

© А.В. Ильин, В.Д Ильин, 2011

□ **ОБЪЕКТНО-ОРИЕНТИРОВАННОЕ ПРОГРАММИРОВАНИЕ (ООП)** [англ. Object-oriented programming (ООП)] — методология разработки программ для компьютерного моделирования систем произвольного назначения, представимых в виде совокупностей объектов, каждый из которых отнесён к определённому классу и наделён наборами данных (атрибутов объекта) и процедур (методов) их обработки. Классы объектов представлены в виде иерархии наследования атрибутов и методов. □

С развитием компьютерного моделирования и усложнением моделируемых систем всё более явными становились недостатки методологий разработки программ с использованием процедурно-ориентированных языков программирования (алгола, фортрана и др.).

Это объясняется отсутствием в них средств, удобных для описания сложных систем, слабой поддержкой повторной применимости ранее разработанных программ и совместных разработок коллективами программистов.

В конце 1960-х был создан язык Simula 67, ставший первым языком ООП. В нём объединение данных и процедур их обработки было названо объектом, а совокупность схожих объектов — классом.

Среди языков ООП, созданных вслед за Simula 67, наиболее удачным признан Smalltalk-80, разработанный в конце 1970-х.

Его успех способствовал развитию и распространению концепции ООП: в начале 1980-х на основе языка С был создан С++; в середине 1980-х на основе Pascal — Object Pascal; в середине 1990-х был создан язык Java.

Основные понятия

Объект в ООП – это модель экземпляра определённого класса сущностей моделируемой системы.

Класс содержит определение данных и методов, являющихся общими для входящих в него объектов.

Он задаётся атрибутами (свойствами класса), описывающими состояние его объектов, совокупностью процедур (методов класса) и правилами доступа извне (из др. частей программы) к атрибутам и методам, определяющими интерфейс класса.

В иерархии наследования класс-потомок порождается путём добавления одного или нескольких атрибутов и/или методов к атрибутам и/или методам одного или нескольких классов-родителей.

Напр., от класса «точка» [с атрибутами «абсцисса» (m_rX), «ордината» (m_rY) и методами создания, чтения и изменения координат] может быть порождён класс «цветная точка», объекты которого будут иметь дополнительные атрибут

[«цвет» (m_rgbColour)] и некоторые методы управления цветом.

На языке ООП С++ объявления классов «точка» (class Point) и «цветная точка» (class ColourPoint) могут быть записаны следующим образом [здесь все атрибуты и методы доступны извне (это правило определяется ключевым словом public)]:

```
class Point
{
public:
// Атрибуты
// Вещественная переменная, представляющая абсциссу
double m_rX;
// Вещественная переменная, представляющая ординату
double m_rY;
// Методы
// Метод создания, или конструктор, имеющий параметры – абсцисса и ордината
Point(double rX, double rY);
// Метод чтения абсциссы
double Get_X();
// Метод чтения ординаты
double Get_Y();
// Метод изменения абсциссы
void Set_X(double rX);
// Метод изменения ординаты
void Set_Y(double rY);
};
class ColourPoint : public Point
{
public:
// Целая переменная, представляющая цвет точки – дополнительный атрибут
long m_rgbColour;
// Метод создания, имеющий параметры – абсцисса, ордината, цвет
ColourPoint(double rX, double rY, long rgbColour);
// Метод чтения цвета
long GetColour();
// Метод изменения цвета
void SetColour(long rgbColour);
};
```

Определение класса задаёт тип данных, который может быть использован при определении других классов.

Напр., класс «круг» (class Circle) может иметь атрибут «центр» типа «точка» (Point m_center).

Начало объявления класса «круг» на С++:

```
class Circle
```

```

{
public:
Point m_center;
// Вещественная переменная, представляющая радиус
double m_rRadius;
// Метод создания, имеющий параметры – координаты центра и радиус
Circle(double rX, double rY, double rRadius);
// далее объявления некоторых методов для работы с кругом
};

```

Важнейшей особенностью ООП является возможность утаивания деталей реализации за интерфейсом класса (т.н. инкапсуляция).

Предположим, нужно, чтобы вне класса координаты точки могли изменяться не произвольно, а только по определенному правилу (напр., поворотом точки вокруг начала координат на заданный угол).

В таком случае в др. частях программы изменение координат делается доступным только посредством интерфейсного метода, которому передаётся значение параметра «угол поворота» (rAngle).

При этом алгоритм поворота скрыт внутри реализации метода изменения координат.

Объявление класса на C++:

```

class Point
{
// Правило доступа к атрибутам определяется ключевым словом protected
// (защищённый)
protected:
double m_rX;
double m_rY;
// Методы, доступные извне
public:
Point(double rX, double rY);
double Get_X();
double Get_Y();
// Метод поворота
void Rotate(double rAngle);
};

```

/ Применение и перспективы развития

Преимущества ООП особенно отчётливо проявляются при создании сложных программных систем, выполняемых коллективами разработчиков: одни могут проектировать функциональное поведение и структуру системы; другие — её составляющие и способы их взаимодействия; третьи — заниматься программной реализацией.

При этом разработчикам составляющих и занимающимся программной реализацией необязательно знать о системе в целом, а разработчикам системы в целом — о деталях её составляющих, способах их взаимодействия и программной реализации.

ООП, позволяющее повысить безопасность и производительность совместной разработки программного обеспечения коллективами программистов (за счёт повторного использования программного кода и утаивания деталей реализации классов) продолжает интенсивно развиваться.

Прирастает семейство языков ООП, совершенствуются системы программирования, увеличивается число программ различного назначения (для [Интернет](#)-сервисов, систем мобильной связи и др.), разработанных на языках ООП (C++, Java и др.).

/ Об изучении ООП

Можно писать программы на C++ или другом языке ООП, не следуя рекомендациям методологии ООП, а на C разрабатывать программы, следуя методологии ООП.

Полезная статья с рекомендациями по освоению ООП:

[Haibin Zhu, MengChu Zhou. Methodology First and Language Second: A Way to Teach Object-Oriented Programming](#)

См. также статью из англоязычной энциклопедии Encyclopedia of Computer Science 4th, Nature Publishing Group ©2000. ISBN:156159248X. Editors: Anthony Ralston, Edwin D. Reilly, David Hemmendinger

[Peter Wegner.OBJECT-ORIENTED PROGRAMMING \(OOP\)](#)