

Основы программирования

Основы ООП

Принципы ООП

Методологии программирования

- Процедурное программирование
- Структурное программирование
- Функциональное программирование
- Логическое программирование
- Объектно-ориентированное программирование
- ...

Процедурное программирование

Стиль программирование при котором последовательно выполняемые операторы можно собрать в подпрограммы, то есть более крупные целостные единицы кода, с помощью механизмов самого языка.

Особенность: задачи разбиваются на шаги и решаются шаг за шагом -- определяются языковые конструкции для выполнения последовательности алгоритмических шагов.

Си, Фортран, Pascal, Kotlin и другие.

Структурное программирование

В основе парадигмы (совокупности идей и понятий, определяющих стиль написания компьютерных программ) структурного программирования, лежит представление программы в виде иерархической структуры блоков.

Появилось как следствие возрастания сложности решаемых на компьютерах задач, и соответственно, усложнения программного обеспечения в конце 1960-х — начале 1970-х годов

Функциональное программирование

Процесс вычисления трактуется как вычисление значений функций в математическом понимании последних (в отличие от функций как подпрограмм в процедурном программировании).

Лисп, APL, Haskell и другие.

Объектно-ориентированное программирование

Основано на представлении программы в виде совокупности объектов, каждый из которых является экземпляром определённого класса, а классы образуют иерархию наследования.

Возникло в результате развития идеологии процедурного программирования, где данные и подпрограммы (процедуры, функции) их обработки формально не связаны.

Smalltalk,

Python, Ruby, ...

C++, Visual Basic, Delphi, Java, C#, PHP,...

...

- В каждой из рассмотренных методологий задача должна быть разбита на более мелкие единицы.
- Разработчики программного обеспечения могут выбрать одну или комбинацию из более чем одной из этих методологий для разработки программного обеспечения.

ООП. Определение

Объектно-ориентированное программирование — одна из методологий программирования, основанная на ***представлении программы в виде совокупности объектов***, каждый из которых является экземпляром определённого класса, а классы образуют иерархию наследования.

ООП

ООП подход к написанию программ, основывается на объектах, а не на функциях и процедурах.

Эта модель ставит в центр внимания объекты, а не действия, данные, логику.

Объект — реализация класса.

Все реализации одного класса похожи друг на друга, но могут иметь разные параметры и значения.

Объекты могут задействовать методы, специфичные для них.

Основные принципы ООП

(от Алана Кэя)

- Всё является объектом.
- Вычисления осуществляются путём взаимодействия (обмена данными) между объектами, при котором один объект требует, чтобы другой объект выполнил некоторое действие. Объекты взаимодействуют, посылая и получая сообщения. Сообщение — это запрос на выполнение действия, дополненный набором аргументов, которые могут понадобиться при выполнении действия.
- Каждый объект имеет независимую память, которая состоит из других объектов.

Основные принципы ООП

(от Алана Кэя)

- Каждый объект -- представитель класса, выражающего общие свойства объектов (таких, как целые числа).
- В классе задаётся поведение (функциональность) объекта. То есть все объекты, которые являются экземплярами одного класса, могут выполнять одни и те же действия.
- Классы организованы в единую древовидную структуру с общим корнем, называемую иерархией наследования. Память и поведение, связанное с экземплярами определённого класса, автоматически доступны любому классу, расположенному ниже в иерархическом дереве.

ООП. Достоинства

- использование понятий близких к предметной области;
- разделение функциональности, разделение кода - упрощение разработки;
- расширяемость программных решений;
- многократное использование кода;
- легкость модификации программ, сохранение совместимости.

ООП. Недостатки

- избыточность кода;
- сложность проектирования;
- неэффективность исполнения;
- неэффективность распределения памяти.

Объекты ...

- состояние объекта – его характеристики, параметры;
- поведение объекта – возможные действия с объектом;
- уникальность объекта – его отличие от других объектов того же класса.

Итак класс это...

Класс это тип данных, определенный пользователем. Представляет собой одну логическую сущность, например, являться моделью реального объекта или процесса.

Элементы класса: данные и функции для их обработки.
(Павловская Т.А.)

Итак класс это...

Класс это шаблон который определяет форму объекта. Он задает, как данные, так и код, который оперирует этими данными.

Спецификация класса служит для создания экземпляров класса.

Методы и переменные составляющие класс называются членами класса.

(Шилдт Г.)

Класс и экземпляр класса

Класс – это описание множества объектов,
объект – это один экземпляр класса.

Все объекты, являющиеся экземплярами одного класса, ведут себя одинаково и имеют одинаковый набор свойств. Каждый объект хранит информацию о своем состоянии.

Еще одно определение класса...

Это объявленный программистом составной тип данных, имеющий в составе:

- *поля данных* - параметры объекта (необходимые в программе), задающие его состояние (свойства объекта предметной области).
Фактически поля представляют собой значения (переменные, константы), объявленные как принадлежащие классу.
- *методы* - процедуры и функции, связанные с классом. Они определяют действия, которые можно выполнять над объектом такого типа, и которые сам объект может выполнять.

Принципы ООП

- **Абстрагирование** — это выделение набора значимых характеристик объекта, исключая не значимые для данной задачи характеристики.
- **Наследование**: наследование делает возможным повторное использование кода.
- **Инкапсуляция**: сокрытие деталей реализации класса (объекта).
- **Полиморфизм**: это свойство одних и тех же объектов и методов принимать разные формы.

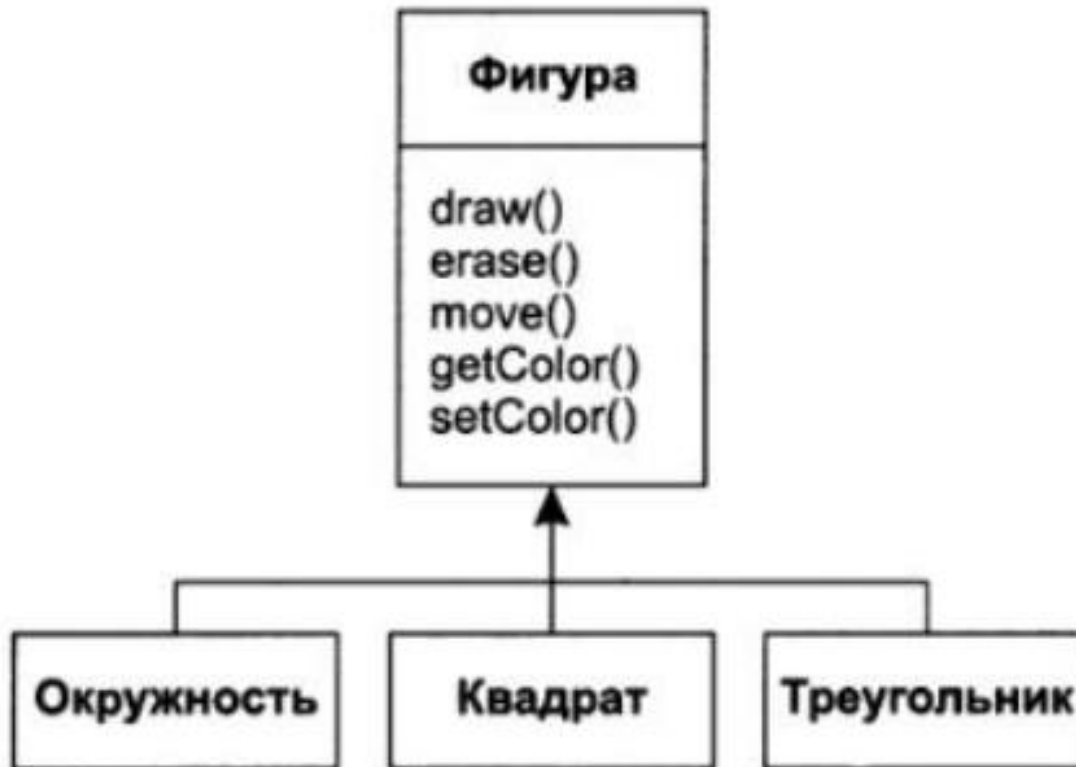
ООП. Инкапсуляция

Инкапсуляция позволяет скрыть детали реализации, и открыть только то, что необходимо в последующем использовании.

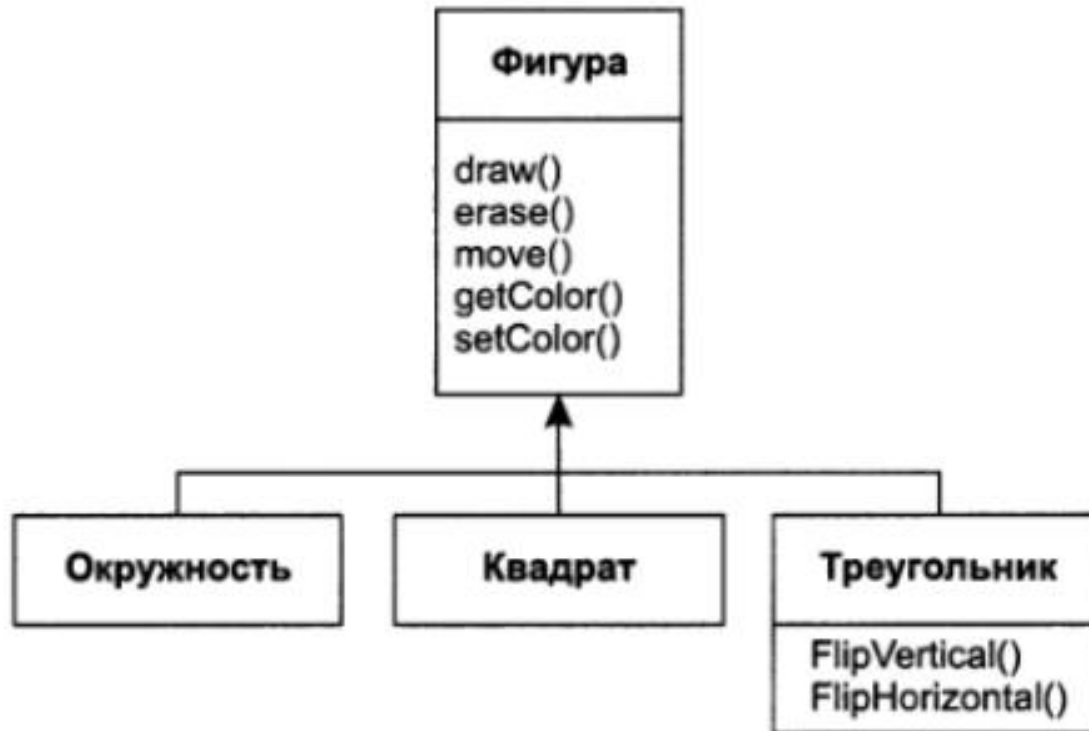
Другими словами ***инкапсуляция*** – это механизм контроля доступа.

Цель *инкапсуляции* – уйти от зависимости внешнего интерфейса класса (то, что могут использовать другие классы) от реализации. Чтобы малейшее изменение в классе не влекло за собой изменение внешнего поведения класса.

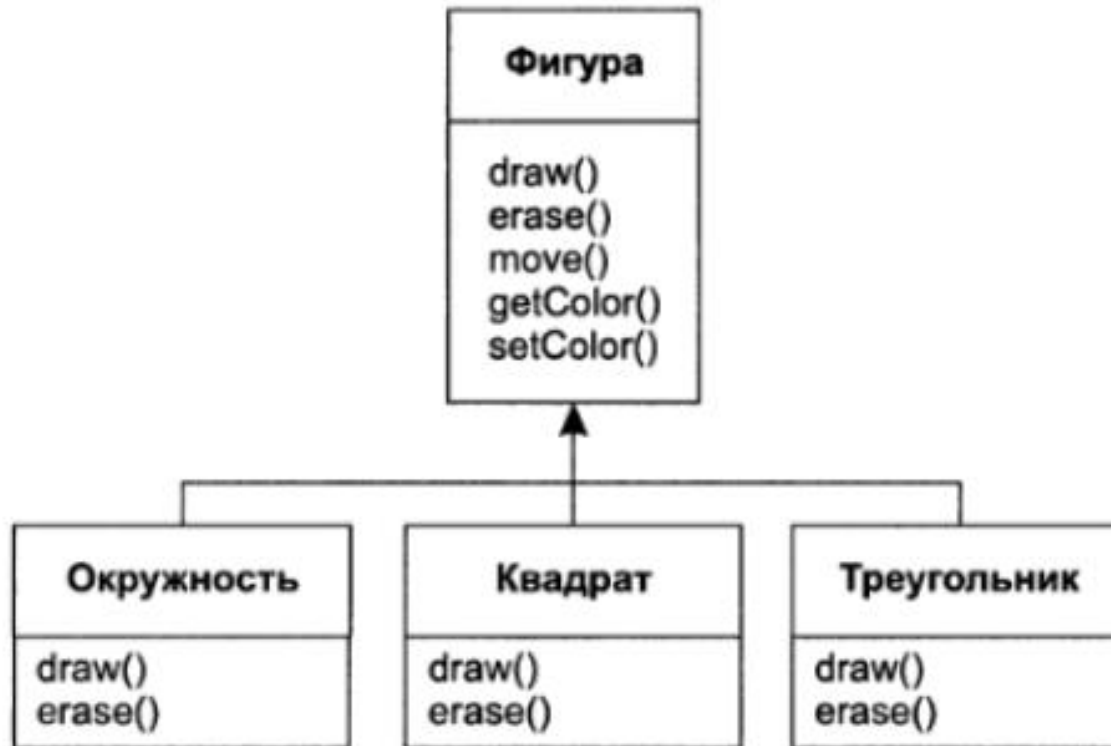
ООП. Наследование



ООП. Наследование



ООП. Наследование



ООП. Полиморфизм

В более общем смысле, концепцией полиморфизма является идея *“один интерфейс, множество методов”*.

Это означает, что можно создать общий интерфейс для группы близких по смыслу действий.

Преимуществом полиморфизма является то, что он помогает снижать сложность программ, разрешая использование того же интерфейса для задания единого класса действий.

Абстрагирование vs Абстракция

Абстрагирование – это способ выделить набор значимых характеристик объекта, исключая из рассмотрения незначимые.

Абстракция – это набор всех таких характеристик.

ООП. Абстракция

Выделение и использование только значимых для решения конкретной задачи характеристик объекта, которые представляют его с достаточной точностью.

Объект должен быть представлен минимальным набором полей и методов.

Разделение несущественных и существенных характеристик может быть осуществлено через специальный «интерфейс», в котором сосредоточено описание всех возможных применений.