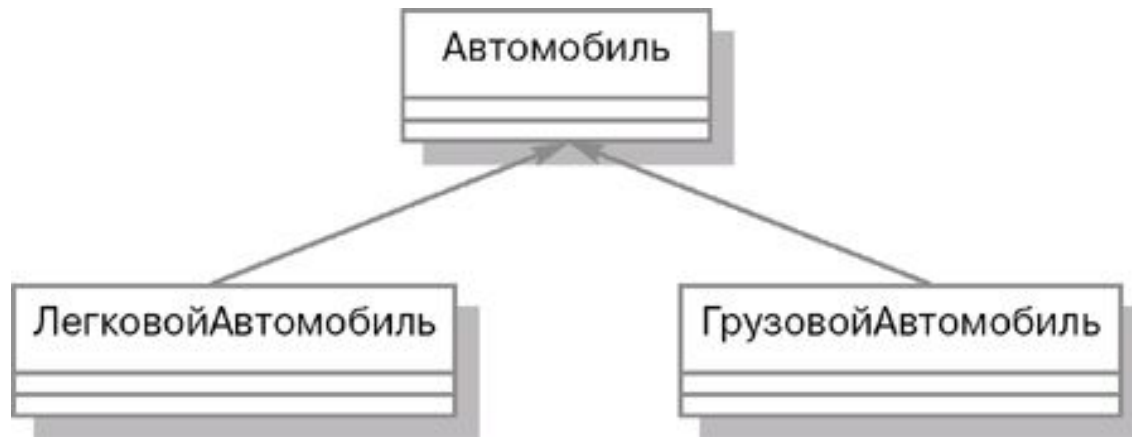


Рис. 4.5

3. Базовые принципы ООП

3.3. Полиморфизм

Слово "**полиморфизм**" греческого происхождения и означает "имеющий много форм".

Полиморфизм (polymorphism) - положение теории типов, согласно которому имена (например, переменных) могут обозначать объекты разных (но имеющих общего родителя) классов. Следовательно, любой объект, обозначаемый полиморфным именем, может по-своему реагировать на некий общий набор операций.

Полиморфизм – более сложная для понимания концепция. В сущности, она означает, что объекты, принадлежащие к одной той же ветви иерархии, получая одно и то же сообщение (то есть когда им предписывают делать одно и то же), могут действовать по-разному.

3. Базовые принципы ООП

3.3. Полиморфизм

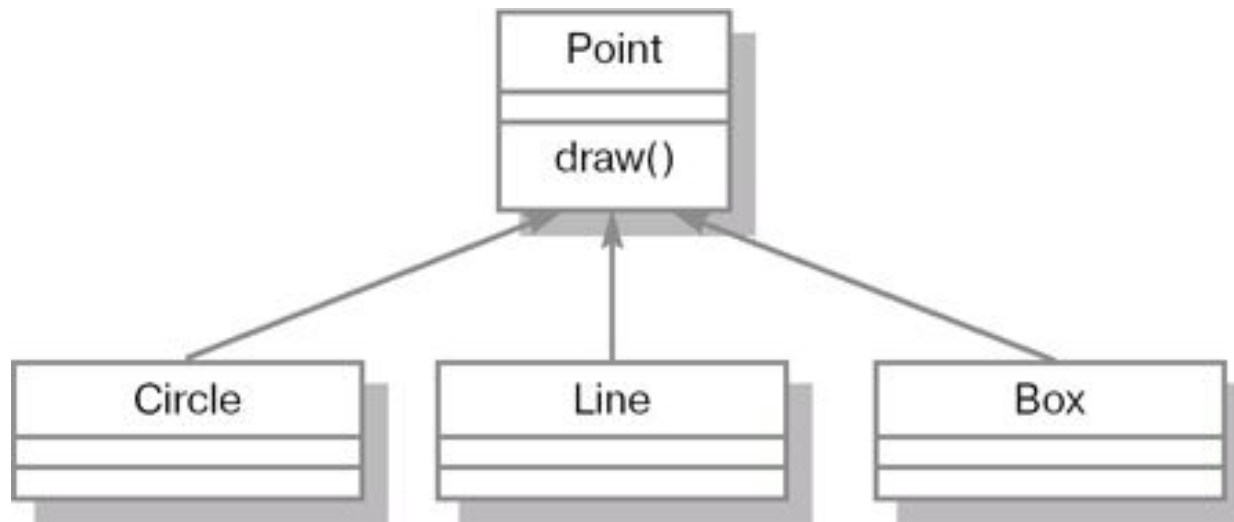


Рис. 4.6

Выводы

1. Преимущества ООП:

- уменьшение сложности программного обеспечения;
- повышение надежности программного обеспечения;
- обеспечение возможности модификации отдельных компонентов программного обеспечения без изменения остальных его компонентов;
- обеспечение возможности повторного использования отдельных компонентов программного обеспечения.

2. Основные понятия ООП - **объект** и **класс**

3. Методология ООП опирается на **три** базовых принципа:

- инкапсуляцию
- наследование
- полиморфизм