Обнинский институт атомной энергетики
ИАТЭ НИЯУ "МИФИ"
Институт интеллектуальных кибернетических систем
Направление Информационные системы и технологии

Объектно-ориентированное программирование

Юрий Романович Кофтан
Заведующий лабораторией ВНИИГМИ-МЦД
koftan@obninsk.ru
Skype: koftyu
т. 910-912-1220

Объектно-ориентированное программирование (ООП)

ООП – это методология программирования, при которой программа строится в виде совокупности взаимодействующих между собой объектов. Объект представляет собой модель некой сущности, которая содержит совокупность данных и алгоритмов, реализующие методы обработки (преобразование) этих данных. Множество объектов, моделирующих сущности реальных систем, связаны между собой, обмениваются сообщениями и каждый объект на основе сообщения сам выбирает метод его обработки.

Следовательно, объектно-ориентированная программа представляет собой модель некоторого процесса или явления реальной действительности. Модель, по определению, описывает только часть признаков и свойств моделируемого процесса (если все, то тогда это и есть сам процесс). В зависимости от назначения модели (задачи моделирования) определяется, что включить в модель, а что не учитывать в ней, т.е. определяется степень *абстрактности* моделирования.

Существенным моментом является также то, что модель может основываться на иных физических принципах, чем моделируемый объект, например, описание колебательного контура математическим выражением или программой, реализующей это математическое выражение.

Для понимания связи и взаимоотношений ООП с реальной действительностью, моделями которой являются программы для ЭВМ, надо владеть основными понятиями

Теории систем и системного анализа.

Понятия теории систем и системного анализа (ТСиСА)

Система (С.) – совокупность (множество) отдельных объектов и (обязательные) связи между ними, которые придают данной совокупности больше свойств, чем простая сумма свойств составляющих С. объектов при отсутствии связей между ними (С. от греч. σύστημα, "составленный").

Теория систем (Общая теория систем) – это наука, предметом которой является разработка логико-концептуального и математического аппарата теоретического исследования объектов, представляющих собой системы.

Целью исследований в рамках этой теории является изучение:

- различных видов и типов С.;
- основных принципов и закономерностей поведения С.;
- функционирования и развития С.

Системный анализ - совокупность понятий, методов, процедур и технологий для изучения, описания, реализации явлений и процессов различной природы и характера, междисциплинарных проблем; это совокупность общих законов, методов, приемов инженерного исследования таких систем.

Цель системного анализа:

- полнее понять механизмы функционирования С.;
- повысить эффективность функционирования С.;
- получить систематические, планомерно-поэтапные методы (способы) анализа и создания (построения) С.

Системный подход – это подход, при котором любая система (объект) рассматривается как совокупность взаимосвязанных компонентов, имеющая выход (цель), вход (ресурсы), связь с внешней средой, обратную связь.

Любую *предметную область* можно рассматривать как систему.

Понятия теории систем и системного анализа (ТСиСА)

Задачи ТСиСА:

- Декомпозиция – представление С. в виде подсистем и элементов.
- Анализ – определение свойств С. и среды её окружающей.
- Синтез – построение С., обладающей заданными свойствами.
- Оптимизация – нахождение С. с экстремальными значениями её функционалов.

Некоторые термины:

Система – это некоторая совокупность взаимосвязанных объектов (компонентов С.), обладающая свойствами, не сводящимися к свойствам отдельных объектов.

Подсистема – это часть С. с некоторыми связями и отношениями, любая С. состоит из подсистем, подсистема любой С. может быть сама рассмотрена как С. (иерархическая вложенность подсистем).

Надсистема – это более крупная с С., частью которой является рассматриваемая С.

Элемент – компонент нижнего иерархического уровня С. (не имеющий внутренней структуры).

Границы системы – это материальные и нематериальные ограничители, дистанцирующие систему от окружающей среды.

Место С. – место в иерархии, определяет надсистему, в которую входит данная система, системы одного уровня с которыми взаимодействует С., подсистемы входящие в данную С.

Рамки С. – ресурсные, технологические (функциональные), географические и другие границы данной С.

Структура – устойчивая картина взаимоотношений между компонентами С. (картина связей и их стабильностей).

Процесс – динамическое изменение С. во времени.

Функция – процесс, происходящий внутри С. и имеющий определённый результат.

Состояние – положение С. относительно других её положений.

Поведение – целеориентированная активность С., служащая для осуществления контакта с окружающей средой.

Понятия теории систем и системного анализа (ТСиСА)

Основные принципы системного анализа:

- **Целостность – рассмотрение системы одновременно как единое целое и как подсистему для вышестоящих уровней.**
- **Иерархичность строения – наличие множества компонентов, расположенных на основе подчинения компонентов низшего уровня компонентам высшего уровня.**
- **Структуризация – проведение анализа компонентов системы и их взаимосвязи в рамках конкретной организационной структуры. Как правило, процесс функционирования системы обусловлен не столько свойствами ее отдельных компонентов, сколько свойствами самой структуры (связями между компонентами).**
- **Множественность – использование множества математических, кибернетических и экономических моделей для описания отдельных компонентов и системы в целом.**

Понятия теории систем и системного анализа (ТСиСА)

Упрощенная последовательность практических действий при системном анализе некого объекта (системы):

1. Определение задач исследования (анализа) или синтеза С.
2. Определение назначения и целей С.
3. Декомпозиция С.: выявление подсистем и элементов.
4. Определение структуры системы (наличия и характеристик связей между компонентами С.)
5. Определение функций системы и её компонентов.
6. Определение задач решаемых С., её подсистемами и элементами.
7. Определение принципов моделирования компонентов С. и С. в целом (в зависимости от задач исследования (анализа) или синтеза).
8. Разработка модели системы и её компонентов (в настоящее время чаще всего программной модели).
9. Исследование модели С. (проведение с ней экспериментов).
10. Анализ результатов исследования модели С.
11-а. Формирование выводов анализа, если исследуется существующая система.
11-б. Синтез (создание) новой системы.

Системный подход и программирование

Один из основных инструментов системного анализа при решении задач является моделирование системы.

Модели и моделирование

Модель – это упрощенное представление реальности. Моделью является, например, чертеж системы.

Основное свойство модели в том, что она – семантически замкнутая абстракция системы. Она строится для того, чтобы лучше понять разрабатываемую систему, визуализировать ее, определить структуру или поведение.

Сложные системы моделировать просто необходимо, поскольку иначе невозможно их воспринять как единое целое.

Моделирование – метод исследования систем на основе переноса изучаемых свойств системы на объекты другой природы. Главное в моделировании – отношение подобия между объектом моделирования и его моделью. Это один из основных методов исследования окружающей действительности.

Четыре основных принципа моделирования:

- **Выбор модели оказывает определяющее влияние на подход к решению проблемы и на то, как будет выглядеть это решение,**
- **Каждая модель может быть воплощена с разной степенью абстракции. При этом переход от одной степени абстракции к другой дает нам новую модель,**
- **Лучшими моделями являются те, которые ближе к реальности (по некоторому выбранному критерию),**
- **Системный подход рекомендует использовать совокупность нескольких моделей.**

Системный подход и программирование

Модели и моделирование

Моделирование при разработке программ: собственно разработку программ можно рассматривать как последовательную трансформацию моделей, двигаясь от самых простых общих моделей к алгоритмической модели и, наконец, к компьютерной программе.

Выделяют три основных вида моделей в зависимости от их сущности. Это физические модели (например, макеты), математические (например, алгебраическое уравнение) и информационные (например, алгоритм или программа).

Описание модели может быть: вербальное, аналитическое, численное, имитационное.

Вербальные модели – словесное описание каких-либо свойств объекта.

Аналитические – строгие математические модели объекта.

Численные – основаны на получении большого числа реализаций стохастического (случайного) процесса, который формируется таким образом, чтобы его вероятностные характеристики совпадали с аналогичными величинами решаемой задачи.

Имитационные – описание моделируемого объекта в виде алгоритма его функционирования. Такие модели описывают процессы так, как они проходили бы в действительности, её можно "проиграть" во времени как для одного испытания, так и заданного их множества. При этом результаты будут определяться случайным характером процессов. По этим данным можно получить достаточно устойчивую статистику.

После создания **модели** можно разработать **алгоритм** решения задачи.

Понятия объектно-ориентированного программирование (ООП)

Имея теперь некоторые базовые представления о системах и системном анализе можно увидеть, что создание парадигмы[1] ООП – это распространение на деятельность по программированию системного подхода, который более полноценно описывает модели реальных систем (процессов, явлений).

Объект в ООП – это реальная сущность, описывающая (моделирующая) некоторый компонент процесса или явления. Он обладает *состоянием* и *поведением,* обменивается *сообщениями* с другими объектами. Это может быть подсистема или элемент С.

Состояние объекта описывается совокупностью параметров и свойств, характерных именно этому объекту. Значения этих параметров и свойств изменяется во времени как у реального объекта.

Поведение объекта определяется совокупностью операций, которые можно выполнять над этим объектом.

Объекты, имеющие сходные (в определенных пределах) наборы свойств и параметров, а также набор операций объединяются в класс однотипных объектов.

---

[1] **Парадигма – устоявшиеся системы научных взглядов, в рамках которых ведутся исследования – комплекс теорий, стандартов и методов, которые представляют способ организации знаний.**
Парадигма программирования — это парадигма, определяющая стиль программирования, иначе говоря – некоторый цельный набор идей и рекомендаций, определяющих стиль написания программ.
Парадигма программирования представляет (и определяет) то, как программист видит выполнение программы. Например, в объектно-ориентированном программировании программист рассматривает программу как набор взаимодействующих объектов, тогда как в функциональном программировании программа представляется в виде цепочки вычисления функций.

Понятия объектно-ориентированного программирования (ООП)

Три главные свойства объектно-ориентированного языка (ООЯ) программирования: Инкапсуляция, Полиморфизм, Наследование.

Инкапсуляция (encapsulation, incapsulation).

Это механизм, связывающий воедино код и данные, которыми код манипулирует, и защищает их от несанкционированного и неправильного использования. В ООЯ код и данные можно "упаковать" в "чёрный ящик" – **объект (object)**. Объект и есть средство инкапсуляции.

Объект представляет собой сложную переменную, тип которой определён программистом.

Внутри объекта код и данные могут быть закрытыми (private) или открытыми (public). Открытая часть объекта доступна извне из любой части программы и обеспечивает управляемое взаимодействие (интерфейс) между объектами.

Полиморфизм (polymorphism).

Это атрибут, позволяющий организовать через один интерфейс доступ к целому классу методов. Выбор конкретного метода определяется компилятором в зависимости от ситуации (например, в типом переданных извне в объект данных).

Наследование (inheritance).

Это процесс, в ходе которого один объект приобретает свойства другого. Тем самым в ООЯ реализуется идея классификации (classification), когда конкретный объект является специфическим экземпляром более общей разновидности.

Понятия объектно-ориентированного программирование (ООП)

Классы.

Класс – это абстрактный тип данных, определяемый программистом, модель реального объекта (процесса, явления), состоящая из модели данных и модели функционирования, описывающей алгоритм работы объекта (преобразование входных данных в выходные).

Данные класса называются полями, синоним – данные-члены класса (по аналогии с полями структуры), а функции методами, синоним – функции-члены. Поля и методы называются элементами класса.

Конкретные переменные типа "класс" называются объектами (экземплярами или членами класса). Например, class myclass.

Конструкторы.

Конструктор – это особая функция, являющаяся членом класса. Её имя всегда совпадает с именем класса. Например, myclass(). Конструктор предназначен для инициализации нужной части данных-членов объекта и автоматически вызывается программой в момент создания объекта.

Для глобальных и статических локальных объектов конструкторы вызываются лишь однажды. Для локальных объектов конструкторы вызываются каждый раз при входе в соответствующий блок.

Деструкторы.

Деструктор – это особая функция-антипод конструктора. Её имя совпадает с именем класса с тильдой (~) перед ним. Например, ~myclass(). Деструктор предназначен для удаления объекта. Это может потребоваться для освобождения памяти или закрытия открытого ранее файла. Деструктор вызывается автоматически при выходе объекта из области видимости (для локальных объектов - при выходе из блока, где они были объявлены; для глобальных – при выходе из main; для объектов, заданных через указатели – неявно при использовании операции delete).

Программы на языке C++

Структура программ на C

#директивы препроцессора
Объявления глобальных переменных
Прототипы функций или Определения функций
int main()
{
// тело функции main
}
Определения функций, прототипы которых описаны перед **main**

Структура программ на C++

#директивы препроцессора
Объявления базовых классов
Объявления производных классов
Прототипы обычных функций

int main()
{
// тело функции main
}
Определения обычных функций